



Дубовиченко С.Б.

ОСНОВЫ ИНФОРМАТИКИ

Часть I

Алматы
2002

*ИНСТИТУТ МЕЖДУНАРОДНОГО ПРАВА
И МЕЖДУНАРОДНОГО БИЗНЕСА
«ДАНЕКЕР»*

Дубовиченко С.Б.

ОСНОВЫ ИНФОРМАТИКИ

Часть 1

*Алматы
2002*

УДК 681.14

ББК

Д

Печатается по решению Учебно - методического и Ученого Совета
Института международного права и международного бизнеса

Рецензент:

*Декан факультета подготовки иностранных граждан КазНУ,
профессор Аккушкарова К.А.,*

Дубовиченко С.Б.

Основы информатики. Часть 1. Учебник для ВУЗов. - Алматы,
Институт международного права и международного бизнеса, Алма-
ты, 2002. - 365с.

ISBN

Настоящий учебник предназначен для студентов ВУЗов, изучающих информатику. Имеющиеся издания не всегда ориентированы на начинающих пользователей и бывают сравнительно трудны для начального изучения основ информатики.

Настоящая книга преследует цель несколько упростить и систематизировать изложение материала по использованию основных компьютерных программ, в том числе: операционной системе DOS - 6, оболочки Norton Commander - 5, языка программирования Turbo Basic - 1, системы Windows - 98, которые включены в первую часть. Вторая часть книги посвящена работе с текстовым редактором Word - 97, локальной компьютерной сетью, работе в Интернете, с группами новостей в Интернете и с электронной почтой.

Приведены описания большинства основных команд этих программ, разделов различных управляющих меню, кнопок панелей инструментов и описана практическая работа с ними.

ISBN

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	11
ОПЕРАЦИОННАЯ СИСТЕМА DOS	13
УСТАНОВКА ОПЕРАЦИОННОЙ СИСТЕМЫ	13
Загрузка системы DOS	14
Установка даты	14
Установка времени	15
Дата и времена по умолчанию	16
Запрос системы	16
Перезагрузка системы.....	17
Форматирование дискеты	17
Смена дисководов.....	18
ФАЙЛЫ	20
Идентификация файлов	20
Спецификация файла	22
Копирование файла	22
Система с двумя дисковыми	23
Замещаемые символы	23
ДИРЕКТОРИИ	25
Иерархические системы файлов	25
Организация директорий	26
Изменение рабочего директория.....	27
Главная директория.....	28
Включение файла в директорий	29
Поддиректории	29
Дерево каталогов	33
Удаление поддиректория.....	34
Команда RATH.....	34
Команда APPEND.....	36
КОМАНДНЫЕ ФАЙЛЫ	37
Организация командного файла.....	37
Замещаемые параметры	39
Файл AUTOEXEC.BAT	41
НЕКОТОРЫЕ КОМАНДЫ СИСТЕМЫ DOS	55
Способы задания команд	55
Список некоторых команд системы	58
APPEND - внешняя	58
BACKUP - внешняя.....	60
BUFFERS - внутренняя	64
CHDIR - внутренняя.....	65
CLS - внутренняя.....	67
COPY - внутренняя.....	67
DEL - внутренняя	72

DEVICE - внутренняя.....	73
DIR - внутренняя	74
DISKCOMP - внешняя	77
DISKCOPY - внешняя	79
ERASE - внутренняя.....	81
FILES - внутренняя.....	81
FIND - внешняя.....	82
FOR - внутренняя	83
FORMAT - внешняя	84
IF - внутренняя.....	88
IF NOT - внутренняя	89
MKDIR - внутренняя.....	89
PATH - внутренняя.....	90
PRINT - внешняя	92
RECOVER - внешняя	95
RENAME - внутренняя	97
RESTORE - внешняя	98
RMDIR - внутренняя	100
SYS - внешняя.....	101
TREE - внешняя	102
TYPE - внутренняя	102
XCOPY - внешняя	103
ОБОЛОЧКА NORTON COMMANDER.....	105
ОСНОВНЫЕ СВЕДЕНИЯ О NC	105
Запуск Norton Comander	107
Вид экрана в Norton Comander	108
ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ В NC.....	112
Основные сочетания клавиш в NC	112
Выделение группы файлов или каталогов	114
Переключение дисков Alt + F1 или Alt + F2.....	114
Вызов меню пользователя -	116
клавиша F2	116
Просмотр содержимого файла - F3	117
Редактирование содержимого файла - F4	119
Копирование файла или каталога - F5	121
Фильтры	123
Переименование и перенос файла или каталога - F6	124
Создание каталога – клавиша F7.....	127
Удаление файла или каталога - F8	127
Главное меню NC - клавиша F9	129
Работа с архивами Alt + F5 и Alt + F6	132
Поиск файла - Alt + F7	132
Журнал команд - Alt + F8	135
Режим работы экрана - Alt + F9	135

Дерево каталогов - Alt + F10	136
Обработка расширений	136
Редактор расширений	137
Быстрый поиск файла	138
Панель инфо для жесткого диска - Ctrl + L	139
Информация о файле или каталоге - Ctrl + Q или Ctrl + Z	139
Разрезать и слить файл - Ctrl + F10	141
Установка атрибутов файла	141
НЕКОТОРЫЕ УТИЛИТЫ NC	143
Утилита копирования дискет	143
Утилита воспроизведения звука	144
Утилита форматирования дискет	144
Утилита NC_EXIT	145
Утилита уборки диска	146
Конфигурация NC	147
Режим эмуляции Терминала	152
ПРОГРАММИРОВАНИЕ НА TURBO BASIC	154
ЭЛЕМЕНТЫ ЯЗЫКА	154
Константы	154
Переменные	156
Массивы переменных	157
Преобразование типов	157
Выражения и операции	158
ОСНОВНЫЕ ОПЕРАТОРЫ BASIC	163
Оператор CALL	163
Оператор CHAIN	163
Оператор CLOSE	164
Операторы COMMON и SHARED	164
Оператор DATA	165
Оператор DEF FN	166
Операторы DEFINT/LNG/SNG/DBL/STR	167
Оператор DO ... LOOP	168
Операторы DIM и REDIM	169
Оператор END	169
Оператор ERASE	170
Оператор EXIT	170
Операторы FOR и NEXT	170
Оператор FUNCTION	172
Операторы GOSUB и RETURN	173
Оператор GOTO	174
Операторы IF...THEN [...ELSE] и IF...GOTO	174
Оператор INPUT	175
Оператор INPUT #	176
Оператор LINE INPUT	177

Оператор LINE INPUT#.....	178
Оператор MOD	179
Операторы LPRINT и LPRINT USING	179
Операторы ON...GOSUB и ON...GOTO	180
Оператор OPEN	180
Оператор OPTION BASE.....	182
Оператор PRINT	183
Оператор PRINT USING	185
Операторы PRINT# и PRINT# USING	188
Оператор RANDOMIZE.....	189
Оператор READ.....	190
Оператор REM.....	191
Оператор RESET	191
Оператор RESTORE	192
Оператор RUN	192
Оператор STOP.....	192
Оператор SUB.....	193
Операторы WHILE ... WEND	193
Оператор WRITE	194
Оператор WRITE#.....	194
ВСТРОЕННЫЕ ФУНКЦИИ	196
Функция ABS.....	196
Функция ATN	196
Функция CINT	197
Функция COS.....	197
Функция EXP.....	198
Функция INPUT\$.....	198
Функция INT	199
Функция LOG	199
Функция RND.....	200
Функция SIN	200
Функция SQR.....	201
Функция TAN	201
ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ.....	202
Оператор DATE\$	202
Оператор TIME\$	203
Оператор COLOR	204
Оператор CLS	205
Оператор SCREEN	206
Оператор PALETTE	207
Оператор KEY	207
Оператор BEEP.....	208
Оператор SOUND.....	209
Оператор PLAY	209

ПРИМЕРЫ ПРОГРАММ НА BASIC	212
Вычисление значений функции	212
Вычисление суммы значений функции.....	213
Запись и считывание информации на диске	214
Вычисление интеграла	215
Вычисление произведения двух матриц.....	217
Поиск корня функции	219
Нахождение минимума функции	222
Определение детерминанта матрицы	223
Решение системы линейных уравнений	225
Обращение матрицы	229
Вычисление Гамма функции	233
СИСТЕМА WINDOWS - 98.....	235
ЗАПУСК WINDOWS.....	237
Вход в Windows	238
Знакомство с компьютером	238
Панель задач и Главное меню	239
Папка Мой компьютер	240
Программа Проводник.....	241
Работа с мышкой	242
Работа с окнами	243
РАБОЧИЙ СТОЛ, ПАНЕЛЬ ЗАДАЧ И ГЛАВНОЕ МЕНЮ.....	244
Общий вид Рабочего стола	245
Настройки стиля Рабочего стола и папок.....	245
Настройка Рабочего стола для нескольких пользователей.....	258
Настройка Главного меню и Панели задач	259
Новые элементы Главного меню	264
Выключение компьютера	266
НАСТРОЙКА МЫШИ И КЛАВИАТУРЫ.....	268
Изменение параметров работы мыши	268
Изменение параметров клавиатуры.....	271
СОЗДАНИЕ ЯРЛЫКОВ.....	273
Создание ярлыка в папке	273
Помещение ярлыка на Рабочий стол	273
Добавление подменю в Главное меню	276
ШРИФТЫ.....	278
Установка нового шрифта	278
Удаление шрифта с компьютера.....	279
Просмотр установленных шрифтов.....	279
Поиск схожих шрифтов	281
МНОГОЯЗЫКОВАЯ ПОДДЕРЖКА.....	282
Установка многоязыковой поддержки	282
Смена раскладки клавиатуры	282
Установка дополнительного языка.....	284

Удаление языка.....	285
РАБОТА С ПАПКАМИ И ФАЙЛАМИ.....	286
Открытие файла или папки.....	286
Изменение программы, используемой для открытия файла	287
Открытие недавно использованных файлов	288
Создание новой папки.....	289
Изменение имени файла или папки	290
Удаление файла или папки	291
Восстановление удаленных объектов.....	291
Окончательное удаления файлов	292
Изменение способа отображения объектов в папке.....	293
Отображение имен файлов и расширений имен файлов	294
Копирование файла или папки.....	294
Отправка файла или папки на диск.....	295
РЕЖИМ ПОИСКА	296
Поиск файла или папки.....	297
Дополнительные возможности поиска.....	299
СПОСОБЫ ЗАЩИТЫ ФАЙЛОВ	301
Архивация файлов.....	301
Использование паролей	301
Защита файлов назначением пароля документу Word.....	304
МУЛЬТИМЕДИА.....	306
Сопоставление звуковых сигналов событиям в программах	306
Регулировка уровня воспроизведения.....	306
Изменение размеров окна воспроизведения видеоклипов	309
ПЕЧАТЬ ДОКУМЕНТОВ.....	310
Установка принтера	310
Установка принтера "по умолчанию".....	312
Печать документов.....	312
Назначение общего принтера.....	313
ПРЕОБРАЗОВАНИЕ ДИСКА В FAT 32	315
Принцип хранения информации	315
Преобразование файловой системы.....	317
СТАНДАРТНЫЕ ПРОГРАММЫ	319
Игры Windows	319
Калькулятор	319
Графический редактор Paint.....	321
Текстовый редактор Word Pad.....	322
Текстовый редактор Блокнот.....	324
Лазерный проигрыватель.....	325
Универсальный проигрыватель	326
Программа звукозаписи	326
Программа Kodak Imaging.....	327
Использование Портфеля	328

Программа Нурер Terminal	331
Набор телефонного номера	331
СЛУЖЕБНЫЕ ПРОГРАММЫ	332
Таблица символов.....	332
Окно буфера обмена.....	333
Программа очистки диска.....	333
Дефрагментация жесткого диска	334
Мастер обслуживания	335
Индикатор ресурсов	336
Программа проверки диска	338
Программа планировщика заданий.....	338
Проверка системных файлов.....	343
Отображение системных данных.....	344
Системный монитор.....	347
Меню Справка позволяет получить более подробные сведения о работе этой программы.....	347
Приветствие Windows.....	347
ПРИКЛАДНЫЕ ПРОГРАММЫ	348
Запуск прикладных программы	348
Свертывание всех открытых окон	348
Выход из программы.....	350
Примеры прикладным программ	350
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	364

*Автор выражает искреннюю
благодарность Институту
международного права и
международного бизнеса
«Данекер» за публикацию данной
книги*

ВВЕДЕНИЕ

Первый раздел книги посвящен описанию давно известной операционной системы (ОС) MS - DOS. Название MS - DOS расшифровывается, как дисковая операционная система. ОС - это программа, координирующая действия вычислительной машины - компьютера так, что под ее управлением осуществляется выполнение различных прикладных программ.

Что нужно знать об операционной системе вычислительной машины, зависит от решаемых на ней задач. Если вы хотите стать прикладным программистом и вам нужно уметь отлаживать свои программы и работать с файлами и папками - директориями, то для этого достаточно понимания основ операционной системы.

Команды ОС можно набирать и маленькими, и большими буквами. В дальнейшем, в файлах они будут приведены строчными буквами, а в тексте прописными. То же самое относится и к именам файлов. Имена дисков будут приводиться заглавными буквами.

При введении команды необходимо соблюдать синтаксические правила (включая пробелы и знаки пунктуации). Не забывайте оставлять пробел между именем команды и шифром устройства, между именем команды и именем файла. В противном случае, машина не поймет, что от нее требуется. Не набирайте "l" (маленькое эль) вместо "1" (единица) и букву "O" или "o" вместо "0" (ноль). Если после введения команды машина не заработает, проверьте введенный текст - заданные команды.

Во втором разделе книги приводится описание программы оболочки Norton Commander (NC). Программа NC представляет собой классический пример программной оболочки, которая работает на базе операционной системы DOS или Windows и позволяет существенно упростить общение с компьютером по сравнению с работой только в рамках какой -нибудь ОС. Новая пятая версия NC существенно расширена по сравнению с предыдущими реализациями этой программы и позволяет активно использовать мышку.

В программу включены новые возможности такие, как резка и сборка файлов. Новая версия программы работает с русифицированным терминалом, расширены возможности по копированию и пере-

носу файлов, и многое другое. Новый графический интерфейс и активная работа мышки максимально приближают ОС к графическим оболочкам типа Windows, хотя ни в коем случае ОС, даже в таком варианте, не может их заменить. По сути, ОС аналогичен Проводнику Windows 95/98 или Диспетчеру файлов Windows 3.1/3.11. Здесь будет рассмотрена пятая версия этой программы, впервые полностью русифицированная, с очень широким набором дополнительных программ - утилит, существенно расширяющих ее возможности.

Третий раздел книги содержит сравнительно полное описание алгоритмических языков программирования Quick Basic и Turbo Basic, работающих на основе стандартной операционной системы MS - DOS. Описываемые здесь версии языка Basic представляет собой наиболее мощную реализацию этого языка для персональных ЭВМ типа IBM PC. Они имеют намного более широкие возможности по сравнению со старыми версиями языка Basic A или Basic Q. Рассматриваемые версии по своим структурным возможностям очень похожи на хорошо известный алгоритмический язык Fortran версии Дубна или 77.

Четвертый раздел книги посвящен описанию системы Microsoft Windows - 98. Новая версия операционной системы Windows 98 года обладает повышенной производительностью, надежностью и улучшенным оформлением **Рабочего стола** по сравнению с Windows - 95. Из самой операционной системы и любой прикладной программы открыт непосредственный доступ к Интернету. Новые возможности **Рабочего стола**, **Панели задач** и **Главного меню** помогают быстро открывать или создавать **Ярлыки** и запускать различные программы. Новые возможности позволяют работать быстрее и эффективнее, и делают ее самой удобной в работе, и самой настраиваемой из всех доступных на сегодняшний день операционных систем Windows.

ОПЕРАЦИОННАЯ СИСТЕМА DOS

Первый раздел книги посвящен описанию операционной системы (ОС) DOS, выпущенной фирмой Microsoft в начале 80 - х годов. С тех пор появилось множество версий этой системы, которая постоянно совершенствовалась. Последняя версия ОС под названием 6.2 будет рассмотрена в этом разделе. Название MS - DOS расшифровывается, как дисковая операционная система. Операционная система - это программа, координирующая действия вычислительной машины так, что под ее управлением осуществляется выполнение прикладных программ.

Перечислим три основные функции операционной системы:

1. Обмен данными между компьютером и различными периферийными устройствами, например, принтерами.
2. Обеспечение системы организации и хранения файлов в постоянной памяти компьютера на его жестком диске или на дискетах.
3. Загрузка прикладных программ в оперативную память и обеспечение их выполнения на компьютере.

В дальнейшем изложении будем обозначать командные клавиши на клавиатуре с большой буквы, например, Esc, Del, Alt. Комбинации клавиш, например, Ctrl + D и Ctrl + N на экране будут выглядеть следующим образом: ^D, ^N. Какое бы обозначение не встретилось в книге, реально необходимо одновременно нажать клавишу Ctrl и, удерживая ее, нужную клавишу на клавиатуре. Для ввода команды нужно набрать ее с клавиатуры и нажать клавишу Enter, чтобы система "знала", что вы закончили набирать директиву.

УСТАНОВКА ОПЕРАЦИОННОЙ СИСТЕМЫ

Операционная система это основная, базовая программа, под управлением которой работают все остальные, в том числе прикладные программы пользователя. На ее основе работают и, так называемые, программы "Оболочки", например программа Norton Commander.

Самые первые версии DOS назывались 1.X, 2.X, где X обозначает некоторую цифру, определяющую вариант версии. Затем появились версии 3.X, 4.0, 5.0 и 6.0. Самая последняя версия системы называется 6.2, а ее русифицированный вариант 6.22. В дальнейшем выпуск операционных систем DOS прекратился и фирма Microsoft перешла на разработку различных версий системы Windows.

Загрузка системы DOS

Операционная система MS - DOS обычно хранится на гибкой дискете, если машина не имеет жесткого диска - постоянной памяти. При включении компьютера операционная система с дискеты загружается в оперативную память.

Перед включением машины поместите дискету с MS - DOS на дисковод A:. Если компьютер имеет два дисковода и вы не знаете, какой из дисководов обозначен A:, а какой B:, уточните это в руководстве, приложенном к вашему компьютеру. Если на компьютере установлен большой дисковод для гибких дисков на 5,25 дюйма, то он обычно называется A:. Если второй дисковод предназначен для дисков на 3,5 дюйма, то он называется B:. После установки дискеты закройте крышку дисковода и включите машину. Несколько секунд машина будет бездействовать. На самом деле, в это время производится серия проверок работоспособности внутренних устройств машины. Затем на экране в командной строке появится курсор - маленькая светящаяся черточка, указывающая место запроса системы. Вы услышите легкие щелчки, после чего зажжется лампочка индикатора дисковода. Включение индикатора означает, что MS - DOS переписывается с дискеты в оперативную память машины. После загрузки на экране появится стандартное приглашение к вводу команд DOS A:\>.

Если компьютер имеет жесткий диск C: и операционная система находится на нем, то загрузка системы происходит сразу после включения компьютера. После загрузки на экране появляется командная строка вида C:\>, в которой можно задавать команды системы.

Установка даты

После загрузки операционной системы, если компьютер не имеет жесткого диска обычно вводится точная дата. На экране появится следующая (или подобная ей) запись:

```
Current date is Tue 01-10-1999 - текущая дата.  
Enter new date: - введите новую дату.
```

Последняя строка - это вопрос на введение новой (сегодняшней) даты. Вводимая дата используется для регистрации организованных (новых) файлов. Дата регистрации может оказаться крайне полезной даже много времени спустя после ее введения. В первой строке указано значение Current date. Это дата генерации системы на дискете. Она будет фигурировать при каждой загрузке.

Для введения в систему новой даты необходимо выполнить следующие действия:

1. Набрать номер текущего месяца, например, 01 = январь, 02 = февраль и т.д. Первый (или ведущий) ноль можно опустить (т.е. январь можно вводить и, как "1", и как "01").
2. Набрать тире (-) или слеш (/), разделяющий месяц и день.
3. Набрать сегодняшнее число. Первый ноль можно опустить.
4. Набрать тире (-) или слеш (/), отделяющий число от года.
5. Набрать год. MS - DOS воспринимает любое значение в интервале от 1980 до 2099. Можно набрать только последние две цифры года (например, 1999 можно ввести и, как 1999, и как 99).
6. Нажать клавишу Enter.

На экране появится значение новой даты:

```
Current date is Tue 01-10-1999
Enter new date: 5/20/99
```

После нажатия клавиши Enter, MS - DOS проверяет, попадают ли значения числа, месяца и года в интервал допустимых значений, а также синтаксис записи. Если дата задана правильно, она заносится в память машины. В противном случае на экран выводится сообщение Invalid date.

Если компьютер имеет жесткий диск, то текущая дата обычно обновляется автоматически и задавать ее при каждом включении не нужно.

Установка времени

После введения даты MS - DOS выдает вопрос на введение нового (т.е. текущего) времени. Введенное значение используется для регистрации организованных в текущий период времени файлов. Время регистрации, также как и дата, может оказаться необходимым год или полгода спустя после создания файла. На экране появится значение времени:

```
Current time is 10:08:32
Enter new time:
```

Для введения нового времени необходимо произвести следующие операции:

1. Набрать текущий час (допускаются значения от 01 до 24). Ведущий ноль можно опустить, т.е. можно ввести 01 или 1).
2. Набрать двоеточие (:), отделяющее часы от минут.
3. Набрать минуты. Ведущие нули необязательны.
4. Набрать двоеточие, а затем секунды. Секунды можно не набирать.
5. После этого можно указать доли секунд, набрав перед значением сотых долей отделяющую их от значения секунд десятичную точку (.
6. Нажать клавишу Enter.

После нажатия клавиши Enter MS - DOS проверяет введенное значение. Если время введено в правильном формате, то оно записывается в память машины. В противном случае выдается сообщение Invalid time. Значение времени корректируется в соответствии со значением таймера несколько раз в секунду.

Если компьютер имеет встроенный жесткий диск, то текущее время обычно обновляется автоматически и задавать ее при каждом включении не нужно.

Дата и время по умолчанию

Могут возникнуть ситуации, когда пользователь не хочет вводить новую дату и время при загрузке системы. В этом случае при запросе на введение даты и времени нужно просто нажать клавишу Enter. Время и дата устанавливаются в соответствии со значениями Current date и Current time, т.е. в память загружаются время и дата генерации MS - DOS, никак не связанные с реальными значениями. При такой загрузке системы DOS экран остается пустым в тех местах, где обычно высвечиваются новые значения времени и даты.

Запрос системы

После введения времени и даты MS - DOS выдает запрос на введение команды. За символами запроса следует курсор, сигнализирующий о готовности системы принять команду:

The IBM Personal Computer MS - DOS

Version 6.2 Copyright Micrisoft Corp 1995

A:\>

Если загрузка проводилась с жесткого диска C:, то командная строка будет выглядеть:

```
C:\>
```

Перезагрузка системы

Описанная выше процедура загрузки операционной системы начинается с включения компьютера и называется "холодным стартом". Однако, при сбое или зависании компьютера, когда он перестает реагировать на нажатия на любые клавиши, можно перезагрузить систему и с включенной машиной. Не обязательно ее выключать, а затем включать снова. В этом случае процесс загрузки называется "теплым стартом". Не выключая машину, одновременно нажмите клавиши Ctrl + Alt + Del. Система MS - DOS начнет заново загружаться в память. Как и в случае "холодного старта", вы получите запрос на введение времени и даты, а затем появится командная строка.

Форматирование дискеты

Чтобы операционная система могла обслужить новую дискету (записать на нее данные, стереть их и т.п.), дискета должна быть отформатирована. В процессе форматирования дискета разбивается на доступные MS - DOS участки, называемые секторами. При этом производится анализ на сбойные, испорченные участки, а также формируется ее директорий - область дискеты, где хранится информация о записанных на ней файлах. Большинство версий MS - DOS предусматривают автоматическое форматирование дискеты (если она не была отформатирована заранее) при работе команды DISKCOPY. Если процедура форматирования при выполнении DISKCOPY не предусмотрена в вашей версии MS - DOS, то при копировании информации на дискету на экране появится сообщение:

```
Disk error reading drive A: - ошибка диска при чтении.  
Abort, Retry, Ignore? - отменить, повторить или игнорировать.
```

Введите "A" или "a" вместо Abort (Отмена) и на экране появится стандартный системный запрос. Установите системную дискету и наберите: FORMAT A:. Вы получите на экране:

```
A:\>format A:  
<=Enter
```

```
Insert new diskette for drive A: - вставьте новую дискету.
```

and strike any key when ready - когда будете готовы нажмите любую клавишу.

Символ <=Enter - в дальнейшем будет обозначать нажатие клавиши Enter. На дисковом A: установите дискету, которую нужно отформатировать. В процессе форматирования старая информация стирается. Поэтому необходимо убедиться, что на дискете нет полезной информации. Чтобы проверить дискету, можно остановить процесс форматирования именно в этой его точке, нажав одновременно клавиши "Ctrl" и "C" (Ctrl + C) или Esc. Чтобы продолжить форматирование дискеты, нажмите любую клавишу и ждите сообщения:

Formatting... - идет процесс форматирования.

Format complete - формат закончен.

362496 bytes total disk space - размер дискеты может быть и другим

362496 bytes available on disk - размер дискеты, доступный для использования

format another (Y/N)? N - форматировать еще? N - нет.

<= Enter

На экране появится:

A:\>

Дискета отформатирована и теперь ее можно снять с дисковода, и использовать для дальнейшей работы. При наличии на компьютере системного жесткого диска процедура форматирования выполняется так же, но системную дискету использовать не нужно.

Смена дисковода

Большинство команд MS - DOS производят запись, восстановление или корректировку данных на диске. Чтобы указать диск, на котором установлен обрабатываемый диск, в текст команды включается его буквенное обозначение. Если при введении команды обозначение дисковода опущено, то это означает, что обрабатываемый диск находится на дисковом, с которым вы работали ранее и который отражает стандартный системный запрос (A:\> или B:\>). Этот диск называется рабочим или текущим. При первоначальной загрузке системы, например, с дискеты на дисковом A:, статус рабо-

чего устройства получает именно этот дисковод. При наличии жесткого диска рабочим будет диск C:\> .

Для переключения системы на обслуживание другого дисковода наберите обозначение желаемого дисковода, затем двоеточие и нажмите клавишу Enter:

```
A:\> B:
```

```
<=Enter
```

```
B:\>
```

Двоеточие - это признак дисковода. Если вы забыли его набрать, то операционная система будет интерпретировать "B", как команду и попытается ее выполнить. Она просмотрит перечень зашитых в систему команд, и не обнаружив среди них команды B, выдаст сообщение об ошибке, а затем стандартный системный запрос:

```
A:\> B
```

```
Bad command or file name - неверная команда или имя файла.
```

```
A:\>
```

В этом случае надо повторить ввод команды, исправив допущенную ошибку.

ФАЙЛЫ

Одна из основных обязанностей MS - DOS - обслуживание (хранение, создание, уничтожение и т.п.) файлов. Файл в MS - DOS это набор взаимосвязанных данных, находящихся в специально отведенном месте постоянной памяти. В отличие от обычных документов, хранящихся в специальных архивных папках или сейфах, файлы MS - DOS хранятся на жестких дисках. При обработке файла компьютером он загружается в оперативную память машины. И загрузка в память, и хранение файлов на диске или дискете входят в функции операционной системы.

Идентификация файлов

Каждый файл в MS - DOS имеет конкретное имя. Имя файла может быть простым и сложным. Простое имя состоит только собственно из имени файла. Сложное имя состоит из основного, простого имени и расширения. По имени файл распознается операционной системой. Имена некоторых файлов, например, файлов на системной дискете, заранее определены и резервируются операционной системой. Имена остальных файлов назначаются пользователем. Обычно стараются придумать имя, отражающее назначение находящейся внутри файла информации.

Расширение используется для обозначения типа файла, например, текстовой (TXT) или файл данных (DAT). Оно может служить и для идентификации файлов с близкой по смыслу информацией, например, для разделения файлов с личной и служебной перепиской. При записи файла на диск его имя автоматически помещается в область памяти диска, называемую каталогом или директориумом. Чтобы узнать, какие файлы находятся на системной дискете, установите ее на дисковод A: и введите команду:

```
A:\>dir/w  
<=Enter
```

По команде DIR на экран выводится список имен имеющихся на дискете файлов. Отметим, что в последней ее строке указано количество файлов на диске, например, 33, а также объем свободной памяти в байтах - 84992. Имя первого файла - COMMAND COM, второго - DEBUG COM, третьего - EXE2BIN EXE и т.д.

Отметим, что каждое имя состоит из двух частей. Имена расположены по столбцам, и каждый столбец состоит из двух колонок. В первой из них указано собственно имя файла (его простое имя), во второй - расширение. Простое имя с расширением составляют слож-

ное имя файла. Так, при рассмотрении первого имени таблицы мы видим, что оно состоит из простого имени COMMAND и расширения COM.

Существуют определенные правила идентификации файлов. Имя файла должно быть уникальным в пределах диска. Другими словами, директорий диска не может включать два одинаковых имени (хотя, если имена сложные, то могут совпадать либо их простые имена, либо расширения). Например, нельзя назвать два файла именем LETTERS.PER - операционная система не будет знать, в каком случае вы обращаетесь к первому из них, а в каком ко второму. Однако, можно пользоваться одним и тем же простым именем файла, указывая разные расширения. Например, файл со служебной перепиской можно назвать LETTERS.BUS (Business), а файл с личной перепиской LETTERS.LET (Letters).

Некоторые имена резервируются MS - DOS исключительно для системного пользования. Это имена системных программ, обработчиков команд MS - DOS и стандартные имена внешних устройств (например, имя PRN для принтера).

Простое имя файла может включать до восьми символов, расширение - до трех. Расширение является необязательной составляющей имени файла. Если оно имеется, то от простого имени отделяется десятичной точкой. Если случайно вы ввели простое имя, включающее более восьми символов, то после восьмого символа автоматически вставляется десятичная точка, и три символа после нее интерпретируются операционной системой, как расширение, остальные символы отбрасываются. Если расширение содержит более трех символов, то, начиная с четвертого, символы игнорируются.

Для образования имени файла может использоваться только определенный набор символов - буквы алфавита цифры от 0 до 9 и специальные символы:

\$ # @ % () - { } ` ' _ ^ ~

Для образования имени или расширения файла нельзя использовать символы:

+ = [] : ; " , . / ? пробел, табуляция и управляющие символы.

Программно можно организовать файл, имя которого содержит один или несколько символов из последней группы, однако этот файл не будет обслуживаться MS - DOS. Строчные буквы в MS - DOS интерпретируются, как заглавные, поэтому имя команды и параметры командной строки (в частности, имя файла) могут набираться, как

маленькими, так и большими буквами (можно вводить и комбинацию из строчных и заглавных букв). Здесь имена и расширения файлов набираются строчными буквами, так как они чаще всего вводятся в командную строку операционной системы MS DOS на любом компьютере.

Спецификация файла

Чтобы операционная система могла обнаружить файл, ей нужно сообщить его имя. Кроме этого, необходимо указать диск, где находится файл. Для обозначения диска используется шифр устройства (дисковод), на котором он установлен. Шифр устройства представляет собой букву алфавита, за которой следует двоеточие.

Первому дисководу в системе присваивается шифр А.; второму В:. Первый дисковод для жесткого диска обычно помечается шифром С:. Шифр устройства, записанный в комбинации с именем файла, представляет собой спецификацию файла. В командную строку системы спецификация вводится в следующем формате:

"шифр устройства" "простое имя. расширение"

Например, спецификация файла с простым именем INSTRUCT и расширением TXT, находящегося на диске А:., выглядит следующим образом:

```
A:\>instruct.txt
```

Копирование файла

Копирование файлов - одна из самых распространенных операций. Опытный программист регулярно копирует все свои файлы, чтобы в случае повреждения любого из них, файл можно было восстановить. Процедуры копирования файлов различаются в зависимости от количества имеющихся в системе дисководов для гибких и жестких дисков.

Рассмотрим процедуру копирования для системы с двумя дисковыми при отсутствии жесткого диска. Если жесткий диск на компьютере имеется (а также имеется один дисковод для гибких дисков), то копирование выполняется точно так же, как для системы с двумя дисковыми.

При копировании дискета с исходным файлом называется исходной, а дискета для копии целевой - конечной. Этой терминологией пользуются и для обозначения дисководов (исходный и целевой).

Система с двумя дисковыми

Поместите исходную дискету (с копируемым файлом) на дискетод A:, а отформатированную целевую (дискету для копии) на дискетод B:. Предположим, что копируется файл INSTRUCT.TXT. Наберите команду COPY, спецификацию копируемого файла и спецификацию копии:

```
A:\>copy A:\instruct.txt B:\instruct.txt  
<=Enter
```

По окончании процесса копирования вы получите сообщение:

```
1 File(s) copied  
A:\>
```

При копировании файла на жесткий диск спецификация копии содержит шифр дискетода для жесткого диска. Например, при копировании файла INSTRUCT.TXT с диска A: на жесткий диск C: вводится команда:

```
A:\>copy A:\instruct.txt C:\instruct.txt  
<=Enter
```

Замещаемые символы

Замещаемые символы это специальные символы, используемые для обозначения одного или нескольких символов имени файла. В MS - DOS их два - звездочка (*) и вопросительный знак (?). Вопросительный знак заменяет только один символ имени или расширения, звездочка - группу символов. Замещаемые символы особенно удобны в применении с командами DIR, COPY, ERASE и RENAME, т.к. действие этих команд часто распространяется на несколько файлов.

Замещаемый символ "?"

Предположим, что на вашей дискете имеются следующие файлы:

```
last.txt  
list.txt  
lost.txt  
lust.txt
```

И вы хотите перенести или скопировать их на другую дискету. Файлы можно копировать двумя способами. Во - первых, по одному - четыре раза вводить команду COPY с соответствующей спецификацией. Во - вторых, одной командой, включив в спецификацию файла замещаемый символ. Если вы выбрали второй способ, введите команду:

```
A:\>copy A:\l?st.txt B:\l?st.txt  
<=Enter
```

Вопросительный знак во второй позиции имени файла указывает на то, что этот символ является замещаемым. То есть, при выполнении команды операционная система выберет с диска все файлы с расширением TXT и простым именем, первый символ которого "l", третий символ "s", и четвертый "t". На втором месте может находиться любая символ.

Замещаемый символ "*"

Символ "*" в имени файла означает, что в данной позиции и справа от нее (вплоть до окончания имени) могут находиться любые символы. Предположим, что вы хотите посмотреть, какие файлы на системной дискете имеют расширение .COM и начинаются с буквы F. Поместите дискету на дисковод A: и введите команду:

```
A:\>dir f*.com/w  
<=Enter
```

На экране появится сообщение:

```
format.com  
fdisk.com  
2 File(s) 84992 bytes free
```

Ключевой параметр /W означает, что на экран выводятся только имена файлов и директориев (дополнительная информация опущена). Спецификация файла может включать более одного замещаемого символа. Например, запись F*.COM эквивалентна записи F??????.COM. В следующем примере все файлы с дискеты A: копируются на дискету B:

```
A:\>copy A:\*.* B:\*.*  
<=Enter
```


ДИРЕКТОРИИ

Система управления файлами в MS - DOS построена на использовании данных директория (или каталога) диска. Директорий - это область памяти на диске, выделяемая в процессе его форматирования. Директорий представляет собой таблицу, куда заносятся данные о хранящихся на диске файлах. Каждому файлу в директории соответствует одна запись. Запись директория включает следующую информацию: полное имя файла (имя и расширение), дату и время его создания или последней корректировки, объем занимаемой памяти в байтах, а также некоторую дополнительную информацию, используемую при обслуживании файла операционной системой.

Все файлы расположены на одном уровне относительно директория. Такое расположение файлов называется неиерархическим. Ранние версии MS - DOS обеспечивают только неиерархическую структуру организации файлов. Новые версии отличается от предыдущих, прежде всего тем, что обеспечивает возможность объединения файлов в иерархические структуры. Разработка такого способа организации была обусловлена необходимостью хранения на диске большого количества файлов.

Иерархические системы файлов

Размещение на жестком диске сотен и даже тысяч файлов - обычное явление в практике пользователей персональных компьютеров. Для обслуживания такого количества файлов, естественно, требуется более гибкая система их организации и более надежная система поиска, чем в ранних версиях MS - DOS. DOS поздних версий (начиная с версии 2.0), обеспечивает возможность организации иерархических структур хранения файлов.

В таких структурах файлы объединяются в группы и располагаются на логических уровнях относительно главного директория диска. Главный директорий называется корневым, так как все файлы и директории расположены ниже него по уровню. Корневой директорий включает имена отдельных файлов и имена директориев, называемых поддиректориями. Поддиректории также могут включать имена отдельных файлов и поддиректориев следующего уровня. Объединяя несколько файлов в один директорий, мы сокращаем время поиска отдельного файла. Каждый последующий уровень иерархической системы через все предыдущие связан с корневым директором диска. При просмотре содержимого корневого директория по команде DIR на экран выводится информация, описывающая входящие в главный директорий файлы и директории. Запись, относящаяся к директорию, обязательно содержит метку <DIR>. В последней

строке экрана указывается общее количество файлов на диске (включая файлы в поддиректориях) и объем свободной памяти в байтах.

Число записей в корневом директории ограничено. Для односторонних дискет максимальное количество записей главного директория - 64 (файла и/или поддиректория), для стандартных двухсторонних дискет - 112. Для жестких дисков максимальное число записей директория зависит от количества мини - дисков и объема занимаемой им памяти. Далее термин "директорий" везде обозначает и корневой директорий, и поддиректорий, кроме специально оговариваемых случаев.

Организация директорий

Введите команду DIR для диска C:

```
C:\>dir  
<= Enter
```

На экране появится:

```
Volume in drive C: has no label Directory of C:\  
C:\COMMAND.COM 25307 1-10-99 10:00p1
```

```
File(s) 10510200 bytes free
```

Строка "Directory of C:\:" обозначает, что на экран выводится содержимое корневого директория диска C:. В данном случае в корневом директории содержится всего одна запись - для файла COMMAND.COM. Включением в корневой директорий нескольких поддиректориев мы расширим полученную таблицу. Воспользуемся для этого командой MKDIR (MaKe DIRectory). Организуем три директория: WRITE, PROGRAMS и BUSINESS. Прежде чем перейти к созданию директориев, познакомимся с правилами их идентификации. Имя поддиректория может содержать до восьми символов. Оно может дополняться расширением, состоящим из 1 - 3 символов. Поддиректорий должен иметь уникальное имя - не совпадающее с любым другим именем файла или директория на диске (в последних версиях системы имена файлов и директорий могут совпадать). Для образования имени директория используется тот же набор символов, что и для образования имени файла.

Для создания директория наберите команду MKDIR (или - сокращенно - MD) и спецификатор пути нового директория. Если новый директорий включается в рабочей директорий, то его спецификатор пути можно опустить. Рабочий директорий - это директорий, с

которым вы в данный момент работаете. Итак, для создания директория WRITE набираем команду MKDIR (или MD), обратный слеш (вместо имени корневого директория) и имя нового директория:

```
C:\>mkdir\write.  
< = Enter
```

Теперь новый поддиректорий организован. Таким же образом создаются поддиректории PROGRAMS и BUSINESS.

Для проверки введем команду DIR:

```
C:\>dir  
<=<Enter
```

```
Volume in drive C: has no label Directory of C:\  
C:\COMMAND.COM 25307 1-10-99 10:00p  
WRITE <DIR> 9-17-99 11:42a  
PROGRAMS <DIR> 9-17-99 11:43a  
BUSINESS <DIR> 9-17-99 11:44a
```

```
File(s) 10505080 bytes free
```

На вашей машине файл COMMAND.COM может занимать меньший или больший объем памяти. Маркеры времени и даты будут отличаться от приведенных. Смысл этого примера заключается в организации трех поддиректориев, вложенных в корневой директорий диска. Имена поддиректориев помечаются обозначением <DIR>. Отметим, что для самих имен этих директориев на диске отводится 5 120 байтов.

Изменение рабочего директория

Спецификатор пути описывает путь от текущего положения на диске до обрабатываемого директория. При отсутствии спецификатора в командной строке команда выполняется применительно к рабочему директорию. Для изменения рабочего директория служит команда CHDIR (CHange DIRectory). Наберите CHDIR (или CD) и спецификатор пути требуемого директория. В нашем примере рабочим директорием служил корневой директорий диска C:. При введении имени команды (без спецификатора пути) на экран выводится спецификатор пути рабочего директория. Введите CHDIR:

```
C:\>chdir  
< = Enter
```

На экране появится:

```
C:\>
```

Обратный слеш в последней строке обозначает, что конечной директорией является рабочим на диске C:. Чтобы назначить рабочим директориум поддиректорий WRITE, в командную строку операционной системы вводится спецификатор пути:

```
C:\>chdir\write  
< = Enter
```

Спецификатор пути может не включать имя первого директория, если он является рабочим. Так как, мы ввели команду, когда рабочим директориум являлся корневой директорией жесткого диска, то она могла быть представлена и так:

```
C:\>chdir write  
< = Enter
```

Теперь выполнен переход в поддиректорий и командная строка будет выглядеть:

```
C:\write>
```

Главная директория

Любой поддиректорий описан в некотором директории. Например, поддиректории WRITE, PROGRAMS и BUSINESS входят в корневой директорий диска. Директории, включающие поддиректории, называются родительскими по отношению к этим поддиректориям. Корневой директорий является родительским по отношению к поддиректориям WRITE, PROGRAMS и BUSINESS.

Обычно считают, что если директория находится в корневом каталоге, т.е. когда командная строка имеет вид:

```
C:\>
```

то такие директории называются именно директориями. Директории, входящие в них обычно называют поддиректориями или вложенными директориями. Можно сказать, что директории корневого каталога это директории первого уровня, а вложенные в них директории имеют второй уровень вложения - директории второго уровня.

Поддиректории, вложенные в директории второго уровня, называются директориями третьего уровня и т.д.

Большинство пользователей объединяют программные файлы в отдельный директорий, уменьшая вероятность совпадения их имени с именами пользовательских файлов. Однако, если файлы CONFIG.SYS и COMMAND.COM могут потребоваться в процессе работы MS - DOS, то они должны находиться в корневом директории диска.

Включение файла в директорий

Итак, рабочим директориум является теперь поддиректорий WRITE. Допустим, что в директорий WRITE нужно включить файл WP.EXE, который находится на диске A:. Установите дискету с файлом WP.EXE на дисковод A: и введите команду перехода на диск A: (предполагается, что системным диском является диск C: и загрузка компьютера происходит с него):

```
C:\write>A:  
< = Enter
```

Командная строка примет вид:

```
A:\>
```

Теперь введем команду копирования:

```
A:\>copy wp.exe C:  
< = Enter
```

В соответствии с введенной командой файл WP.EXE копируется на диск C:. Т.к. командная строка не включает спецификатор пути копируемого файла, то операционная система считает, что файл находится в рабочем директории дискеты A: (в данном случае, в ее корневом директории). Так как, в командной строке отсутствует спецификатор пути нового файла, то он будет записан в рабочий директорий диска C: (в данном случае, в директорий WRITE).

Поддиректории

Для включения нового поддиректория в поддиректорий WRITE можно воспользоваться командой MKDIR. Напомним, что в этом случае нужно набрать команду и спецификатор пути нового директо-

рия. Для включения поддиректория BOOK в поддиректорий WRITE, если он является рабочим наберите:

```
C:\write>mkdir book  
< = Enter
```

Запишем в имеющиеся поддиректории несколько файлов (т.е. организуем на диске несколько файлов и включим их в наши поддиректории). Допустим, что директорий LETTERS диска A: содержит файл HILBURN.DOC. Назначим этот директорий рабочим на диске A:

```
C:\>write>A:  
< = Enter
```

```
A:\>
```

Далее набираем:

```
A:\>cd letters  
< = Enter
```

На экране в командной строке появится сообщение:

```
A:\letters >
```

Теперь введем команду копирования:

```
A:\letters>copy hilburn.doc C:  
< = Enter
```

Файл будет скопирован в рабочий директорий диска C:, в данном случае WRITE. Следующим шагом скопируем файл START.DOC из корневого каталога диска A: и введем его в поддиректорий BOOK. Для этого назначим BOOK рабочим директорием. В данный момент рабочим директорием является LETTERS. Перейдем на диск C:.

```
A:\letters>C:  
< = Enter
```

Теперь мы возвращаемся в директорию WRITE. Спецификатор пути для перехода в BOOK должен выглядеть следующим образом:

```
C:\ write>cd book
```

< = Enter

Теперь START.DOC можно копировать - он будет включен в поддиректорий BOOK. Поместим дискету с этим файлом на устройство A: и введем команду:

C:\write\book>A:

< = Enter

A:.\>copy start.doc C:

< = Enter

Далее перейдем на диск C: и назначим рабочим директориум директорий WRITE. Это родитель поддиректории BOOK (т.е. нашего рабочего директория). Для того чтобы WRITE стал рабочим директориум, можно ввести команду:

C:\write\book.\>cd..

<=Enter

То же самое можно проделать и так:

C:\write\book>cd\write

<=Enter

На экране будет написано:

C:\write>

Посмотрим содержимое директория WRITE. Введем команду:

C:\>dir

< = Enter

Volume in drive C: has no label Directory of C:\write

.<DIR> 9-17-99 11:42a

..<DIR> 9-17-99 11:42a

WP EXE 72960 6-20-85 5:02p

BOOK <DIR> 8-17-99 2:00p5

File(s) 10262392 bytes free

Обратите внимание, что в первых двух строчках на месте имени директория находятся десятичные точки (.) и (..) Точка (.) в первой

строке обозначает рабочий директорий. Точки во второй строке (..) обозначают родитель рабочего директория. Последующие строки содержат описания файла WP.EXE и входящих в директорий WRITE поддиректориев.

Создадим теперь еще две директории. Сначала корневой директорий назначается рабочим:

```
C:\write>cd\  
<=Enter
```

В него включаются подчиненные директории (поддиректории) PROGRAMS и BUSINESS:

```
C:\>md programs  
<= Enter
```

```
C:\>md business  
<= Enter
```

Затем рабочим директорием назначается PROGRAMS. Для этого выполните команду:

```
C:\>cd programs  
< = Enter
```

Скопируем некоторые файлы в директорию PROGRAMS с диска A:. Дискета с файлами GWBASIC.EXE и LIFEX.BAS помещается на диск A:. После копирования файлы автоматически включаются в директорий PROGRAMS:

```
C:\programs>A:  
< = Enter
```

```
A:\>copy gwbasic.exe C:  
< = Enter
```

```
A:\>copy lifex.bas C:  
< = Enter
```

После этого рабочим директорием назначается BUSINESS. Для этого выполняются команды:

```
A:\>C:  
< = Enter
```



```
C:\programs>cd\business  
< = Enter
```

```
C:\business>A:  
< = Enter
```

На устройство A: установим дискету с файлами GWBASIC.EXE и RECORDS.BAS и перейдем на него. Файлы копируются и автоматически регистрируются в директории BUSINESS:

```
A:\>copy gwbasic.exe C:  
<=Enter
```

```
A:\>copy records.bas C:  
<=Enter
```

Дерево каталогов

MS - DOS предоставляют пользователю возможность объединения хранящихся на диске файлов в логическую структуру - дерево. "Дерево" представляет собой разветвленную сеть, сходящуюся к корневому директорию (корню дерева) диска. Каждой ветке, выходящей из корневого директория, соответствует в нем отдельная запись. Ветка представляет собой либо файл, либо поддиректорий (включающий группу взаимосвязанных файлов). Ветки, находящиеся на третьем уровне, также представляют собой либо файл, либо вложенный поддиректорий и выходят из поддиректориев второго уровня и т.д.

При такой организации обращение к описанному в конкретном директории файлу производится так, как будто на диске находятся только файлы этого директория, что существенно сокращает время поиска конкретного нужного в данный момент файла.

С увеличением количества файлов и поддиректориев усложняется и структура их организации. Для получения схемы иерархической структуры файлов служит команда TREE. Введите команду:

```
C:\>tree/f  
< = Enter
```

На экране в иерархическом порядке появится список имеющих на диске директориев. Ключевой параметр /F означает, что в иерархической последовательности будут отображены все имеющиеся на диске файлы. Отметим, что дополнительно указывается размер каждого файла в байтах.

Удаление поддиректория

Для удаления поддиректория служит команда RMDIR (Remove DIRectory) или сокращенно RD. Наберите RMDIR (или RD) и спецификатор пути удаляемого поддиректория. Помните, что перед удалением поддиректория необходимо удалить все имеющиеся там файлы. Предположим, нам нужно удалить поддиректорий BOOK. Сначала сотрем все хранящиеся там файлы. Для этого можно воспользоваться командой ERASE. После введения команды на экране появится вопрос:

```
C:\>erase\write\book\*.*  
< = Enter
```

```
*.* Are you sure? (Y/N) Y - Вы уверены ?  
< = Enter
```

При ответе "Yes" (Y), файлы уничтожаются, после чего директорий можно удалить командой:

```
C:\>rd\write\book  
< = Enter
```

Команда PATH

Выполняемый файл - это группа инструкций (или директив), в результате последовательного выполнения которых операционная система производит некоторое законченное задание. Таким файлом может являться прикладная программа (например, редактор), обработчик внешней команды MS - DOS или командный файл. При введении имени выполняемого файла MS - DOS выбирает его из рабочего директория. Если файла там нет, то имя директория, в котором находится файл, должно быть передано операционной системе. Эту функцию и выполняет команда PATH.

Наберите PATH и путь от рабочего директория к выполняемому файлу. При введении нескольких спецификаторов пути, они разделяются точкой с запятой (;). При введении имени команды PATH без параметров на экран выводятся спецификаторы пути, заданные при последнем использовании команды. Если после имени команды набрать только двоеточие, то введенные в последний раз спецификаторы отменяются.

Выше были приведены примеры организации иерархических структур файлов на жестком диске. Для простоты рассматривались структуры, состоящие всего из нескольких файлов и поддиректориев.

Однако, на практике жесткий диск предназначен для хранения большого количества файлов.

Предположим, что работая с редактором, вы составляете и записываете на диск документы разных видов. К примеру, вы одновременно составляете руководство по эксплуатации компьютера, пишете книгу, фиксируете вашу частную и деловую переписку, вводите справочную информацию и некоторые дополнительные данные. Пусть в процессе работы, вы ввели 30 глав некоторого руководства и 30 глав книги. Общее число оставшихся документов составляет 400. Эти документы делятся на следующие категории: частные (расширение файлов PER), деловые (BUS), справочные (SPV).

Каждая глава или документ представляют собой отдельный файл. Итак, всего у вас 460 файлов. Рассмотрим следующий вариант структуры поддиректорий. Создадим свой поддиректорий для каждой категории документов. Пусть все поддиректории выходят из корневого директория. В каждый поддиректорий запишем файлы соответствующего типа. И, дополнительно, в директорий WRITE запишем копию редактора. Предположим, вы сели к машине, решив поработать над книгой (файлы с расширением .DOC). Вам нужно быстро просмотреть несколько законченных глав, чтобы исправить последнюю незаконченную. На некоторый период времени вас не будет интересовать корреспонденция и все не связанные с книгой документы. Для начала, назначьте рабочим директорием диска C: поддиректорий NOVEL:

```
C:\>cd novel  
< = Enter
```

Теперь, чтобы получить список законченных глав романа, введите команду DIR\W. На экране появится содержимое поддиректория NOVEL. При желании книгу можно перенести на дискету B:

Наберите:

```
C:\novel>copy *.doc B:  
< = Enter
```

Копируются только главы книги. На остальные файлы действие команды копирования не распространяется. Из примера видно, что статус рабочего директория NOVEL "изолирует" хранящиеся в директории файлы от остальных файлов на диске.

Однако, рассматриваемая структура не гарантирует пользователю от некоторых затруднений. Например, при обращении к редактору вводится его имя: WP. Получив команду, MS - DOS, естественно, будет выбирать редактор из рабочего директория, в данном случае

NOVEL, и не обнаружит его. Чтобы стартовать редактор, нужно ввести дополнительную информацию о его местонахождении. Здесь - то и применяется команда PATH. Перед обращением к редактору введите:

```
C:\>path C:\write  
< = Enter
```

Теперь MS - DOS "знает", что если требуемого файла в рабочем директории нет, то его нужно искать в поддиректории WRITE.

Команда APPEND

Команда PATH обеспечивает доступ только к выполняемым файлам MS - DOS (т.е. к файлам с расширениями COM, EXE или BAT). Ее действие не распространяется на обычные файлы данных. Для обеспечения доступа к обычному файлу (например, к файлу типа HELP или DOC) служит команда APPEND. Формат команды совпадает с форматом команды PATH. Например, допускается следующая запись этой команды:

```
C:\>append C:\programs  
< = Enter
```

Теперь MS - DOS "знает", что если требуемого файла (как выполняемого, так и обычного) в рабочем директории нет, то он находится в поддиректории PROGRAMS.

КОМАНДНЫЕ ФАЙЛЫ

Командный файл - это текстовый файл, состоящий из группы команд MS - DOS. Правила идентификации командных файлов совпадают с общими правилами идентификации файлов. Единственное исключение - командный файл всегда записывается на диск с расширением .BAT (BATch). Обратиться к командному файлу крайне просто. Набирается команда старта - имя файла, и нажимается клавиша Enter. После введения команды файл выбирается из рабочего директория указанного или рабочего диска. Если в рабочем директории его нет, то поиск файла будет производиться в директориях, описанных командой PATH.

При нахождении файла, первая из его команд загружается в память, отображается на экране и выполняется. Этот процесс повторяется последовательно для всех команд файла (от первой до последней команды). Выполнение командного файла можно прервать в любой момент, нажав на клавиши Ctrl + Break. На экране появится сообщение:

```
Terminate batch job (Y/N)?_
```

При введении "Y" выполнение командного файла прерывается и на экран выводится стандартный системный запрос. При введении "N" отменяется выполнение только текущей команды.

Организация командного файла

Существует несколько способов организации командных файлов. Файл можно создать с помощью какого - либо текстового редактора или системной программы EDLIN. Еще один способ - введение команд непосредственно с клавиатуры. В этом случае ввод оформляется в виде файла, который записывается на диск. В MS - DOS клавиатура называется CON (CONsole) и для организации файла используется команда COPY CON:.

Например, для создания файла SAMPLE.BAT введите:

```
C:\>copy con: sample.bat  
< = Enter
```

После этого введите составляющие файл команды, нажимая клавишу Enter после каждой из них. Набрав последнюю команду, нажмите клавиши Ctrl + Z, а затем клавишу Enter.

На рабочем диске будет организован командный файл SAMPLE.BAT. Если в рабочей директории файл SAMPLE.BAT уже имеется, то старый файл заменяется новым. Командные файлы удобно использовать, скажем, для архивации системных файлов или для копирования, или стирания простых файлов данных.

Предположим, что вы в редакторе, который находится в директории NOVEL обрабатываете файл ESSAY.DOC. Его потеря крайне нежелательна, поэтому вы копируете файл после каждой корректировки. Организовав командный файл, вы избавитесь от этой работы - файл будет копироваться автоматически.

Пусть редактор в нашем примере называется WP.EXE и находится в каталоге WRITE. Для создания командного файла, находясь в корневом каталоге, введите:

```
C:\>copy con: backup.bat  
< = Enter
```

Далее набираем команды файла:

```
path=C:\write; C:\novel  
wp.exe  
copy essay.doc A:  
^Z  
< = Enter
```

При нажатии клавиш Ctrl + Z и Enter подключается требуемый дисковод и на диске C: организуется командный файл BACKUP.BAT. На экран выдается свидетельствующее об этом сообщение и стандартный системный запрос.

Чтобы запустить командный файл, введите его имя:

```
C:\>backup  
<= Enter
```

Перед выполнением все команды файла выводятся на экран. Поэтому сразу после введения BACKUP на экране появится:

```
C:\>wp.exe  
< = Enter
```

и будет запущен текстовый редактор. После выхода из редактора управление передается MS - DOS. На экране автоматически отображается следующая команда файла:

```
C:\>copy essay.doc A:  
< = Enter
```

Файл ESSAY.DOC копируется на дискету A: и выполнение командного файла завершается. На экран выводится стандартный системный запрос C:\> - MS - DOS готова принять новую команду.

Замещаемые параметры

Внутри командного файла допускается использование замещаемых параметров. Параметр - это символьная переменная, расположенная в командной строке после имени команды. Он содержит дополнительную информацию, необходимую операционной системе при обработке команды. Параметром, например, может быть имя файла, к которому относится действие команды.

Замещаемый параметр - это специальная переменная, которая в процессе выполнения команды подменяется обычным параметром (например, именем файла). В командном файле замещаемый параметр обозначается знаком процента (%) и цифрой от 0 до 9. Таким образом, командный файл может включать до десяти замещаемых параметров. Символьные переменные, предназначенные для подмены замещающего параметра, вводятся в командной строке при обращении к командному файлу. Набирается команда старта (имя файла) и список параметров в порядке, соответствующем последовательности замещаемых параметров внутри файла.

Параметры заменяются в порядке следования символьных переменных в командной строке. Первая переменная подменяет параметр %1, вторая - параметр %2 и т.д. Вместо замещаемого параметра %0 автоматически подставляется спецификация командного файла. При введении замещаемых параметров, командный файл становится более гибким. Предположим, что на диске имеется несколько файлов, которые нужно копировать после каждой корректировки. В рассмотренном выше примере командный файл использовался для копирования конкретного файла. Этим же командным файлом можно воспользоваться и для копирования любого файла. В этом случае вместо имени копируемого файла подставляется замещаемый параметр. Имя копируемого файла будет вводиться в командной строке при обращении к командному файлу. Назовем наш командный файл COPYALL.BAT. Введем команды:

```
C:\>copy con: copyall.bat  
< = Enter  
path=C:\write; C:\novel
```

```
wp.exe
copy %1 A:
^Z
<= Enter
```

При нажатии клавиш Ctrl + Z и Enter на диске C: создается командный файл COPYALL.BAT. При обращении к файлу набирается его имя и через пробел - имя копируемого файла (в нашем примере SHOPLIST.DOC). Введите команду:

```
C:\>copyall shoplist.doc
<= Enter
```

Сначала начинает работать редактор. После выхода из редактора управление возвращается MS - DOS и передается в командный файл. На экран выводится его вторая команда:

```
C:\>copy shoplist.doc A:
1 File(s) copied
```

MS - DOS автоматически подставила имя файла на место замещаемого параметра %1.

Организуем теперь командный файл DIFNUM.BAT, автоматически копирующий любой указанный файл и присваивающий копии любое указанное имя:

```
C:\>copy con: difnum.bat
<= Enter
```

```
path=C:\write; C:\novel
wp.exe
copy %1 A:%2
^Z
<= Enter
```

Для обращения к этому файлу наберите его имя, имя копируемого файла (например, NEW.DOC), и имя копии (OLD.DOC):

```
C:\>difnum new.doc old.doc
<= Enter
```

Сначала стартует редактор. После выхода из редактора управление возвращается MS - DOS, и на экране появляется следующая команда файла DIFNUM.BAT:


```
C:\>copy new.doc A:old.doc  
1 File(s) copied
```

Первое имя в командной строке (New) поставлено вместо замещающего параметра %1. Второе имя (Old) - вместо замещающего параметра %2.

Файл AUTOEXEC.BAT

Для удобства пользователя в системе предусмотрен специальный командный файл AUTOEXEC.BAT. Если он находится в корневом директории рабочего диска, то при загрузке MS - DOS автоматически выполняются его команды. Назначение файла - экономия времени (он состоит из обычно вводимых при загрузке команд). В общем случае AUTOEXEC.BAT определяет спецификаторы пути каждого директория, назначает рабочий директорий каждого диска, а также производит загрузку управляющей программы операционной системы. Если AUTOEXEC. BAT не включает команды TIME и DATE, то пользователю при загрузке с диска A: самому приходится вводить время и дату. Ниже приведен пример файла AUTOEXEC.BAT:

```
echo off  
prompt $p$g  
path C:;C:\windows;C:\dos;C:\program files;C:\aids;C:\arh  
append \e \x  
append C:\multimate;C:\edit  
set comspec=C:\command.com  
cd C:\book\newchp  
cd A:newchp  
rem Старт текстового редактора vplus  
vplus  
rem XCOPY - копирование всех новых и измененных файлов.  
xcopy * A: /m
```

Команда PAUSE

Если необходимо временное прерывание выполнения командного файла, то можно воспользоваться командой PAUSE. При введении этой команды откладывается выполнение текущей команды файла. На экране появляется сообщение:

Strike any key when ready... - нажмите любую клавишу, когда будете готовы.

Нажав любую клавишу (кроме Ctrl + C), вы вернете управление команде, выполнение которой было отложено. Нажав клавиши Ctrl + C, вы получите сообщение:

```
Abort batch job (Y/N)?_ - Прервать выполнение ?_
```

При введении "Y" управление возвращается MS - DOS и на экране появляется стандартный системный запрос C:\>. При введении "N" управление возвращается в командный файл и отложенная, приостановленная команда выполняется.

Следующий пример показывает, что PAUSE удобно использовать, если в процессе работы командного файла может возникнуть необходимость замены диска на рабочем дисковом. Командный файл COPYTWO.BAT выполняет двукратное копирование любого указанного файла. При этом первой и второй копиям могут присваиваться любые допустимые имена. Копии можно разнести на разные диски. После создания первой копии, выполнение командного файла прерывается и пользователь получает возможность поменять диски на устройстве A:

```
C:\>copy con: copytwo.bat
```

```
< = Enter
```

```
path=C:\write; C:\novel
```

```
copy %1 A:%2
```

```
pause
```

```
copy %1 A:%3
```

```
^Z
```

```
<= Enter
```

При обращении к этому файлу, наберите COPYTWO, через пробел - имя копируемого файла, имя первой и имя второй копии. Нажмите клавишу Enter. Процесс работы файла после его запуска будет фиксироваться на экране. Введите команду:

```
C:\>copytwo new.doc old1.doc old2.doc
```

```
< = Enter
```

Будут выполняться следующие команды файла:

```
C:\>copy new.doc A:old1.doc
```

```
1 File(s) copied
```

```
C:\>pause
```

Strike any key when ready... - нажмите любую клавишу, когда будете готовы.

Нажимаем клавишу Пробел:

```
C:\>copy new.doc A:old2.doc
1 File(s) copied
```

Еще раз отметим, что параметры в командной строке команды старта заменяют соответствующие замещаемые параметры внутри командного файла. Сначала копируется файл NEW.DOC. Затем команда PAUSE откладывает выполнение следующей команды COPY. Пользователь может заменить диск на дисковом A:. Нажатием любой клавиши управление возвращается отложенной команде и файл NEW.DOC копируется еще раз. Созданием второй копии завершается работа командного файла.

Команда PAUSE может служить для вывода сообщений на экран дисплея. В этом случае после имени команды через пробел набирается текстовая строка длиной до 121 символа. В процессе работы командного файла текст выводится на экран:

```
C:\>copy con: copytwo.bat
<= Enter

path=C:\write; C:\novel
copy %1 A:%2
pause put disk number2 in drive A:
copy %1 A:%3
^Z
<= Enter
```

Теперь, в отличие от предыдущего случая, при выполнении команды PAUSE на экран выводится сообщение:

```
C:\>copy new.doc A:old1.doc

1 File(s) copied
```

C:\>pause put disk number 2 in drive A: - поместите диск номер 2 в дисковод A: .

Strike any key when ready...

Нажимаем, например, клавишу с цифрой 5:

```
C:\>copy new.doc A:old2.doc
```

```
1 File(s) copied
```

Команда REM

Команда REM (REMark) служит исключительно для вывода сообщений на экран дисплея в процессе работы командного файла. Она вводится в командный файл вместе с требуемым сообщением. Длина сообщения не может превышать 123 символов. Введем, например, следующее:

```
C:\>copy con: copytwo.bat
```

```
< = Enter
```

```
rem making copy number 1 - создается копия номер 1.
```

```
copy %1 A:%2
```

```
pause put disk number2 in drive A: - поместите диск 2 в дисковод
```

A: .

```
rem making copy number 2 - создается копия номер 2.
```

```
copy %1 A:%3
```

```
^Z
```

```
<=Enter
```

Команда REM дает возможность пользователю следить за ходом выполнения командного файла после его запуска:

```
C:\>rem making copy number 1
```

```
C:\>copy new.doc A:old1.doc
```

```
1 File(s) copied
```

```
C:\>pause put disk number2 in drive A:
```

```
Strike any key when ready...
```

Нажимаем клавишу 5:

```
C:\>rem making copy number 2
```

```
C:\>copy new.doc A:old2.doc
```

```
1 File(s) copied
```

Команда ECHO

Как уже отмечалось, команды командного файла непосредственно перед выполнением отображаются на экране. Команда ECHO позволяет управлять выводом этих команд. Команда ECHO вводится в командный файл в следующем виде: набирается имя команды - ECHO и режим ее работы - ON или OFF.

При введении ECHO ON команды файла отображаются в обычном (описанном выше) режиме. ECHO OFF подавляет выдачу команд на экран, включая и команды REM. Однако на экране будут появляться все сообщения, генерируемые системой в процессе работы командного файла. При отсутствии ECHO в командном файле по умолчанию работает режим ON. Если произошло прерывание выполнения командного файла (аварийное или нормальное), то команда ECHO автоматически переходит в режим ON. При введении в командную строку имени команды ECHO на экране отображается режим ее работы в текущий момент времени. Рассмотрим работу команды на примере командного файла:

```
C:\>copy con: example.bat
```

```
<= Enter
```

```
rem - это сообщение выводится на экран, т.к. ECHO в режиме ON
```

```
echo off - ECHO переходит в режим OFF
```

```
rem - это сообщение не выводится на экран, т.к. ECHO в режиме OFF
```

```
echo - выводится режим ECHO
```

```
echo on - ECHO переходит в режим ON
```

```
rem ECHO опять в режиме ON
```

```
echo - выводится режим ECHO
```

```
^Z
```

```
<= Enter
```

Запускаем командный файл:

```
C:\>example
```

```
<= Enter
```

На экране видим:

```
C:\>rem это сообщение выводится на экран, т.к. ECHO в режиме ON
```

```
C:\>echo off - ECHO переходит в режим OFF.
```

```
echo is off
```

```
C:\>rem ECHO опять в режиме ON/  
C:\>echo
```

echo is on

Первая команда REM выводится на экран, т.к. ECHO по умолчанию находится в режиме ON. Первая команда ECHO переводит экран в режим OFF, поэтому следующая команда REM на экране не появляется. Вторая команда ECHO отображает текущий режим ECHO - OFF. Третья команда ECHO переводит режим экрана в состояние ON и комментарии оператора REM будут выведены на экран.

Если ECHO вводится в командный файл вместе с сообщением, то оно появится на экране вне зависимости от режима работы команды:

```
C:\>copy con: example2.bat  
<= Enter
```

```
echo off  
rem данное сообщение не выводится  
echo а это выведется на экран  
echo on  
rem сообщение появится на экране  
echo и это тоже  
^Z  
<= Enter
```

Запускаем созданный файл:

```
C:\>example2  
<= Enter
```

На экране видим:

```
C:\>echo off
```

а это выведется на экран

```
C:\>rem сообщение появится на экране  
C:\>echo и это тоже
```

Первая команда переводит ECHO в режим OFF. Поэтому первая REM не выводится на экран. Следующая команда - ECHO. Т.к. ECHO находится в режиме OFF, то и эта команда не появится на экране,

однако сообщение, находящееся в области действия команды будет отображено. Четвертой командой ECHO переводится в режим ON и поэтому следующая за ней команда REM появляется на экране. В последней команде файла ECHO сообщение появится на экране.

Использование ECHO для вывода на экран пустой строки

Текст на экране воспринимается легче, когда он разбит пустыми строками, отделяющими один смысловой абзац от другого. Для введения пустой строки можно воспользоваться командой ECHO. Команда ECHO. (ECHO и десятичная точка) выводит на экран пустую строку.

Подавление выдачи на экран ECHO OFF

MS - DOS позволяет отменить вывод строки командного файла на экран дисплея. В этом случае первым символом строки должен быть символ @. Этим пользуются, скажем, для подавления изображения команды ECHO OFF на экране, если она является первой командой командного файла. В процессе работы файла, первой строкой которого является

```
@echo off
```

на экране не появится ни одной команды. При отсутствии символа @ на экране отображается команда ECHO OFF, а остальные команды не отображаются.

Команда GOTO

Команда служит для передачи управления в границах области действия командного файла. Метка строки командного файла состоит из двоеточия (:), за которым следует от 1 до 8 символов. Введите, например, следующее:

```
C:\>copy con: example3.bat  
< = Enter
```

```
rem Это первая строчка  
rem Это вторая строчка  
goto four  
rem Это третья строчка  
:four
```

```
rem Это четвертая строка
^Z
<= Enter
```

Запускаем, созданный файл:

```
C:\>example3
<= Enter
```

На экран выводится:

```
C:\>rem это первая строка
C:\>rem это вторая строка
C:\>goto four
C:\>rem это четвертая строка
```

Сначала выполняются первые две команды. Затем - команда под меткой `four`. Далее продолжается последовательное выполнение команд до конца файла.

Метка `GOTO` может представлять собой замещаемую переменную. В этом случае позволяет переходить на метку, заданную командой старта командного файла. Рассмотрим пример:

```
C>copy con: example3.bat
< = Enter
goto %1
:one
rem this is one - Это первый
goto finish
:two
rem this is two - Это второй
goto finish
:three
rem this is three - Это третий
:finish
^Z
<= Enter
```

Запуск файла:

```
C:\>example3 three
<= Enter
```

На экране видим:


```
C:\>goto three
C:\>rem this is three - Это третий
```

Команда старта включает параметр three. При выполнении первой команды командного файла, замещаемая переменная %1 заменяется на three. Далее выполняется команда помеченная :three - команда REM (this is three). Последняя строка командного файла - это метка строки. В процессе работы командного файла метки на экран не выводятся.

Команда IF

Обычно команда IF используется для выделения команд, выполняющихся в случае, если выполняются некоторые заданные условия. Существует три типа условий, которые могут тестироваться командой IF:

1. IF EXIST команда
2. IF строка1 = строке2
3. IF ERRORLEVEL N

Первый тип условий называется условием типа EXIST. По этому условию производится проверка на существование указанного файла. Если файл существует, то условие считается выполненным и выполняется заданная команда. Рассмотрим команду:

```
if exist somefile.dat type somefile.dat
```

При ее выполнении производится проверка на наличие файла SOMEFILE.DAT на рабочем диске. Если файл существует, то выполняется команда TYPE. В противном случае команда TYPE пропускается и выполняется следующая по порядку команда. Команда IF EXIST может использоваться для проверки наличия файлов не только на рабочем диске. В этом случае перед именем файла необходимо указать шифр дисковод (например, A: или B:). Однако, необходимо помнить, что проверка производится только для файлов в рабочих директориях. Чтобы произвести проверку файлов в другом директории, нужно назначить его рабочим.

Второй тип условия - проверка идентичности двух символьных строк:

```
if string1 == string2
```

Рассмотрим командный файл:

```
C>copy con: example4.bat
<= Enter

echo off
if %1==roses goto roses
if %1==candy goto candy
if %1==perfume goto perfume
goto finish
:roses
echo вы послали розы
goto finish
:candy
echo вы послали конфеты
goto finish
:perfume
echo вы послали духи
:finish
^Z
<= Enter
```

После запуска такого файла:

```
C:\>example4 perfume
<= Enter
```

На экране видим:

```
C:\> echo off
```

вы послали духи

Каждая команда IF производит сравнение замещаемого параметра с символьной переменной. Отметим, что при сравнении знак тождества обозначается двойным знаком равенства (==). В командную строку команды старта вводится параметр - символьная переменная, которая в процессе работы командного файла подставляется вместо замещаемого параметра %1. Если результат сравнения положительный, то выполняется команда области действия данного IF. В нашем примере это команда перехода на метку строки perfume. В противном случае выполняется следующая по порядку команда.

Третий вариант команды:

```
IF ERRORLEVEL N
```

ERRORLEVEL - это имя переменной операционной системы, которая контролирует работу команд MS - DOS. Большинство команд, если при их выполнении произошла ошибка, присваивают переменной ERRORLEVEL некоторое значение. Это значение (код ошибки) устанавливается в зависимости от типа ошибки. Команда:

IF ERRORLEVEL N command

означает, что если значение ERRORLEVEL больше или равно значению N, то выполняется команда COMMAND. В противном случае выполняется следующая за IF команда.

Команда IF NOT

Команда IF NOT служит для проверки некоторого условия, подобного условиям команды IF. Команда в области действия оператора IF NOT выполняется, если результат проверки отрицательный. Рассмотрим далее следующую команду:

if not exist somefile.bak copy somefile.txt somefile.bak

Эта команда производит проверку наличия файла на диске. Если файл не найден, то выполняется команда в области действия оператора IF NOT. Эту команду можно использовать для проверки любого допустимого условия.

Команда FOR

По команде FOR другие команды командного файла выполняются несколько раз подряд - один раз для каждого из группы заданных параметров. Команда FOR имеет усложненный синтаксис, поэтому для начала рассмотрим пример:

for %%A IN (file1 file2 file3) DO del %%A

Данное командное предложение начинается командой FOR. За ней следует пустая переменная %%A. Пустая переменная в командном файле обозначается двойным знаком процента (%%). Затем следует слово IN, которое обязательно должно набираться заглавными буквами. После IN в скобках перечисляются параметры, впоследствии обрабатываемые следующей за ними командой DEL. Эта группа параметров обычно состоит из имен файлов, разделенных пробелом. Группа параметров замыкается словом DO, также набранным заглавными буквами. Затем набирается имя команды - в нашем примере это

DEL %%A. Она выполняется три раза, последовательно уничтожая файлы file1, file2 и file3. Команда FOR оказывается незаменимой, когда одной командой требуется обработать несколько файлов, имена, которых нельзя объединить замещаемыми символами. Предположим, что на диске имеются файлы: EXAMPLE.BAT, PROGRAM.TXT и LETTER, и каждый из них необходимо распечатать. Можно ввести команду COPY EXAMPLE.BAT PRN и ждать, пока файл распечатается, затем ввести ту же команду для файла PROGRAM.TXT, опять подождать и ввести ту же команду для третьего файла. При этом много времени теряется на ожидание.

Команда FOR, введенная в командный файл, избавит вас от потери времени:

```
for %%A IN (example.bat program.txt letter) DO copy %%A prn
```

Три файла распечатываются одной командой. Количество команд FOR в командном файле не ограничено. Команда может работать, и как стандартная команда MS - DOS. Если она используется вне командного файла, то пустой переменной должен предшествовать только один знак процента %. Любой файл, используемый в качестве параметра команды FOR, должен находиться в рабочем директории рабочего диска вашего компьютера.

Команда CALL

В настоящее время концепция модульного программирования получила очень широкое распространение. Принцип модульного программирования состоит в разбиении большой прикладной программы на несколько отдельных подпрограмм - модулей, таким образом, чтобы каждый модуль выполнял некоторую отдельную функцию (например, обработку файла, копирование файла и т.д.). Программисты пытаются писать универсальные модули, т.е. такие, к которым могли бы обращаться самые разнообразные программы. Такой способ избавляет пользователя от необходимости "изобретения колеса". Еще одно преимущество модульного программирования - удобство и простота отладки маленькой программы по сравнению с большой. Обращение к модулю осуществляется с помощью команды CALL. Программирование командных файлов - первый шаг на пути создания универсальных командных модулей. Однако MS - DOS не предусматривает простого способа обращения к командному модулю.

Рассмотрим, например, принцип работы следующих командных файлов:

```
C>copy con: one.bat
```

<= Enter

echo starting one - старт первого.

two

echo ending one - конец первого.

^Z

<= Enter

и

C:\> copy con: two.bat

<= Enter

echo starting two - старт второго.

echo ending two - конец второго.

^Z

<= Enter

При запуске файла ONE.BAT происходит следующее:

C:\>one

<= Enter

C:\>echo starting one

starting one

C:\>two

C>echo starting two

starting two

C:\>echo ending two

ending two

C:\>

Команда ECHO файла ONE.BAT выводит на экран стартовое сообщение, затем производится обращение к файлу TWO. BAT. В процессе работы этого файла на экране появляются сообщения начала и окончания работы. Однако, в большинстве случаев после этого

управление возвращается MS - DOS, а не в файл ONE.BAT. Его конечное сообщение так и не появится на экране.

Этой ситуации можно избежать, немного изменив командный файл ONE.BAT. А именно, для обращения к файлу TWO.BAT вводится команда CALL:

```
C>copy con: one.bat  
< = Enter
```

```
echo starting one  
call two  
echo ending one  
^Z  
<= Enter
```

Теперь после завершения TWO.BAT управление передается следующей после CALL команде файла ONE.BAT:

```
C:\>one  
< = Enter
```

```
C:\>echo starting one  
starting one
```

```
C:\>call two  
C>echo starting two  
starting two
```

```
C:\>echo ending two  
ending two
```

```
C:\>echo ending one  
ending one
```

Как теперь видно, после выполнения второго файла управление было передано первому.

НЕКОТОРЫЕ КОМАНДЫ СИСТЕМЫ DOS

MS - DOS работает по принципу "команда - действие системы". Другими словами, MS - DOS обрабатывает каждую введенную команду, производя операции, необходимые для ее выполнения. Данная глава книги посвящена рассмотрению некоторых команд MS - DOS. В процессе изложения материала перечисляются характеристики, рассматриваются способы применения и формат команды, а также приводятся реальные примеры применения каждой команды.

Способы задания команд

Каждый раздел главы начинается с заголовка, аналогичного следующему:

CLS - внутренняя

Первая часть заголовка - сокращенное название команды (как правило, это начальные заглавные буквы ее полного имени). Вторая часть - характеристика команды (команда может быть внутренней или внешней).

Внутренние команды встроены в операционную систему. При загрузке системы они автоматически переносятся в память. Внутренняя команда выполняется непосредственно после ее введения. Внутренними командами можно оперировать свободно, не производя дополнительных обращений к системному диску. Примерами внутренних команд MS - DOS являются команды: CHDIR, COPY, CLS, TIME и TYPE.

Внешние команды хранятся на системном диске в виде файлов и загружаются в память по мере необходимости (т.е. после введения команды). Примерами внешних команд могут служить: DISKCOPY, CHKDSK, FORMAT, PRINT и RECOVER. При введении внешней команды, сначала загружается в память требуемый файл и, только после этого, производятся операции обработки. Поэтому перед введением команды на рабочем дисковом устройстве устанавливается диск с ее обработчиком. Если обработчик команды находится в рабочей директории рабочего диска вашего компьютера, то команду (и ее параметры) можно смело вводить:

```
C:\>chkdsk A:
```

```
< = Enter
```

Если обработчик внешней команды находится в рабочем директории нерабочего диска, то перед именем команды набирается шифр устройства, на котором находится диск:

```
A:\>C:\chdisk B:  
< = Enter
```

Если обработчик находится в нерабочем директории, то перед именем команды набирается его спецификатор пути:

```
A:\>C:\dos\chdisk B:  
< = Enter
```

Ввести группу спецификаторов пути для организации поиска обработчика внешней команды можно с помощью команды PATH. После введения PATH отпадает необходимость указывать эти спецификаторы в командой строке.

Формат команды

Если на экране находится стандартный системный запрос (A:\>, B:\>, C:\>), значит можно вводить команду. Однако, вводимая запись должна соответствовать принятому в системе формату. Формат команды CLS (Clear Screen) соответствует записи CLS. Это означает, что для введения команды нужно набрать символы CLS и нажать клавишу Enter.

Напомним, что команду можно набрать, как заглавными, так и строчными буквами - операционная система автоматически переводит любые буквы в заглавные:

```
C:\>cls  
<= Enter
```

Большинство команд MS - DOS требует введения дополнительных параметров. Параметры содержат информацию о назначении команды и о способах ее обработки операционной системой. Формат включает все возможные параметры команды. Например, формат команды SYS (SYStem files) следующий: SYS D:. "D:" - это параметр, указывающий на необходимость введения шифра устройства. Если при введении команды шифр устройства будет опущен, то по умолчанию MS - DOS обработает диск на рабочем дисковом. Предположим, вам требуется диск A:. Тогда командная строка будет выглядеть следующим образом:


```
C:\>sys A:  
<= Enter
```

Некоторые параметры являются необязательными. При описании формата команды они заключаются в квадратные скобки, например, [LIKE THIS]. Рассмотрим формат команды VOL (VOLume). Он выглядит: VOL [d:]. Так как, параметр "d:" находится в квадратных скобках, то при введении команды его можно опустить.

В MS - DOS команда с параметром, отличается от команды без параметра. В этих двух случаях она по - разному интерпретируется и обрабатывается. В формате многих команд встречается слово PATH. Этот параметр обозначает спецификатор пути. Реально вместо слова PATH указываются составляющие спецификатору имена директориев. Имена разделяются обратным слэшем (\). Рассмотрим формат команды MKDIR (MaKe DIRectory):

```
mkdir [d:] path
```

Видно, что при введении команды шифр устройства можно опустить - параметр "d:" находится в квадратных скобках. Однако, спецификатор пути - параметр обязательный и вводить его необходимо. К часто встречающимся параметрам относятся FILENAME и .EXT. Параметр FILENAME - это простое имя файла (длиной до 8 символов), параметр .EXT - расширение (до 3 символов). Если имя файла не имеет расширения, то параметр .EXT не указывается. Рассмотрим формат команды TYPE:

```
type [d:][path]filename[.ext]
```

Можно видеть, что при введении команды необходимо указать простое имя файла. Необязательными параметрами являются шифр устройства, спецификатор пути и расширение. При введении некоторых команд необходимо указывать файлы типа SOURCE (источник - исходный файл) и TARGET (целевой - конечный файл). Файл SOURCE содержит входные данные для обработчика команды, файл TARGET - выходные данные обработчика (т.е. результат). С помощью замещаемых параметров можно задать несколько файлов SOURCE и TARGET.

Синтаксис команды

Все знаки препинания, встречающиеся в формате команды, необходимо вводить так, как они приведены в книге. При введении команды могут использоваться следующие знаки препинания: запятая,

двоеточие, точка с запятой, знак вопроса, слэш и кавычки. Также в командную строку необходимо включать круглые скобки и знак плюс (+). Параметры, разделенные вертикальной чертой, являются альтернативными. Например, ON|OFF означает, что ввести нужно либо ON, либо OFF. Многоточие означает повторяемость параметра, т.е. при введении команды параметр, после которого в формате команды стоит многоточие, можно указывать несколько раз подряд. Параметры, приведенные в квадратных скобках указывать необязательно.

Замещаемые параметры

Большинство команд MS - DOS допускают использование замещаемых символов в параметрах FILENAME и .EXT. Эти параметры могут включать один или несколько замещаемых символов ("?" или "*"). Для каждого файла с подходящим именем или расширением команда выполняется один раз.

Ключевые параметры

Еще один тип параметров - ключевые. Эти параметры определяют способ выполнения команды. Ключевой параметр обозначается слэшем (/) - сначала набирается слэш, затем - буква или число. Рассмотрим формат команды DIR (DIRectory):

```
dir[d:][path][filename][.ext]][/p][/w]
```

Команда может содержать два ключевых параметра - /P или /W, и оба являются необязательными. При введении /w содержимое директории располагается во всю ширину экрана. Параметр /P приостанавливает вывод записей директории при заполнении экрана. При рассмотрении конкретной команды перечисляются все ее ключевые параметры.

Список некоторых команд системы

APPEND - внешняя

ФУНКЦИЯ: Обеспечить доступ к невыполняемым файлам в разных каталогах.

ФОРМАТ:

```
append [/x][/e]
```

```
append d: path[:d:]path...
```

ПРИМЕРЫ:

append /x/e: append C:\word; C:\turbo

Команда APPEND является очень нужным дополнением команды PATH. PATH определяет спецификатор пути для поиска конкретного файла. К сожалению, PATH обеспечивает доступ только к файлам с расширениями EXE, COM или BAT (к выполняемым файлам). Команда APPEND устраняет этот недостаток.

Она обеспечивает доступ ко всем невыполняемым файлам указанного директория. Большинство программ (особенно это касается текстовых редакторов) состоят из выполняемого файла (собственно программы) и нескольких невыполняемых (файлов данных).

Примером невыполняемых файлов могут служить файлы с расширением .HLP. Параметры команды APPEND - спецификаторы пути. В командной строке они разделяются точкой с запятой. Спецификатор пути может быть длиной до 128 символов.

Вывод на экран спецификатора

Для получения на экране ранее введенного спецификатора пути вводится одна из следующих команд.

На клавиатуре был задан путь:

```
C:\>append C:\word  
<= Enter
```

Вводим с клавиатуры только команду:

```
C:\>append  
<= Enter
```

На экране получаем результат:

```
append=C:\word
```

Точно так же было задано:

```
C:\>append C:\word; C:\turbo  
<= Enter
```

Вводим:

```
C:\>append  
<= Enter
```

На экране видим:

```
append=C:\word; C:\turbo
```

Отмена введенного спецификатора

Если в командной строке ввести точку с запятой (;), то ранее введенный спецификатор отменяется. Вводим с клавиатуры:

```
C:\>append  
<= Enter
```

На экране получаем:

```
append=C:\word;C:\turbo
```

Вводим на клавиатуре команду отмены:

```
C:\>append;  
<= Enter
```

Вводим для проверки:

```
C:\>append  
<= Enter
```

На экране будет показано:

```
No Append
```

После отмены прежнего спецификатора в любой момент можно ввести новый спецификатор пути.

BACKUP - внешняя

ФУНКЦИЯ: Копирование файлов.

ФОРМАТ:

```
backup d:[path][filename[.ext]]d:[/s][/m][/d:mm/dd/yy]  
[/t:bb:mm:ss][/f][/l[:[d:][path]filename[.ext]]]
```

ПРИМЕРЫ:

```
backup C:A:
```

```
backup C:*.*txt A:
```

```
backup C:\book\*.*txt A:
```

Команда BACKUP копирует файлы с жесткого диска на гибкую дискету. В MS - DOS версий 3.X и выше копирование можно производить на диск любого типа. Копии, полученные с помощью BACKUP, представляют собой архивные файлы и используются, главным образом, для хранения информации. Доступ к архивным файлам осуществляется с помощью команды RESTORE. Таким образом, команды BACKUP и RESTORE служат для создания архивных копий наиболее важных файлов. При копировании файлы отбираются по следующим признакам: "имя файла", "расширение", "маркер даты" или "маркер времени". Таким образом, можно выбрать только те файлы, которые, например, подвергались обработке с момента создания последней копии. Существует два способа хранения файлов, организованных с помощью BACKUP:

1. На гибкой дискете они записываются в корневой директорий.
2. На жестком диске - помещаются в поддиректорий BACKUP.

Первый способ обеспечивает некоторую экономию дисковой памяти компьютера. BACKUP является внешней командой DOS. Это означает, что при обращении к ней на рабочем диске должен находиться файл BACKUP.COM. В примерах данного раздела предполагается, что BACKUP.COM находится в корневом директории диска C.

Копирование файла

При копировании файла с жесткого диска набирается: имя команды - BACKUP; спецификация файла; шифр устройства (например, A:), куда помещается копия (гибкая дискета), и, наконец, один из четырех ключевых параметров. Спецификация файла включает шифр устройства, спецификатор пути и имя файла с расширением. Если в спецификации отсутствует шифр устройства и спецификатор пути, то принимается, что копируемый файл находится в рабочей директории рабочего диска. Имя и расширение файла могут включать замещаемые символы. В этом случае копируются все файлы, отвечающие указанной в командной строке спецификации.

Скопируем файл LOTSA.DAT с жесткого диска компьютера C: на гибкую дискету A:

```
C:\>backup C: lotsa.dat A:  
<= Enter
```

MS - DOS выдаст предупреждение:

Insert backup diskette 01 in drive A: - вставьте дискету для копирования в дисковод A: .

Warning! Diskette files will be erased - Внимание. Файлы на дискете будут стерты.

Strike any key when ready - нажмите любую клавишу, когда будете готовы.

Если в командной строке отсутствует ключевой параметр /A, то перед операцией копирования все файлы на дискете уничтожаются. Выданное предупреждение обеспечивает возможность замены дискеты (в случае, если на ней находится полезная информация). Еще раз проверьте содержимое дискеты, если требуется, замените ее и нажмите любую клавишу.

При завершении процесса копирования на экране появится следующее сообщение:

```
*** Backing up files to diskette 01 *** - Файлы скопированы на
дискету
A:\lotsa.dat
```

Копирование директории

Если в командной строке BACKUP не указано имя файла, то копируются все файлы заданного директория. В следующем примере производится копирование всех файлов поддиректория SUBDIR1. Отметим, что имя каждого скопированного файла последовательно включается в список на экране дисплея.

```
C:\>backup C:\subdir1 A:
<= Enter
```

```
Insert backup diskette 01 in drive A:
Warning! Diskette files will be erased
Strike any key when ready
```

Копирование всех файлов заданного директория, включая файлы вложенных поддиректорий, производится при включении в командную строку ключевого параметра /S. В следующем примере производится копирование всех файлов директория SUBDIR2, а также всех файлов вложенных в него поддиректорий:

```
C:\>backup C:\subdir2 A:/s
<= Enter
```

Insert backup diskette 01 in drive A:
Warning! Diskette files will be erased
Strike any key when ready

Копирование обработанных файлов

Чтобы произвести копирование файлов, подвергавшихся обработке с момента создания последней копии, в командную строку BACKUP добавляется ключевой параметр /M. Благодаря этой особенности команды, пользователь экономит время и дисковую память. При введении параметра /M команда выбирают лишь те файлы, которые копировать действительно необходимо.

Предположим, что ваши файлы, организованные с помощью текстового редактора, хранятся на жестком диске, и все они записаны с расширением .DOC. Если файлов много (к примеру, несколько сотен), то почти невозможно уследить, в какие из них вносились изменения с момента создания архивной копии на Дискете.

Команда BACKUP/M сделает эту работу за вас. Все, что от вас требуется, - это введение следующей команды. Она вводится по завершению каждого сеанса работы в редакторе:

```
C:\>backup *.doc A:/m  
<= Enter
```

Любой обрабатывавшийся файл будет заново откопирован.

Копирование файлов по дате

Ключевой параметр /D вводится при необходимости копирования файлов, которые были организованы или обработаны в определенный день, или позже этого дня. Следующая команда BACKUP копирует файлы корневого директория, организованные или обработанные позже 21 марта 2001 года:

```
C:\>backup C:\ A:/d:03-21-01  
<= Enter
```

Копирование файлов по времени

Ключевой параметр /T вводится при необходимости копирования файлов, которые были организованы или обработаны позже определенного времени дня. Следующая команда BACKUP копирует файлы корневого директория, организованные или обработанные позже 3.00 часов дня. Копии помещаются на дискету A.:

```
C:\>backup C:\*. * A:/t:15:00:00  
<= Enter
```

Оставшиеся ключевые параметры BACKUP

Параметр /F позволяет копировать файлы на заранее не отформатированную дискету. Дискета форматируется в процессе копирования. При выполнении этой операции файл FORMAT.COM должен быть доступен операционной системе.

При введении параметра /L формируется журнал регистрации работы BACKUP (LOG FILE). Журнал содержит список скопированных файлов, а также время и дату их копирования. Его можно использовать в качестве протокола работы BACKUP. При организации журнала можно ввести его спецификацию: шифр устройства, спецификатор пути и имя файла. По умолчанию в корневом директории исходного диска (на котором находится копируемые файлы) организуется файл BACKUP.LOG.

BUFFERS - внутренняя

ФУНКЦИЯ: Определить количество буферов диска, выделяемых после загрузки MS - DOS.

ФОРМАТ:

buffers=xx

ЗАМЕЧАНИЕ: xx - это число в интервале значений 1 - 99. Команду BUFFERS можно использовать только в качестве оператора файла конфигурации CONFIG.SYS.

ПРИМЕР:

buffers=5

Буфер диска - это область памяти, выделяемая для хранения данных, участвующих в дисковых операциях ("запись на диск" или "чтение с диска"). Буфер является областью связи работающей программы с диском. Пересылка данных из буфера на диск и обратно осуществляется блоками по 512 байт. Чтобы проиллюстрировать функции буфера, рассмотрим, что произойдет, если программе потребуется хранящаяся на диске 128 - байтовая запись. Операционная система считает с диска порцию данных файла в 512 байт, внутри которой находится требуемые 128. Затем эти 128 байт пересылаются в область данных программы, расположенную в памяти. Если программе потребуется следующая запись в 128 байт, то MS - DOS сначала проверит, не находится ли она в буфере. Если запись в буфере, то не нужен повторный доступ к диску.

Команда BUFFERS определяет количество буферов диска, выделяемых операционной системой при загрузке. Увеличение числа буферов может сократить время выполнения программы. Чем больше буферов, тем больше секторов диска может находиться в памяти, и следовательно, тем реже производятся непосредственные обращения к диску. Однако, чем больше буферов, тем дольше они просматриваются при поиске нужного сектора. При большом количестве буферов операция обращения к диску происходит быстрее, чем операция сканирования буферов. При определении количества буферов необходимо учитывать количество занимаемой ими памяти. Для каждого буфера выделяется 512 байт, и эта память добавляется к резервируемой MS - DOS. Количество буферов нужно рассчитывать так, чтобы свободной (оставшейся) памяти хватало для работы программ.

Команда может выделить от 1 до 99 буферов диска. Оптимальное количество буферов варьируется в зависимости от назначения прикладной программы. Если программа рассчитана на небольшое число дисковых операций ("чтение" или "запись"), то подойдет количество буферов, принимаемое по умолчанию, - 2. В случае большого количества дисковых операций, количество буферов необходимо увеличить. Наиболее удачный вариант определяется только методом проб и ошибок. Если файл CONFIG.SYS не включает команду BUFFERS, то количество буферов определяется зашитым в операционную систему значением.

CHDIR - внутренняя

ФУНКЦИЯ: Изменить рабочий директорий или вывести на экран рабочий путь по дереву директориев.

ФОРМАТ:

```
chdir [[d:]path]
```

ПРИМЕРЫ:

```
chdir \suba1\suba2
```

```
cd B:\subb1
```

```
chdir
```

```
cd
```

Рабочий директорий - это директорий, находящийся в активном состоянии в данный момент времени. На каждом устройстве в текущий момент может быть только один рабочий директорий. Команда CHDIR (CHange DIRectory) изменяет рабочий директорий. Она также используется для получения на экране рабочего пути по дереву директориев на заданном устройстве. Команду можно вводить и, как CHDIR, и как CD.

Изменение рабочего директория

Чтобы изменить рабочий директорий, набирается имя команды CHDIR (или CD) и спецификатор пути следующего рабочего директория. Предположим, что путь от рабочего корневого директория к поддиректорию SUBA2 выглядит:

ROOT DIRECTORY (корневая директория), SUBA1, SUBA2

Тогда спецификатор пути директория SUBA2: \SUBA1\SUBA2. Отметим, что корневой директорий обозначен первым обратным слэшем. Назначим SUBA2 рабочим директорием диска C:

```
C:\>cd \suba1\suba2  
<= Enter
```

Чтобы назначить рабочим директорием директорий на нерабочем диске, перед спецификатором пути директория указывается соответствующий шифр устройства:

```
C:\>chdir A:\subb1  
<= Enter
```

Получить путь к рабочему директорию

Если командная строка не содержит параметров пути, то на экран выводится спецификатор пути рабочего директория на рабочем устройстве. Введем команду:

```
C:\>chdir  
<= Enter
```

На экране появится:

```
C:\suba1\suba2
```

Чтобы получить спецификатор пути рабочего директория на нерабочем диске, в командной строке указывается соответствующий шифр устройства:

```
C:\>cd A:  
<= Enter
```

A:\>subb1

Предполагается, что команда вводится в командной строке.

CLS - внутренняя

ФУНКЦИЯ: Очищает экран, возвращая курсор в исходную позицию - слева вверху экрана.

ФОРМАТ:

cls

ПРИМЕР:

cls

COPY - внутренняя

ФУНКЦИЯ: Копирование файлов. Объединение нескольких файлов в один общий. Обмен данными между периферийными устройствами и файлами.

ФОРМАТ:

copy [/a] [/b] [d:] [path]filename [.ext] [/a] [/b] [+[:] [path] filename [.ext] [/a] [/b]...] [d:][path] [filename [.ext]] [/a] [/b] [/v]

ПРИМЕРЫ:

copy file1

copy file1+file2 B:file3

copy con:file4.txt

COPY - одна из важнейших команд MS - DOS. Главным образом, она используется для копирования файлов. Однако, она может служить для объединения файлов или для обмена информацией между файлами и периферийными устройствами.

Копирование файлов

Чтобы скопировать файл, наберите COPY, затем спецификацию копируемого (исходного) файла и спецификацию файла - копии (выходного файла). Файл будет скопирован и на экране появится сообщение о количестве сделанных копий:

```
C:\>copy file.txt B:file1.txt
```

```
<= Enter
```

```
1 File(s) copied
```

Если опущено имя копии, то файл копируется с исходным именем. В этом случае копию необходимо записать на другой диск или в

другой директорий того же диска. В следующем примере файл FILE.TXT копируется на диск В:

```
C:\>copy file.txt B:  
<= Enter
```

```
1 File(s) copied
```

В MS - DOS версиях 3.X и выше в командную строку можно ввести спецификатор пути для исходного файла и для файла копии. Если спецификаторы не указаны, то COPY обрабатывает рабочий директорий. К примеру, скопируем файл FILE.TXT, находящийся в рабочем директории диска C:, в директорий DATA на том же диске:

```
C:\>copy file.txt \data  
<= Enter
```

```
1 File(s) copied
```

Файл можно скопировать в тот же директорий, но, обязательно, с другим именем:

```
C:\>copy file.txt file2.txt  
<= Enter
```

```
1 File(s) copied
```

При введении имени файла допускается использование замещаемых символов. Скопируем, например, все файлы директория LETTERS на диске C: с расширением .DOC. Копии запишем в директорий BACKUP на диске В:. В процессе копирования на экран выводится список откопированных файлов - файлы копируются с исходным именем. Также выдается сообщение о количестве сделанных копий:

```
C:\>copy \letters *.doc B:\backup  
<= Enter
```

```
На экране увидим:  
country.doc  
tictock.doc  
whatsup.doc  
ithurts.doc  
quack.doc
```

5 File(s) copied

Объединение файлов

COPY можно использовать для объединения файлов. В командной строке спецификации файлов разделяются знаком "+". Файл, получающийся в результате слияния, представляет собой комбинацию файлов, располагающихся в порядке их следования в командной строке. В следующем примере два файла LIST1.TXT и LIST2.TXT объединяются в файл BIGLIST.TXT, но исходные файлы сохраняются. На экране появляется сообщение о количестве сделанных копий:

```
C:\>copy list1.txt+list2.txt biglist.txt  
<= Enter
```

1 File(s) copied

Если имя нового файла не указано, то результат копируется в первый файл. Его содержимое заменяется содержимым конечного файла. При введении имени файла также можно использовать замещаемые символы. Объединим, к примеру все файлы рабочего директория с расширением .TXT. Конечный файл назовем COMBINE.DAT:

```
C:\>copy *.txt combine.dat  
<= Enter
```

```
letter1.txt  
insert1.txt  
insert2.txt
```

1 File(s) copied

При использовании замещаемых символов необходимо указывать имя конечного файла. В противном случае MS - DOS попытается скопировать первый файл (LETTER1.TXT) в первый файл и процесс копирования будет остановлен. Следующая команда объединяет файл с расширением .TXT с файлом, имеющим то же основное имя и расширение .DAT (если он существует). Результат сохраняет основное имя, но получает расширение .DOC. Например, файл LETTER.TXT объединяется с файлом LETTER.DAT и результат записывается файлом LETTER.DOC:

```
C:\>copy *.txt+*.dat *.doc  
<= Enter
```

letter1.txt
letter1.dat
letter2.txt
letter2.dat
essay1.txt
essay1.dat

3 File(s) copied

Замещаемые символы требуют аккуратности в обращении. Предположим, вам нужно объединить все файлы с расширением .DOC в файл BIG.DOC. Вы даете команду:

```
copy *.doc big.doc  
<= Enter
```

К сожалению, вы ошиблись. Процесс слияния начинается организацией конечного файла, в нашем случае, файла BIG.DOC. Если ранее на диске существовал файл с этим именем, то его содержимое стирается и он заменяется новым файлом. Далее, MS - DOS начинает отбор файлов с расширением .DOC. Вновь организованный файл BIG.DOC оказывается подходящим для включения в группу объединяемых файлов. К счастью, MS - DOS не отнесет его к этой группе, однако на экране появится сообщение:

Content of destination lost before copy - файл назначения потерян перед копированием

Отбор и копирование остальных файлов производится в обычном порядке. Указанное сообщение появляется независимо от того, существовал или не существовал файл BIG.DOC до введения команды. Если файл существовал, то его содержимое теряется, и не включается в новый файл. Обойти эту ситуацию можно двумя способами. Во - первых, можно определить BIG.DOC, как старый файл, к которому добавляются все остальные файлы с расширением .DOC:

```
copy big.doc+*.doc  
<= Enter
```

Во - вторых, конечному файлу можно присвоить другое расширение, а именно:

```
copy *.doc big.dat  
<= Enter
```

Периферийные устройства

Команду COPY можно использовать для пересылки файлов на периферийные устройства и для организации обмена информацией между ними. В этом случае вместо спецификации файла указывается имя соответствующего устройства. Предположим, например, что вам нужно распечатать файл SECRET.TXT. Для этого можно воспользоваться COPY и в качестве выходного файла указать PRN (зарезервированное имя стандартного принтера):

```
C:\>copy secret.txt prn  
<= Enter
```

1 File(s) copied

Можно произвести и обратный процесс - записать информацию, полученную с периферийного устройства, в виде файла. Чаще всего этот способ используют для организации текстового файла непосредственно с клавиатуры (CON) с помощью команды COPY CON:FILESPEC. Введение строки данных заканчивается нажатием Enter, а конец ввода всех данных нажатием: Ctrl + Z и Enter, после чего данные записываются на диск:

```
C:\>copy con: keyboard.txt  
<= Enter
```

```
This is A sample file that is being created from  
the keyboard. Ctrl + Z is typed and the Enter key is  
pressed to send the file to the disk7 The file can then  
be viewed by entering the command TYPE KEYBOARD.TXT.
```

```
^Z  
<= Enter
```

1 File(s) copied

Копирование с устройства на устройство

Команду COPY можно использовать при копировании данных с одного устройства на другое. Команда вводится в обычном формате. Однако, в командной строке вместо спецификаций файлов указываются названия устройств. В следующем примеры данные, введенные с клавиатуры (CON), пересылаются непосредственно на принтер (PRN). При введении данных с клавиатуры конец каждой строки от-

мечается нажатием клавиши Enter; по окончании ввода нажимается комбинация клавиш Ctrl + Z, а затем Enter:

```
C:\>copy con: prn  
<= Enter
```

```
This is A sample file to demonstrate the use of COPY  
is sending data between periphepal devices. At the end  
of the input you will press Ctrl + Z and then press Enter.  
This text will be sent to the printer.
```

```
^Z  
<= Enter  
1 File(s) copied
```

При попытке копировать с устройства, работающего в двоичном режиме, вы получите сообщение:

```
Cannot do binary read from A device - Нельзя выполнить бинарное  
чтение с устройства A
```

Положение можно исправить, указав символьный параметр /A.

DEL - внутренняя

ФУНКЦИЯ: Уничтожение дисковых файлов.

ФОРМАТ:

```
del[d:][path][filename[.ext]]
```

ПРИМЕРЫ:

```
del badfile.txt  
erase badfile.txt
```

Команда DEL (или аналогичная команда ERASE) служит для удаления файлов с диска. Чтобы удалить файл, наберите DEL (или ERASE) и спецификацию ненужного файла. Если спецификация не содержит шифр устройства и спецификатор пути, то файл выбирается из рабочего директория рабочего диска.

Удалим, к примеру, файл BADFILE.TXT из рабочего директория рабочего диска:

```
C:\>del badfile.txt  
<= Enter
```

При использовании замещаемых символов в параметрах FILENAME или .EXT, одной командой можно удалить группу фай-

лов. Однако, пользоваться замещаемыми символами следует аккуратно, чтобы не удалить нужных файлов вместе с ненужными. Следующая команда удаляет все файлы с расширением .DOC из рабочего директория диска В:

```
C:\>del B:*.doc  
<= Enter
```

При введении спецификации файла в виде "*".*" удаляются все файлы рабочего директория. Операционная система выдает вопрос на удаление, чтобы лишний раз подстраховать программиста:

```
C:\>del *.*  
<= Enter
```

```
Are you sure (Y/N)? Y - Вы уверены ?Y (или N)  
<= Enter
```

Если у вас возникли сомнения относительно удаления файлов, введите "N", в противном случае введите "Y" и нажмите Enter.

В действительности файлы не удаляются с диска. Команда изменяет соответствующую запись директория так, чтобы система в дальнейшем обращалась с файлом, как с несуществующим. Поэтому если вы случайно удалили нужный вам файл, вы можете восстановить его с помощью программы DEBUG. Существует несколько вспомогательных программ, например, из комплекса программ Norton Utilities, которыми можно пользоваться для восстановления уничтоженных файлов.

Однако, чтобы файл можно было восстановить, после удаления, до его восстановления на диск не должны быть записаны новые данные.

DEVICE - внутренняя

ФУНКЦИЯ: Включает в операционную систему драйвер периферийного устройства.

ФОРМАТ:

```
device=[d:][path]filename[.ext]
```

ПРИМЕР:

```
device=ansi.sys
```

Команда DEVICE присваивает параметру FILENAME имя драйвера, который необходимо загрузить в память. Команда может использоваться только в качестве оператора файла конфигурации

CONFIG.SYS. Операторы этого файла обрабатываются операционной системой в процессе загрузки. Если оператор включает команду DEVICE, то драйвер с соответствующим именем загружается в память. Чтобы ввести команду, наберите DEVICE= и имя нужного драйвера (включая расширение). Команда вводится в CONFIG.SYS с помощью текстового редактора. Для организации CONFIG.SYS можно использовать и команду COPY CON:.

Обычно CONFIG.SYS находится в корневом директории рабочего диска. В следующем примере организуется файл конфигурации CONFIG.SYS. При загрузке драйверы ANSI.SYS и VDISK.SYS будут выбраны из поддиректория \DOS на диске C:

```
C:\>copy con: config.sys  
<= Enter
```

```
device=C:\dos\ansi.sys  
device=C:\dos\vdisk.sys  
^Z  
<= Enter
```

```
1 File(s) copied
```

ANSI.SYS - это драйвер клавиатуры MS - DOS. VDISK.SYS - драйвер виртуального диска (RAM). DRIVER.SYS - драйвер дисковод для гибкой дискеты. DISPLAY.SYS - это драйвер, обеспечивающий переключение кода страницы. PRINTER.SYS - еще один драйвер, обеспечивающий переключение кода страницы при печати.

DIR - внутренняя

ФУНКЦИЯ: Вывод содержимого директория на экран.

ФОРМАТ:

```
dir [d:][filename[.ext]][/p][/w]  
dir [d:][path][filename[.ext]][/p][/w]
```

ПРИМЕРЫ:

```
dir  
dir B:  
dir B:\subdir1\*.doc/w
```

Команда DIR выводит на экран содержимое директория, то есть список содержащихся в директории поддиректориев и файлов. Список включает следующие параметры: имя и расширение файла, размер занимаемой памяти, маркер времени и даты.

Если в командной строке не указан спецификатор пути директория, то выводится содержимое рабочего директория рабочего диска:

```
C:\>dir  
<= Enter
```

```
Volume in drive C: is WRITE_DISK1  
Directory of C:\  
COMMAND.COM 17664 3-08-87 12:00p  
C:\<DIR> 1-01-94 12:07a  
WS2PATH.BAT 23 12-07-97 8:18a  
CONFIG.SYS 128 9-25-97 7:24p  
SETCLOCK.COM 853 9-19-96 4:24p  
WS<DIR> 9-08-96 4:27p  
6 File(s) 110269 bytes free
```

В первой колонке перечислены основные имена файлов, во второй - даны их расширения. В третьей колонке приводится размер занимаемой файлом памяти в байтах. Четвертая и пятая колонки представляют собой соответственно маркеры даты и времени создания или последней корректировки файла.

Отметим, что в дополнение к обычной информации, на экране содержится метка тома - WRITE_DISK1. Метка тома - это имя диска (в нашем случае, рабочего диска C:). строка "Directory of C:\:" означает, что выводится содержимое корневого директория диска C:. Две записи помечены ссылкой <DIR>. Эти записи являются описанием поддиректориев, вложенных в корневой директорий диска. Последняя строка содержит информацию о количестве свободной памяти на диске - 110269 байтов. Чтобы посмотреть содержимое любого другого директория, набирается DIR и спецификатор пути этого директория:

```
C:\>dir \ws  
<= Enter
```

```
Volume in drive C: is WRITE_DISK1  
Directory of C:\WS  
.<DIR> 9-08-96 4:27p  
..<DIR> 9-08-96 4:27a  
WS.EXE 60128 6-25-97 7:24p  
WS.HLP 45853 6-25-96 7:24p  
BATES.DOC 4096 9-17-98 4:27p  
GILMORE.DOC 4096 9-18-98 2:15p  
6 File(s) 110269 bytes free
```

Отметим, что первые две записи содержат соответственно одну и две десятичные точки на месте имени файла. Это сокращенное обозначение поддиректориев. Десятичная точка в первой записи обозначает директорий, содержимое которого выводится на экран. В нашем случае это директорий WS. Его родительским директорием является корневой директорий диска. Две десятичные точки во второй записи обозначают родительский директорий по отношению к просматриваемому (т.е. в нашем случае корневой).

Ключевые параметры /P и /W

Ключевой параметр /P обычно используется при просмотре содержимого больших директориев. При выводе на экран большого количества записей, они пробегают по экрану быстрее, чем возможно их прочесть. В этом можно убедиться, установив системную дискету на дисковод A: и введя команду DIR A:. Вы сможете просмотреть только последнюю часть данных, оставшихся на экране после выполнения команды.

Введение параметра /P позволяет управлять выводом. При полном заполнении страницы экрана, вывод прекращается и возобновляется при нажатии любой клавиши, показывая следующую страницу с информацией на экране. Ключевой параметр /W располагает информацию во всю ширину экрана. Параметры "имя файла" и "расширение" выводятся по пять раз в каждой строке. Параметры "размер файла", "маркер даты" и "маркер времени" в этом режиме не выводятся.

Получение информации о конкретных файлах

При введении команды можно задать имя конкретного файла. Тогда на экране появится информация, относящаяся только к этому файлу:

```
C:\>dir ws2path.bat  
<= Enter
```

```
Volume in drive C: is WRITE_DISK1  
Directory of C:\  
WS2PATH.BAT 23 12-07-87 8:18a  
1 File(s) 110269 bytes free
```

Такой способ применения команды используется при необходимости получения информации об одном из большой группы файлов. Предположим, вам нужно узнать, находится ли файл LETTERS.DOC

в поддиректории WS. Вы можете ввести команду DIR \WS и просмотреть содержимое директория. Но можно поступить и по - другому. Введите DIR \WS LETTERS.DOC. Операционная система сама произведет поиск файла и, если файл найден, то на экран выдается относящаяся к нему информация. В противном случае появляется сообщение об отсутствии файла в данной поддиректории на заданном диске (в данном случае - рабочем диске).

Замещаемые символы

С помощью замещаемых символов можно получать информацию о конкретной группе файлов. Предположим, вам нужно просмотреть файлы с первой буквой основного имени "Q" и расширением .DOC. Пусть файлы находятся в корневом директории диска В:. Введем в командной строке следующую команду:

```
C:\>dir B:\q*.doc  
<= Enter
```

MS - DOS выбирает подходящие к данной спецификации файлы и выдает относящуюся к ним информацию на экран. Если в командной строке указано только основное имя файла, а расширение отброшено, то на экран выводится список всех файлов с указанным именем (и с разными расширениями). Выведем, например, список файлов, находящихся в корневом директории диска В:, с основным именем ANIMALS:

```
C:\>dir B:\animals  
<= Enter
```

Если при введении этой команды после основного имени набрать десятичную точку, то на экране появится информация, относящаяся к файлу с именем ANIMALS, не имеющему расширения:

```
C:\>dir B:\animals.  
<= Enter
```

DISKCOMP - внешняя

ФУНКЦИЯ: Сравнение содержимого двух гибких дискет.

ФОРМАТ:

diskcomp [d:][d:]

diskcomp [d:][d:][1][8]

ПРИМЕР:

diskcomp A: B:

DISKCOMP (COMPare DISKette) - это сервисная программа MS - DOS, сравнивающая содержимое двух дискет по секторам. Наиболее часто используется для сравнения копий или копии с оригиналом, если копии получены с помощью команды DISKCOPY. Команда служит для сравнения именно гибких дискет. Ее нельзя использовать для сравнения жестких дисков, дисков RAM или дисков в сетевых системах.

Применение DISKCOMP

DISKCOMP является внешней командой MS - DOS, поэтому перед обращением ней необходимо убедиться в доступности файла DISKCOMP.COM на жестком диске. Если DISKCOMP.COM не содержится в рабочей директории рабочего диска, то его спецификатор пути вводится с помощью команды PATH.

Если ваша система включает два дисководов для гибких дискет, то установите ваши дискеты на дисководы. Такая конфигурация системы экономит время, затрачиваемое на перестановку дискет в системе с одним дисководом.

Введите команду:

```
C:\>diskcomp A: B:  
<= Enter
```

На экране последовательно появятся два сообщения:

Insert the first diskette in drive A: - вставьте первую дискету в дисковод A:.

и

Insert the second diskette in drive B: - вставьте вторую дискету в дисковод B: .

Неважно, какая из дискет установлена на дисководе A:, а какая - на дисководе B:. Если они установлены, начинайте сравнение - ответьте на запросы нажатием любой клавиши. Если ваша система включает лишь один дисковод для гибкого диска, или, если в командной строке вы указали шифр одного дисковода, то на экране также появятся сообщения на установку первой и второй дискет. И здесь, какую - то из своих дискет (неважно, какую) вы условно принимаете за первую и какую - то за вторую. В дальнейшем, необходи-

мо следить за соответствием дискет, назначенным вами номерам - А: или В:. Сравнение дискет производится по дорожкам и секторам. Если в процессе сравнения содержимое всех дорожек совпало, то на экране появится сообщение:

Diskettes compare OK - сравнение прошло успешно (OK).

В противном случае появится список несовпадающих дорожек и адреса несовпадений на соответствующей дорожке. По окончании процесса сравнения на экране появится спрос на продолжение процедуры. При ответе "Y", установите следующую пару дискет для сравнения. При ответе "N", команда возвращает управление MS - DOS.

DISKCOPY - внешняя

ФУНКЦИЯ: Копирование с дискеты на дискету.

ФОРМАТ:

diskcopy [d:][d:]

ПРИМЕР:

diskcopy A: B:

DISKCOPY (COPY DISKette) - это сервисная программа MS - DOS, производящая копирование содержимого одной дискеты на другую. При попытке копирования с жесткого диска или на жесткий диск, будет выдано сообщение об ошибке.

Применение DISKCOPY

DISKCOPY является внешней командой MS - DOS, поэтому перед обращением к ней необходимо убедиться в доступности файла DISKCOPY.COM. Если DISKCOPY.COM не содержится в рабочем директории рабочего диска, то его спецификатор пути вводится с помощью команды PATH. В приведенном ниже примере считается, что этот файл находится на диске С:. Если ваша система включает два дисководов для гибких дискет, установите ваши дискеты. Такая конфигурация системы экономит время, затрачиваемое на перестановку дискет в системе с одним дисководом. Введите команду:

```
C:\>diskcopy A: B:  
<= Enter
```

При введении команды файл DISKCOPY.COM загружается в память, и на экране появляется запрос на установку исходной (копируемой) дискеты на первый в командной строке дисковод и дискеты

для копии - на второй. После установки дискет на запрашиваемые устройства нажмите любую клавишу для начала процесса копирования:

Insert the first diskette in drive A:
Insert the second diskette in drive B:
Strike any key when ready

Если ваша система включает только один дисковод для гибкого диска, или, если в командной строке вы указали шифр одного дисковода, то на экране также появятся сообщения на установку первой и второй дискет. Помните, что первой (A:) является исходная дискета, а второй (B:) - дискета для копии. В дальнейшем, чтобы не потерять информации на исходной дискете, необходимо строго следить за соответствием дискет указанным номерам. Чтобы предупредить возможные недоразумения, можно защитить исходную дискету от записи, заклеив выемку на ее правой стороне (для дискет 1,2М). По окончании процесса копирования на экране появится вопрос на продолжение процедуры. При ответе "Y", установите следующую пару дискет для копирования. При ответе "N", команда возвращает управление MS - DOS. Большинство версий DISKCOPY форматируют дискету для копии, если она не была отформатирована предварительно.

Сравнение COPY И DISKCOPY

Очень важно понимать различие двух копирующих команд - COPY и DISKCOPY. DISKCOPY считывает информацию с первой дорожки исходной дискеты и записывает ее на первую дорожку дискеты для копии, затем считывается вторая дорожка и т.д. В процессе копирования на дискете для копии вся старая информация стирается. COPY считывает информацию из первого сектора копируемого файла и записывает ее в первый свободный сектор на дискете для копии. Затем считывается содержимое второго сектора и т.д. до конца копируемого файла. При этом копируются только файлы, указанные в командной строке. Старое содержимое дискеты для копии сохраняется. Если файл на дискете разбит на несколько участков (не укладывается в один непрерывный участок), то доступ к нему осуществляется медленнее, чем к непрерывному файлу. Файлы, занимающие несколько участков, целесообразно копировать на свободные дискеты с помощью команды XCOPY *.* /S (или COPY *.*). Команда XCOPY или COPY скопирует файл на один непрерывный участок, в то время как DISKCOPY перенесла бы его так, как есть.

ERASE - внутренняя

ФУНКЦИЯ: Уничтожение дисковых файлов.

ФОРМАТ:

erase [d:][path][filename[.ext]]

ПРИМЕРЫ:

erase badfile.txt или del badfile.txt

По действию команда ERASE полностью идентична команде DEL.

FILES - внутренняя

ФУНКЦИЯ: Задаёт объем памяти, выделяемый для ссылок на управляющую запись файла.

ФОРМАТ:

files=xx

ПРИМЕР:

files=25

Команда FILES определяет количество памяти, выделяемой для управляющего блока, в котором содержатся ссылки на управляющую запись файла. Размер выделяемой памяти определяется максимальным количеством ссылок. Ссылка на управляющую запись файла это 16 битовое число, которое присваивается файлу при его создании или открытии. Ссылки используются для управления файлами, одновременно обрабатываемыми одной прикладной программой.

Применение FILES

Команда FILES может быть использована в качестве оператора файла CONFIG.SYS. Файл конфигурации CONFIG.SYS - это текстовый файл, включающий одну или несколько команд, выполняемых в процессе загрузки операционной системы. Каждый оператор файла определяет конкретный параметр режима работы MS - DOS.

Команда FILES задает максимальное количество ссылок на управляющую запись файла. Для создания или обработки CONFIG.SYS обычно используется текстовый редактор. Файл можно организовать и с помощью команды COPY CON: CONFIG.SYS:

```
C:\>copy con: config.sys
```

```
<= Enter
```

```
files=10
```

^Z
<=Enter

1 File(s) copied

Если команда FILES не выполняется в процессе загрузки, то MS - DOS выделяет участок памяти для 8 ссылок на управляющую запись. Введенный параметр играет важную роль для большинства прикладных программ. Если программе требуется большее количество ссылок, чем имеется в наличии, то на экран выводится сообщение:

No free file handles - нет свободных заголовков файлов.

Для каждого дополнительного файла (сверх принятых по умолчанию восьми) требуется по 39 байт оперативной памяти.

FIND - внешняя

ФУНКЦИЯ: Осуществляет поиск указанной текстовой строки внутри одного или группы файлов.

ФОРМАТ:

`find [/v][/c][/n]string[filespec][filespec]...`

ПРИМЕРЫ:

```
find "Bruce" records.txt
find /v "Floyd" records.txt
find /C "Linda" records.txt
find /n "Born" records.txt
```

Команда FIND является фильтром MS - DOS. Она используется для поиска заданной текстовой строки внутри одного или группы текстовых файлов. Текстовая строка, по которой производится поиск, в командной строке заключается в кавычки ("LIKE THIS"). Иначе бы в фильтр могли попадать выходные данные любой программы или команды.

Выходные данные команды FIND можно переслать на устройство стандартного вывода, равно, как и на другое периферийное устройство, или оформить в виде информационного файла. FIND является внешней командой MS - DOS. Это означает, что перед обращением к ней необходимо убедиться в доступности файла FIND.EXE. Он должен находиться на одном из системных дисков.

Все ключевые параметры FIND являются необязательными. При введении параметра /V, на экран выводятся только те строки файла, которые не содержат заданной строки. При введении параметра /C, выводятся только номера строк файла, включающих заданную стро-

ку. При введении параметра /N выводятся только строки текстового файла, включающие заданную строку, причем каждой строке предшествует ее порядковый номер (внутри файла).

FOR - внутренняя

ФУНКЦИЯ: Обеспечивает повторное выполнение команды для каждого из группы заданных параметров.

ФОРМАТ:

for %%variable IN (set of parameters) DO command

ПРИМЕРЫ:

for %%A IN (file1 file2 file3) DO del %%A

for %B IN (example.bat program.txt letter) DO copy %B prn

Команда обеспечивает циклическое выполнение какой - либо команды MS - DOS. Указанная команда выполняется один раз для каждого из группы заданных параметров. Командная строка начинается словом FOR, после которого набирается пустая переменная. Если команда используется в качестве оператора командного файла, то пустой переменной должен предшествовать двойной знак процента (%%). Если команда вводится из командной строки системы - одиночный знак (%).

При выполнении команды пустая переменная последовательно заменяется каждым из введенных параметров. Вслед за пустой переменной заглавными буквами набирается слово IN. Затем в круглых скобках перечисляются параметры. Группа параметров замыкается словом DO, также набираемым заглавными буквами. За ним следует собственно имя команды MS - DOS. В следующем примере команда FOR используется для распечатки файлов EXAMPLE.BAT, PROGRAM.TXT и LETTER:

```
C:\>for %B IN (example.bat program.txt letter) DO copy %B prn  
<=Enter
```

```
copy example.bat prn
```

```
1 File(s) copied
```

```
copy program.txt prn
```

```
1 File(s) copied
```

```
copy letter prn
```

```
1 File(s) copied
```

Применение команды в командных файлах было рассмотрено в предыдущих главах.

FORMAT - внешняя

ФУНКЦИЯ: Инициализация - форматирование гибких и жестких дисков.

ФОРМАТ:

`format [d:]/[s]`

`format [d:]/[s]/[1]/[8]/[v]/[b]`

`format [d:]/[s]/[1]/[8]/[v]/[b]/[4]`

`format [d:]/[s]/[1]/[8]/[v]/[b]/[4]/[N:xx /T:yy]`

ПРИМЕРЫ:

`format B:`

`format B:/s`

`format C:/s/v`

Перед использованием гибкие и жесткие диски необходимо инициализировать. Процесс инициализации называется форматированием. Он производится командой **FORMAT**. В процессе форматирования диски разбиваются на участки, называемые секторами. Несколько секторов составляют дорожку. Каждому сектору и каждой дорожке присваивается индивидуальный номер, по которому они и различаются операционной системой. В процессе форматирования на диске организуется запись загрузки, которая используется при загрузке системы. Кроме того, создается таблица размещения файлов и директорий диска. Эти структуры данных служат для учета находящихся на диске файлов и поддиректориев.

Применение FORMAT

При форматировании дискеты вся старая информация на ней уничтожается. При форматировании жесткого диска стираются данные, расположенные на мини - диске **MS - DOS**. Необходимо форматировать все новые дискеты, которые предполагается использовать в **MS - DOS**. При форматировании старых дискет предварительно копируются все полезные файлы.

Процесс форматирования жесткого диска командой **FORMAT** занимает несколько минут. **FORMAT** является внешней командой **MS - DOS**. Это означает, что перед обращением к ней необходимо убедиться в доступности файла **FORMAT.COM**. Файл должен находиться на одном из системных дисков. Он обязательно включается в число файлов, расположенных на прилагаемой к машине системной дискете. В приведенных примерах считается, что файл **FORMAT.COM** находится на дискете **A:**.

Если в вашей системе имеется два дисководов для гибких дискет, то процесс форматирования облегчается. Копия системной дискеты

устанавливается на дисковод А:, форматируемая дискета - на дисковод В:. Если система включает только один дисковод для гибких дискетов, поместите копию системной дискеты на дисковод А:. Вводимые команды в обоих случаях идентичны, однако, при наличии одного дисковода, в процессе форматирования вы будете получать запросы на замену дискеты. Введите команду:

```
A:\>format B:  
<= Enter
```

FORMAT.COM загружается в память и на экран выводится информация, касающаяся фирмы - изготовителя, и вопрос:

Insert new diskette for drive B: - вставьте новую дискету в дисковод В: .

and strike any key when ready - и нажмите любую клавишу, когда будете готовы.

Нажав любую клавишу, вы стартуете процесс форматирования. На экране появится свидетельствующее об этом сообщение:

```
Formatting... - идет форматирование.
```

Процесс форматирования гибкой дискеты занимает около минуты. По окончании процесса вы получите сообщение:

```
Formatting...Format complete - формат закончен.
```

На экране, если дискета на 360К появится информация о состоянии диска:

```
362496 bytes total disk space - 362496 байт полного дискового пространства.
```

```
362496 bytes available on disk - 362496 байт дискового пространства, возможного для использования.
```

Полученные значения зависят от типа форматируемой дискеты, или от размера мини - диска. После этого на экран выводится вопрос на форматирование следующей дискеты:

```
Format another (Y/N)?_ - форматировать еще ?
```

Если вам нужно отформатировать еще одну дискету, введите "Y". Чтобы вернуть управление MS - DOS, введите "N". При фор-

матировании жесткого диска в командной строке после имени команды указывается шифр устройства (например, FORMAT C:).

Системные файлы

Ключевой параметр /S используется для обозначения системных файлов и файла COMMAND.COM. Скрытые системные файлы MS - DOS называются IO.SYS и MSDOS.SYS. Эти файлы обязательно должны находиться на диске или дискете, с которой производится загрузка MS - DOS. Порядок и месторасположение этих файлов на диске играет существенную роль в организации рабочего процесса.

Если вы перенесете на диск системные файлы командой COPY, то этот диск не станет системным (вы не сможете с него загружаться). Чтобы отформатировать диск и поместить туда системные файлы и файл COMMAND.COM, используется команда:

```
A:\>format B:/s  
<= Enter
```

На экране появляется:

```
Insert new diskette for drive B:  
and strike any key when ready
```

```
Нажимаем любую клавишу:  
Formatting...
```

Когда форматирование закончено на экране видим:

```
Format complete - формат закончен  
System transferred - Система перенесена
```

```
362496 bytes total disk space  
38912 bytes used by the sysytem - 38912 байт используется систе-  
мой.
```

```
323584 bytes available on disk  
Format another (Y/N)?_
```

Отметим, что при использовании этой команды на экран выводится сообщение, включающее строку "System transferred", а также строку, в которой указан размер памяти, занимаемой системными файлами и файлом COMMAND.COM. Если теперь для просмотра содержимого дискеты вы введете команду:

```
dir B:  
<= Enter
```

то на экране вы увидите описание файла COMMAND.COM.

Введение метки тома

Команду FORMAT можно использовать для присвоения диску или дискете метки тома. Метка тома является идентификатором диска. Ее нельзя использовать в качестве параметра команды. Метка выводится на экран при введении команды DIR. Чтобы операционная система "знала", что в процессе форматирования диска ему присваивается метка тома, в командную строку FORMAT вводится ключевой параметр /V. По окончании процесса форматирования на экран выдается запрос на введение метки. Метка тома может включать до 11 символов. Набор символов, используемый при организации имени файла, используется и при организации метки тома:

```
A:\>format B:/s/v  
<= Enter
```

На экране появляется:

```
Insert new diskette for drive B:  
and strike any key when ready
```

Нажимаем, например, клавишу Пробел:

```
Formatting...
```

По окончанию процесса на экран выводится:

```
Format complete  
System transferred
```

Volume label (11 characters, ENTER for none)? - метка тома.

```
WAITE_DISK1
```

```
<= Enter
```

```
362496 bytes total disk space  
38912 bytes used by the sysytem  
323584 bytes available on disk  
Format another (Y/N)?_
```

Теперь, чтобы посмотреть содержимое диска, введем команду DIR B:

```
A:\>dir B:  
<= Enter
```

```
Volume in drive B: is WAITE_DISK1 - метка тома диска B: .  
Directory of A:\
```

```
COMMAND COM 15480 3-01-95 2:00a  
1 File(s) 323584 bytes free - 323584 байт свободно.
```

IF - внутренняя

ФУНКЦИЯ: Обеспечивает выполнение команды при выполнении заданного условия.

ФОРМАТ:
if [not] condition command

ПРИМЕРЫ:
if exist somefile.dat type somefile.dat
if %1==roses goto roses
if not exist file.bak copy file.txt file.bak

Операторы IF, главным образом, применяются внутри командных файлов. Оператор может проверять наличия файла в рабочей директории указанного (или рабочего) диска. Следующий оператор выводит содержимое файла SOMEFILE.DAT на экран, если он находится в рабочей директории диска C: (рабочего диска):

```
C:\>if exist somefile.dat type somefile.dat  
<=Enter
```

Производится так же проверка идентичности двух символьных строк. Этот оператор обычно используется для сравнения символьной переменной, присланной в командный файл в качестве параметра, с символьной переменной, определенной внутри файла. В следующем примере параметр %1 сравнивается с символьной строкой ROSES. Если результат проверки положительный, то производится переход на метку строки командного файла ROSES:

```
if %1==roses goto roses
```

Указанная команда может выполняться или не выполняться, в зависимости от результата выполнения некоторой команды или при-

кладной программы. В результате выполнения программы или команды MS - DOS переменной ERRORLEVEL может быть присвоено некоторое значение (код). Это значение проверяется оператором IF. Команда в области действия этого оператора выполняется, если значение ERRORLEVEL равно или больше заданного числа.

В приведенном ниже примере на экран выводится содержимое рабочего директория, если значение переменной ERRORLEVEL больше или равно двум:

```
C:\>if errorlevel 2 dir
```

IF NOT - внутренняя

Команда в области действия оператора IF NOT выполняется, если результат проверки отрицательный. Оператор обрабатывает те же типы условий, что и оператор IF. В следующем примере файлу FILE.TXT присваивается имя FILE.BAK при условии, если в рабочем директории рабочего диска отсутствует файл с именем FILE.BAK:

```
C:\>if not exist file.bak copy file.txt file.bak
```

```
<=Enter
```

MKDIR - внутренняя

ФУНКЦИЯ: Организация поддиректория в рабочей директории.

ФОРМАТ:

```
mkdir [d:]path
```

```
md [d:]path
```

ПРИМЕРЫ:

```
mkdir\write
```

```
md B:\programs\business
```

Команда MKDIR (MaKe DIRectory) используется при создании поддиректориев. При ее введении используется сокращение MD. В командную строку MKDIR можно включить шифр устройства (например, C: или A:), на котором организуется поддиректорий. Если шифр устройства опущен, то поддиректорий будет организован на рабочем диске. В командной строке необходимо указывать спецификатор пути создаваемого поддиректория. Организуем, например, поддиректорий WRITE:

```
C:\>mkdir \write
```

```
<= Enter
```

В командной строке отсутствует шифр устройства. Следовательно, поддиректорий WRITE создается на рабочем диске. Спецификатор пути нового поддиректория: \WRITE, т.е. он находится на уровень ниже корневого директория. Организуем еще один поддиректорий BUSINESS:

```
C:\>md B:\programs\business  
<= Enter
```

Поддиректорий находится на диске B:. Спецификатор пути поддиректория: \PROGRAMS\BUSINESS. Это означает, что он организован на уровень ниже поддиректория PROGRAMS. PROGRAMS, в свою очередь, расположен на уровень ниже корневого директория диска B:.

PATH - внутренняя

ФУНКЦИЯ: Определяет спецификатор пути поддиректория или файла.

ФОРМАТ:

```
path [[d:]path[[[:d:]path]...]]
```

ПРИМЕРЫ:

```
path \program1\business
```

```
path B:\program2\write1;B:\program2\write2
```

Если обработчик внешней команды или командный файл не найден в рабочей директории рабочего диска, тогда требуется ввести команду PATH, определяющую поддиректорий, в котором необходимо организовать поиск. Параметры команды - это спецификаторы пути поддиректориев, в которых находятся требуемые файлы.

Предположим, что на диске A: находится несколько файлов и поддиректорий PROGRAM1. В поддиректории содержится командный файл BUSINESS.BAT. Пусть рабочий директорий рабочего диска - корневой директорий диска C:. Вам же нужно стартовать BUSINESS.BAT. Чтобы стартовать командный файл, набирается его имя. Посмотрим, что получится при нажатии клавиши Enter:

```
C:\>business  
<= Enter
```

Bad command or file name - не верная команда или имя файла.

Операционная система просмотрела корневой директорий диска C: и не нашла файл. Решив, что файл не существует, она выдала со-

общение "Bad command or file name". Имеется два выхода из создавшегося положения. Можно назначить рабочим директориум поддиректорий PROGRAM1 на диске A:. А можно воспользоваться командой PATH и ввести спецификатор пути этого поддиректория. Тогда вам не придется изменять рабочий директориум, что особенно важно, если в нем находится большинство полезных (используемых в процессе рабочего сеанса) файлов и программ. Введем:

```
C:\>path A:\program1  
<=Enter
```

Теперь MS - DOS "знает", где находится файл BUSINESS.BAT. Последняя PATH (в последовательности ранее вводимых команд PATH) назначает рабочий спецификатор пути. При введении имени команды, он появится на экране дисплея:

```
C:\>path  
<= Enter  
  
path=A:\program
```

Рабочий спецификатор остается активным до введения следующей PATH. В командной строке PATH можно указать несколько спецификаторов, разделенных точкой с запятой. Операционная система просматривает поддиректории в порядке, в котором они перечислены в командной строке.

В следующем примере PATH включает два спецификатора пути поддиректориев на диске B:. При введении команды поиск нужного файла будет производиться в поддиректории WRITE1 (расположенном на уровень ниже поддиректория PROGRAM2). При отсутствии файла в этом поддиректории, MS - DOS перейдет в поддиректорий WRITE2 (также расположенный на уровень ниже поддиректория PROGRAM2):

```
C:\>path B:\program2\write1;B:\program2\write2  
<=Enter
```

Спецификатор пути теряет статус рабочего спецификатора при введении команды PATH,;

```
C:\>path;  
<= Enter  
path=B:\program2\write1;B:\program2\write2
```

```
C:\>path;  
< = Enter
```

```
C:\>path  
< = Enter
```

No Path

PRINT - внешняя

ФУНКЦИЯ: Распечатывает группу файлов, если параллельно выполняется еще одно задание.

ФОРМАТ:

```
print [[d:][filename[.ext]][/t][c][p]...]  
print [/d:device][/b:bufferize][/u:busyticks][/M:maxticks]  
[/s:timeslice][/q:queueize][d:][filename[.ext]][/t]  
[/c][p]...]
```

ПРИМЕРЫ:

```
print file1.txt  
print file?.txt  
print file1.txt file2.txt/C file3.txt file4.txt
```

Команда PRINT - это сервисная программа MS - DOS, позволяющая распечатывать один или несколько файлов при одновременном выполнении других заданий. PRINT является внешней командой MS - DOS. Это означает, что перед обращением к ней необходимо убедиться в доступности файла PRINT.COM. Файл должен быть загружен в память (т.к. PRINT - резидентная программа TSR), либо находиться в рабочем директории рабочего диска.

Применение PRINT

Чтобы воспользоваться командой, наберите PRINT и спецификации распечатываемых файлов. При введении команды, файлы помещаются в выходную очередь. Они распечатываются по одному в порядке, в котором они находятся в очереди. Выходная очередь может включать до 10 файлов. Распечатанный файл из очереди удаляется. Если после загрузки операционной системы команда PRINT вводится впервые, то на экране появляется запрос:

Name of list device [PRN]: - Введите имя принтера

Это запрос на имя принтера. По умолчанию принимается стандартное имя параллельного принтера - PRN. Если к системе подклю-

чен параллельный принтер, соответствующий данному имени, то нажмите Enter. В противном случае введите имя принтера. По следующей команде распечатываются файлы FILE1.TXT, FILE2.TXT и FILE3.TXT. Файлы находятся в рабочем директории диска C:

```
C:\>print file1.txt file2.txt file3.txt  
<=Enter
```

```
Name of list device [PRN]:  
<= Enter
```

Resident part of print installed - Резидентная часть принтера загружена в память

```
C:file1.txt is currently being printed - печатается  
C:file2.txt is in queue - в очереди  
C:file3.txt is in queue - в очереди  
C:\>
```

На экран выводится содержимое выходной очереди, начиная с файла, распечатываемого в текущий момент. Имена остальных файлов появляются в порядке поступления на принтер. На экран также выводится стандартный системный запрос - разрешение на ввод команды. То есть в процессе работы PRINT могут выполняться и другие задания - команды или программы, в процессе работы которых принтер не требуется. В процессе печати файлов можно ввести еще одну PRINT. Обычно эта команда вводится, чтобы добавить или удалить файлы из выходной очереди. В командной строке допускается использование замещаемых символов (* и ?). Команду предыдущего примера можно ввести и так:

```
C:\>print file?.txt  
<=Enter
```

Если в рабочем директории дополнительно имеются файлы с соответствующей спецификацией, то они также будут распечатаны. С помощью PRINT можно распечатать все файлы рабочего директория каждого диска. После введения команды можно изменить рабочий директорий и ввести еще одну PRINT, чтобы распечатать файлы и этого директория. При введении имени команды, на экран выводится содержимое выходной очереди:

```
C:\>print  
<=Enter
```

C:file2.txt is currently being printed - текущий файл печатается

C:file3.txt is in queue

Параметры команды

Ключевой параметр /T служит для удаления файлов из выходной очереди и прерывания выполнения PRINT. При введении PRINT/T печать файлов прекращается, не распечатанные файлы удаляются из выходной очереди, принтер печатает сообщение:

All files cancelled by operator - все файлы удалены из очереди.

и управление возвращается MS - DOS. В MS - DOS версий 3.X и выше допускается использовать до 6 дополнительных ключевых параметров. Их можно вводить только в случае, если PRINT.COM уже загружен в память компьютера:

1. Параметр /D:DEVICE позволяет указать допустимое имя принтера. Если параметр опущен, то на экран выдается вопрос на имя устройства.

2. Параметр /B:BUFFER SIZE устанавливает размер буфера печати. Данные считываются из файла и помещаются в буфер. Только после этого они посылаются на печать (на принтер). Чем больше размер буфера, тем реже обращение к диску, и тем быстрее заканчивается процесс его печати. По умолчанию размер буфера равен 512 байт.

3. Параметр /Q:QUEUE SIZE устанавливает максимально допустимое количество файлов в выходной очереди. Допускаются значения в интервале 1 - 32. По умолчанию, т.е. без задания величины этого параметра принимается 10.

4. Оставшиеся параметры определяют способ разделения ресурсов между PRINT и выполняющимся заданием. Когда PRINT выполняется параллельно с другим заданием, может оказаться, что оба задания (или процесса) выполняются одновременно. В действительности это не так. Компьютер может обрабатывать только один процесс, однако переключение с одного процесса на другой производится очень быстро, поэтому кажется, что процессы работают одновременно.

5. Параметр /M:MAXTICKS определяет интервал времени, в течение которого работает обработчик PRINT до переключения на другую задачу. Допускаются значения от 1 до 255 тиков таймера. По умолчанию принимается 2.

6. Параметр /U:BUSITICKS определяет интервал времени, в течение которого обработчик PRINT может находиться в состоянии ожидания, если принтер занят. Если время ожидания истекло, а принтер все еще занят, то управление передается параллельному процессу. Допускаются значения от 1 до 255 тиков таймера. По умолчанию принимается 1.

RECOVER - внешняя

ФУНКЦИЯ: Восстанавливает файлы с поврежденными секторами. Восстанавливает диски с запарченным директориум.

ЗАМЕЧАНИЕ: Команда не может использоваться в сложных сетевых системах.

ФОРМАТ:

recover [d:][path]filename[.ext]

recover d

ПРИМЕРЫ:

recover badfile.txt

recover B:

Гибкие и жесткие диски разбиваются на участки, называемые секторами. Деление диска на сектора производится в процессе форматирования. Каждый сектор занимает 512 байтов. Чем больше файл, тем больше секторов он занимает. На каждом диске (на жестком и на гибком) имеется директорий. Он представляет собой справочную таблицу, отражающую текущее состояние диска. Директорий организуется в процессе форматирования диска. Его содержимое изменяется при добавлении, удалении или обработке любого файла на диске. В процессе работы с диском случайно могут запортиться один или несколько секторов. Тогда теряется возможность чтения данных в этих секторах. На экран выдается сообщение:

Data error reading C: - ошибка чтения диска.

Abort, Retry, Ignore? - отменить, повторить, игнорировать ?

Команда RECOVER восстанавливает данные при повреждении секторов на диске. С ее помощью можно восстановить, как отдельный файл, так и целый диск, если на диске запорчены сектора директория. При восстановлении файла сохраняются данные вне поврежденных секторов.

Данные в запорченных секторах теряются. Запорченные сектора помечаются (заклеиваются) и в дальнейшем операционной системой не используются. Восстановленный файл можно прочитать. Он сохраняет свое исходное имя (и расширение). В конец файла обычно

добавляются произвольные данные (не имеющие отношения к файлу) так, как новый файл, создаваемый RECOVER, на сектор (512 байт) больше исходного.

Восстановление файла

RECOVER является внешней командой MS - DOS. Это означает, что перед обращением к ней необходимо убедиться в доступности файла RECOVER.COM. Он должен находиться в рабочем директории рабочего диска. В противном случае спецификатор пути файла определяется командой PATH. Чтобы воспользоваться командой, наберите RECOVER и спецификацию восстанавливаемого файла.

При введении команды файл RECOVER.COM загружается в память. Затем выполнение файла приостанавливается - пользователь получает возможность заменить дискеты (при необходимости). Замените дискеты и нажмите любую клавишу - указанный файл будет восстановлен. На экране появится сообщение, отражающее количество восстановленных байтов (исходного файла):

```
C:\>recover B:\badfile.txt  
<= Enter
```

Press any key to begin recovery of the - нажмите любую клавишу для начала восстановления.

```
file(s) on drive B:
```

```
Нажимаем клавишу Пробел
```

```
900 of 1412 bytes recovered - 900 байт из 1412 байт восстановлены.
```

```
C:\>
```

При использовании замещаемых символов звездочка и знак вопроса в спецификации файла, восстанавливается только первый файл с подходящей спецификацией.

Восстановление диска

Диск восстанавливается (т.е. восстанавливаются все файлы на диске) только в самом крайнем случае. Чтобы установить местонахождение каждого файла, RECOVER просматривает таблицу размещения файлов на жестком или гибком диске. RECOVER не может отличить поврежденную запись директория от неповрежденной, по-

этому на диске восстанавливаются все файлы. Чтобы восстановить диск, наберите RECOVER и шифр устройства. При введении команды файл RECOVER.COM загружается в память. Затем выполнение файла приостанавливается. Пользователь, при необходимости, может заменить дискеты. Замените дискеты и нажмите любую клавишу - ваш диск будет восстановлен:

```
C:\>recover B:
```

```
<= Enter
```

```
Press any key to begin recovery of the file(s) on drive B:
```

Нажимается, например, клавиша Пробел

```
22 file(s) recovered
```

Первому восстановленному файлу присваивается имя FILE0001.REC, второму - FILE0002.REC и т.д. Поддиректории интерпретируются обработчиком, как обычные файлы, поэтому все восстановленные файлы записываются в корневой директорий. Если они в корневой директорий не помещаются, то об этом выдается сообщение. В этом случае скопируйте восстановленные файлы на отдельную дискету и удалите их с частично восстановленного диска. Затем еще раз введите RECOVER. Если и на этот раз в корневом директории не хватит памяти, повторите последнюю процедуру (копирование - удаление - восстановление). После восстановления командой DIR можно посмотреть содержимое диска.

RENAME - внутренняя

ФУНКЦИЯ: Переименование файла.

ФОРМАТ:

```
rename [d:][path]filename[.ext]filename[.ext]
```

```
ren [d:][path]filename[.ext]filename[.ext]
```

ПРИМЕРЫ:

```
rename file1 file2
```

```
ren newfile.txt oldfile.txt
```

Команда RENAME (или REN) служит для изменения имени и/или расширения файла. Это одна из наиболее удобных и часто используемых команд. Чтобы изменить имя файла, наберите RENAME, спецификацию файла и его новое имя. Например, для замены имени файла NEWFILE.TXT на OLDFILE.TXT вводится команда:

```
C:\>rename A:\newfile.txt oldfile.txt
```

<= Enter

Шифр устройства перед новой спецификацией файла игнорируется. Если в командной строке перед новым именем указан спецификатор пути, то выдается сообщение об ошибке. При введении команды допускается использовать замещаемые символы.

RESTORE - внешняя

ФУНКЦИЯ: Восстановление файлов с гибких дисков на жесткий диск компьютера.

ФОРМАТ:

```
restore d:[d:][path][filename[.ext]][/s][/p]
restore d:[d:][path][filename[.ext]][/s][/p][/B:mm-dd-yy]
[/a:mm-dd-yy][/m][/n][/l:time][/e:time]
```

ПРИМЕРЫ:

```
restore A: \subdir1\file.doc
restore A: \subdir2
restore A: \subdir3 /s
restore A: \subdir4\*.doc/p
```

Команда восстанавливает файлы, полученные с помощью программы BACKUP. RESTORE не применяется для файлов другого типа и является внешней командой MS - DOS. Это означает, что перед обращением к ней необходимо убедиться в доступности файла RESTORE.COM.

Файл должен находиться на одном из системных дисков. В ниже приведенных примерах принимается, что он находится на жестком диске.

Восстановление файла

При восстановлении файлов на жесткий диск набирается RESTORE и шифр устройства, на котором находятся копии восстанавливаемых файлов. В командной строке также можно указать спецификатор пути директория, в который записываются восстановленные файлы.

Если спецификатор опущен, то файлы записываются в рабочий директорий рабочего диска. Можно указать имя восстанавливаемого файла. Если имя файла опущено, то восстанавливаются все файлы указанного (или рабочего) директория. При введении RESTORE на экране появляется запрос типа: "Установите дискету с файлами на требуемый дискетов". При нажатии любой клавиши процесс восстановления будет запущен. Предположим, что файл FILE1.DOC нахо-

дится в директории SUBDIR1. Копия этого файла (полученная с помощью BACKUP) находится на дискете A:. Чтобы восстановить файл с дискеты A: на диск C: введем команду:

```
C:\>restore A:\subdir1\file1.doc  
<= Enter
```

Insert backup diskette 01 in drive A: - установите восстанавливаемую дискету номер 01 на дисковод A: .

Strike any key when ready - нажмите любую клавишу, когда будете готовы.

Нажимаем любую клавишу:

```
*** Files were backed up 12/11/1998 *** - файлы были восстановлены до 12/11/1998.
```

```
*** Restoring files from diskette 01 *** - восстановленный файл из диска 01.
```

```
\subdir1\file1.doc
```

```
C:\>
```

При введении параметров FILENAME и .EXT допускается использование замещаемых символов. В этом случае восстанавливаются все файлы, соответствующие указанной спецификации.

Восстановление файлов поддиректория

Следующая команда восстанавливает все файлы поддиректория SUBDIR2 на дискете A:. Спецификатор пути этого поддиректория SUBDIR1 \SUBDIR2:

```
C:\>restore A:\subdir1\subdir2  
<= Enter
```

Восстановление файлов поддиректориев, вложенных в указанный директорий

При введении ключевого параметра /S восстанавливаются все файлы указанного директория, а также файлы всех входящих в него поддиректориев. В следующем примере восстанавливаются файлы директория SUBDIR3 и файлы подчиненных ему поддиректориев:

```
C:\>restore A:\subdir3/s  
<= Enter
```

Insert backup diskette 01 in drive A:
Strike any key when ready

```
*** Files were backed up 12/11/1988 ***  
*** Restoring files from diskette 01 ***
```

Восстановление выбранных файлов (ключевой параметр /P)

Нецелесообразно восстанавливать файл, который корректировался после того, как была получена его копия (т.е. при восстановлении вы получите файл, в котором отсутствуют все внесенные изменения). При введении в командную строку параметра /P обработчик команды производит проверку, корректировался ли восстанавливаемый файл после того, как была получена его копия. Если он корректировался, то на экран посылается предупреждающее сообщение и дополнительный вопрос на его восстановление. При ответе "N" процесс восстановления отменяется и управление возвращается MS - DOS. При ответе "Y" восстанавливается удаленный ранее не откорректированный файл.

RMDIR - внутренняя

ФУНКЦИЯ: Удаление поддиректория.

ФОРМАТ:

```
rmdir [d:]path
```

```
rd [d:]path
```

ПРИМЕРЫ:

```
rmdir \write
```

```
rd B:\programs\business
```

Команда RMDIR (ReMove DIrectory) служит для удаления поддиректория. Команду можно вводить, как RMDIR, и как RD. Перед удалением поддиректория необходимо удалить все имеющиеся в нем файлы. Эта мера предосторожности страхует от случайной потери информации на диске. В командной строке можно указать шифр устройства (например, C: или A:), обозначающий диск, на котором находится удаляемый поддиректорий. При отсутствии шифра устройства принимается, что поддиректорий находится на рабочем диске. В командной строке обязательно указывается спецификатор пути удаляемого поддиректория.

К примеру, для удаления поддиректория WRITE вводится команда:

```
C:\>rmdir \write  
<= Enter
```

Поддиректорий находится на рабочем диске и расположен на уровень ниже корневого директория. В следующем примере удаляется поддиректорий \BUSINESS:

```
C:\>rd B:\programs\business  
<= Enter
```

Поддиректорий расположен на диске B:. Он находится на уровень ниже поддиректория \PROGRAMS, который, в свою очередь, находится на уровень ниже корневого директория диска. С помощью RMDIR нельзя удалить рабочий и корневой директории диска.

SYS - внешняя

ФУНКЦИЯ: Переносит системные файлы на указанный диск.

ФОРМАТ:

sys D:

ПРИМЕР:

sys B:

sys A:

Системные файлы - это два "скрытых" файла, составляющие ядро MS - DOS. Наличие этих файлов на диске нельзя проверить командой DIR (поэтому они и называются скрытыми). Файлы должны располагаться в определенном месте диска в строго установленном порядке. Обычно с диска, на котором находятся эти файлы, производится загрузка операционной системы. Команда SYS переписывает системные файлы на указанный диск. Диск - получатель должен быть либо пустой отформатированной дискетой, либо отформатированным жестким диском. В следующем примере система находится на дисководе A:. Наберите SYS и шифр диска, на который переносятся файлы:

```
A:\>sys C:  
System transferred - система перенесена.
```

В последних версиях DOS системные файлы переносятся и при форматировании дискеты с параметром /S.

TREE - внешняя

ФУНКЦИЯ: Отображение дерева директориев на указанном диске.

ФОРМАТ:

tree [d:]/[f]

ПРИМЕРЫ:

tree

tree B: /f

При введении команды на экране появляется описание дерева директориев указанного диска. Каждый директорий (гибкого или жесткого диска) обозначен именем и полным спецификатором пути директория. Поддиректории объединяются в группу относительно родительского директория.

TREE является внешней командой MS - DOS. Это означает, что перед обращением к ней необходимо убедиться в доступности файла TREE.COM. Чтобы воспользоваться командой, наберите TREE и шифр просматриваемого диска. По умолчанию (если шифр устройства опущен) обрабатывается рабочий диск. При введении ключевого параметра /F для каждого директория выводится список содержащихся в нем файлов.

TYPE - внутренняя

ФУНКЦИЯ: Выдача содержимого файла на экран.

ФОРМАТ:

type [d:][path]filename[.ext]

ПРИМЕР:

type B:\letter.txt

Команда TYPE служит для вывода файлов на экран. Обычно она применяется для текстовых файлов. При попытке просмотреть двоичный файл, можно получить непредсказуемый результат.

Чтобы воспользоваться командой, наберите TYPE и спецификацию просматриваемого файла. Указанный файл считывается в память и выводится на экран:

```
C:\>type B:\letter.txt
```

```
<= Enter
```

Если файл состоит более, чем из 23 строк, то его содержимое будет двигаться вдоль экрана. Чтобы остановить движение, нажмите Ctrl + Scroll Lock, чтобы возобновить - нажмите любую клавишу.

Чтобы вывести просматриваемый файл на печать, перед введением команды нажмите клавиши Ctrl + PrtSc.

При введении этой команды запрещается использовать замещаемые символы.

XCOPY - внешняя

ФУНКЦИЯ: Копирование файлов.

ФОРМАТ:

`xcopy [d:] [path] filename [.ext] [d:] [path] [filename] [.ext] [/a] [/d:mm-dd-yy] [/e] [/m] [/p] [/s] [/v] [/w]`

ПРИМЕР:

`xcopy *.* A: /e /s /a`

Команда XCOPY представляет собой расширенную версию команды COPY. Она позволяет:

1. Выборочно копировать архивные файлы (с архивным атрибутом).
2. Выборочно копировать файлы по маркеру даты.
3. Копировать файлы из поддиректория, вложенного в указанный директорий.

Копирование поддиректориев

Пусть \BOOK - рабочий директорий диска C:, а \COMMANDS - поддиректорий \BOOK. При использовании COPY нельзя скопировать файлы директория \BOOK и поддиректория \COMMANDS одной командой. Введение ключевого параметра /S в командной строке XCOPY позволяет копировать содержимое указанного директория вместе с содержимым всех входящих в него поддиректориев. XCOPY является внешней командой MS - DOS. Это означает, что перед обращением к ней необходимо убедиться в доступности файла XCOPY.COM. Введите команду:

```
C:\>xcopy *.* A: /s
<= Enter
```

Reading source file(s) - чтение файла источника.

```
ch1.doc
ch2.doc
commands\cmd1.doc
commands\cmd2.doc
```

4 File(s) copied

При копировании на целевую дискету полностью переносится копируемая структура (т.е. организуются директорий и поддиректорий, файлы которых копируются). Эту особенность очень удобно использовать при копировании многоуровневых структур. Еще одна особенность ХСОPY заключается в том, что копирование группы файлов с помощью ХСОPY производится быстрее, чем с помощью СОPY, так как ХСОPY в процессе работы считывает в память максимально возможное количество исходных файлов и одним приемом переносит их на диск (а не по одному, как СОPY), что значительно сокращает время операции.

Копирование по дате

Введение ключевого параметра /D позволяет копировать файлы, организованные или обработанные в течение указанного дня или позже.

Дата задается в стандартном формате:

```
C:\>xcopy *.* A: /s /d:6-29-88
```

```
Reading source file(s)  
commands\cmd2.doc
```

1 File(s) copied

Копирование архивных файлов

ХСОPY можно использовать для копирования архивных файлов, т.е. файлов с архивным атрибутом. Для этого в командную строку вводится параметр /A. Параметр сохраняет значение атрибута исходного файла. При введении параметра /M архивный атрибут в процессе копирования обнуляется.

ОБОЛОЧКА NORTON COMMANDER

В этом разделе будет описана работа с программой Norton Commander (NC), которая позволяет легко и просто работать с файлами программ и различными каталогами. Программа NC является прекрасным примером программной оболочки, которая может работать на основе операционной системы (ОС) DOS или Windows и позволяет пользователю существенно упростить общение с компьютером по сравнению с работой только в рамках какой-нибудь ОС. Новая пятая версия NC существенно расширена по сравнению с предыдущими реализациями этой программы и позволяет активно использовать мышь.

В программу включены новые возможности такие, как резка на части и последующая сборка частей файлов, новая программа работы с терминалом, позволяющая соединять два компьютера по телефонной линии. Расширены возможности по копированию и переносу файлов и многое другое. Новый графический интерфейс и активная работа мышки максимально приближают NC к графическим оболочкам типа Windows. По сути, NC аналогичен Проводнику Windows 98 или Диспетчеру Файлов Windows 3.1.

Содержание раздела делится на три главы. Первые две главы посвящены изложению основных возможностей NC, которые наиболее часто используются в работе. В третьей главе описаны дополнительные, как бы встроенные в NC, возможности - утилиты, в частности, по работе с жесткими дисками компьютера.

ОСНОВНЫЕ СВЕДЕНИЯ О NC

Прежде чем переходить к описанию NC, еще раз кратко поясним основные понятия, с которыми работает любая программа компьютера, - это файл и директория или каталог. Понятие файл можно определить следующим образом - область постоянной памяти компьютера, которой тем или иным способом задано определенное имя и, возможно, содержащая некоторую информацию. Могут быть и пустые файлы, когда имя задано, а информация в нем еще не записана, хотя отсутствие информации - это тоже информация. В каждой компьютерной программе имеются свои способы и приемы создания файлов и записи в них определенной информации.

Обычно программа сама определяет, где на свободном месте жесткого диска - постоянной памяти компьютера - создать файл, и для пользователя само расположение файла чаще всего не представляет интереса. Важно лишь, что файл создан, и программа - редактор или какая-нибудь другая программа может его найти в любой нужный момент. Наличие файла на диске можно просмотреть и стан-

дартной командой DIR в операционной системе DOS, которая выводит на экран содержимое диска или каталога.

Имя файла обычно состоит из двух частей - собственно имени и расширения. Имя, в общем, может быть произвольным и определять название программы. Расширения подчинены определенным условиям и определяют типы файлов, которых насчитывается несколько сотен. Например, расширение EXE или COM определяют исполняемые файлы, содержащие определенную программу, непосредственно выполняемую компьютером, если задать его имя в командной строке DOS и нажать Enter.

Примером такого файла является файл nc.exe, содержащий программу оболочку Norton Commander. Расширения TXT определяют текстовые файлы в формате DOS, а DOC - тоже текстовые файлы, но обычно в формате редактора Word. Просмотреть содержимое текстовых файлов и создавать их всегда можно с помощью соответствующих редакторов. Расширения BMP или GIF говорят о графических файлах в различных форматах, т.е. файлов, внутри которых находятся рисунки, просмотреть которые можно специальными программами графических редакторов. Расширение BAT определяет командный или пакетный файл, внутри которого записана определенная последовательность вызова исполняемых файлов с программами.

Понятие каталог или директория представляет собой второе основное понятие, с которым работает компьютер. Каждая большая программа состоит из десятков и даже сотен файлов, а для ее запуска обычно нужен только один управляющий файл. Например, в NC нужно запускать файл nc.exe, а несколько десятков файлов, входящих в состав программы, выполняют вспомогательную роль. В нужный момент программа NC сама обращается к ним и использует их возможности.

Если на компьютере, т.е. в его постоянной памяти на жестком диске записаны десятки подобных программ, то общее число файлов составляет несколько тысяч. Найти нужный в данный момент будет достаточно сложно и займет немалое время, даже просматривая их в NC. Поэтому файлы группируются внутри каталогов, которые объединяют их по определенному признаку. Например, файлы NC находятся в каталоге с именем NC, файлы системы DOS - в каталоге DOS и т.д.

Большинство программ, в том числе и NC, имеют определенные возможности для создания каталогов с заданным именем и для помещения в них необходимых файлов определенной программы. Таким образом, каждая программа, состоящая из десятков файлов, находится в своем каталоге и найти нужный файл внутри каталога значительно проще.

Запуск Norton Comander

Наиболее простой способ запуска NC состоит из набора в командной строке операционной системы DOS команды NC с последующим нажатием клавиши Enter, если каталог NC, содержащий Norton Comander, описан в пути (оператор PATH) файла AUTOEXEC.BAT. Обычно этот каталог называется NC или NC с номером версии программы. Если каталог NC не описан в файле AUTOEXEC.BAT, то, используя методы работы с операционной системой DOS, надо войти в этот каталог и только тогда набрать на клавиатуре команду NC и нажать Enter.

На некоторых компьютерах в некоторых ситуациях, связанных с нехваткой оперативной памяти (ОП), быстрый просмотр больших файлов при помощи утилит просмотра NC оказывается невозможен. С этой целью в новой 5-й версии предусмотрены две модели памяти, вызываемые из командной строки. В каждом из этих режимов ограничивается число файлов, одновременно отображаемых на каждой панели Norton Commander (не более 700). NC может запускаться из командной строки DOS со следующими ключами-опциями (параметрами):

nc /s

или

nc /t

После набора команды следует нажать Enter. Система управления памятью, используемая в новой версии Norton Commander, сдвигает ядро программы (файл NCMAIN) в дополнительную или расширенную память для запуска внешних утилит - вспомогательных программ. Если ОП компьютера менее 1 Мбайта, то сдвиг осуществляется на жесткий диск в специальный SWAP файл, расположенный в каталоге временных файлов (TEMP, TMP, NC, NORTON), описанный в одной из переменных Среды (SET) файла AUTOEXEC.BAT. Например:

set = tmp

При запуске Norton Commander с гибкого диска в системе с 1 Мбайт ОП для достижения максимальной производительности надо использовать одну из этих переменных, указав в ней каталог на жестком диске. В противном случае, Norton Commander попытается создать этот файл на гибком диске, что существенно снизит скорость

работы программы. Вообще для нормальной работы всех утилит просмотра Norton Commander необходимо не менее 512К оперативной памяти, а некоторые из них требуют дополнительной памяти при работе в режиме быстрого просмотра.

Вид экрана в Norton Comander

Экран компьютера после запуска NC представляет собой два голубых листа или панели, переход между которыми, т.е. перевод экранного курсора с одной стороны на другую осуществляется клавишей Tab. На каждом из листов имеется три столбика с названиями каталогов - директорий и файлов. Имена директорий на листах показываются большими буквами, а имена файлов - маленькими. Светлая полоса на одном из листов - экранный курсор, управление которым на каждой панели осуществляется клавишами со стрелками или клавишами Page Up и Page Down, End и Home. Панель, на которой находится курсор, называется активной. Общий вид экрана компьютера при работе в NC показан на рис.1.

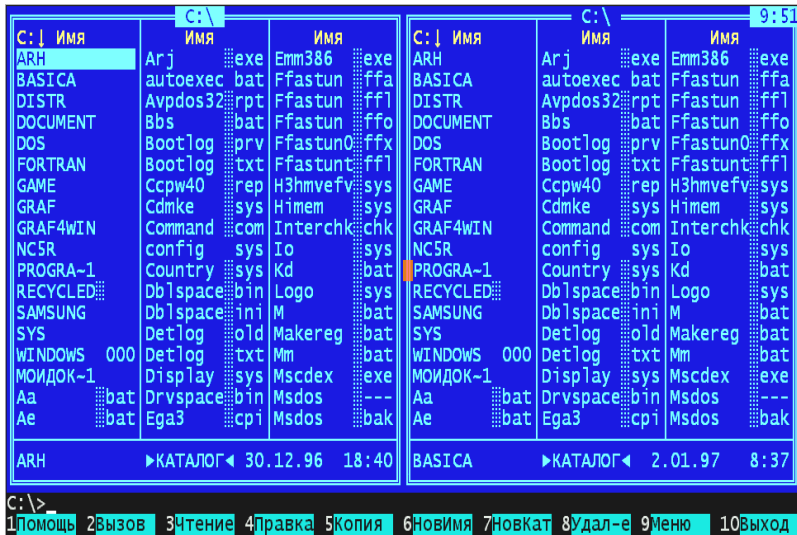


Рис.1. Общий вид панелей на экране в Norton Commander.

Вверху каждого листа показан диск, например, C:\ или путь к каталогу, как, например, на рис.2 - NC5R, содержимое которого показано на соответствующей панели. Если указан только диск, как на рис.1, то на панели показан главный каталог этого диска. Переход

между директориями осуществляется движением курсора. Чтобы выделить директорию или файл надо поставить на него экранный курсор. Чтобы войти внутрь директории, надо выделить ее и нажать Enter. Название этой директории переходит наверх панели, а на самой панели будет показано ее содержимое. Для выхода из директории необходимо поставить курсор в самый верх экрана любой панели, где имеется двоеточие, показанное на рис.2 и снова нажать Enter.

Внизу экрана показана строка с цифрами и словами - панель управления или подсказка, служащая для определения действий функциональных клавиш, работа которых будет рассмотрена ниже. Между панелями NC и подсказкой для функциональных клавиш имеется строка, в начале которой стоит большая буква, например, C:\> - имя диска и мигает горизонтальная черточка - командный курсор. Эта строка называется командной и служит для задания и выполнения стандартных команд операционной системы DOS. После ввода каждой команды следует нажать клавише Enter.

Большинство команд в NC выполняются с помощью функциональных клавиш или различных комбинаций клавиш. Ниже будут приведены их примеры, когда две клавиши записаны друг за другом через знак "+". Такая комбинация означает, что сначала нужно нажать и удерживать в нажатом состоянии первую клавишу, а затем нажать и отпустить вторую и только после этого отпустить первую из них.

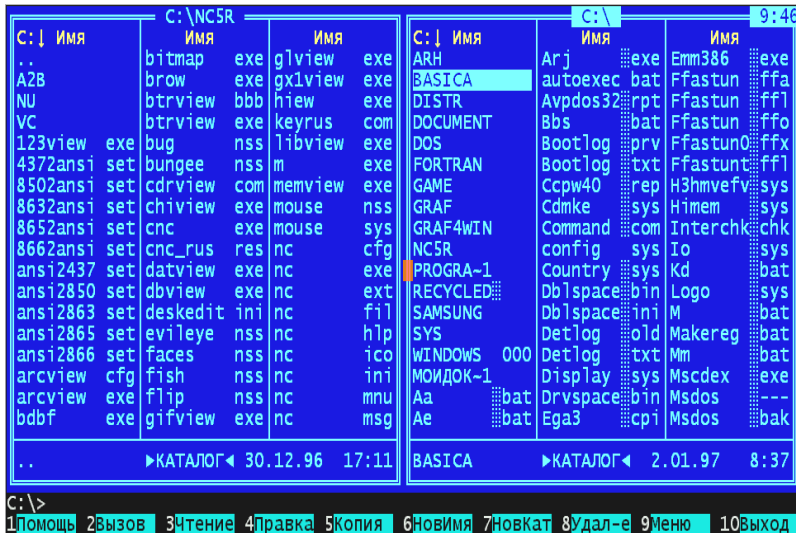


Рис.2. Вид экрана, когда на левой панели показана директория NC5R.

В частности, для работы с панелями NC используются следующие комбинации клавиш:

Ctrl + U - команда переключает позиции панелей NC - меняет их местами.

Ctrl + O - команда включает или выключает на экране обе панели, оставляя только командную строку и строку подсказки функциональных клавиш NC.

Ctrl + P - включает и выключает неактивную панель, не меняя положения и состояние активной панели NC.

Ctrl + B - включает и выключает подсказку для функциональных клавиш внизу экрана.

Ctrl + R - повторное считывание каталога или дерева каталогов, т.е. обновление содержимого панелей.

Ctrl + N - включает, как и клавиша F9, Главное меню NC сверху экрана компьютера.

Ctrl + Q - вывод на неактивную панель информации о директории, выделенной на активной панели.

Ctrl + L - вывод на неактивную панель информации о жестком диске и его составе.

Кроме того, имеется еще одна очень полезная комбинация клавиш - Ctrl + Enter, которая переносит в командную строку имя файла, выделенного экраным курсором. Эту комбинацию можно использовать для запуска исполняемого файла с параметрами - опциями, которые пишутся с помощью клавиатуры в командной строке через пробел после имени файла. Такие же функции выполняет и комбинация Ctrl + J. Если надо поставить какое-то имя файла, как параметр для исполняемого файла, то сначала необходимо поставить курсор на исполняемый файл и нажать Ctrl + Enter, переведя его в командную строку. Затем нажать клавишу Пробел, поставить курсор на вспомогательный файл и снова нажать Ctrl + Enter, поставив имя второго файла вторым в командной строке, и только после этого нажать Enter для выполнения заданных команд.

Для запуска любого исполняемого файла без параметров достаточно поставить на него курсор (выделить подсветкой) и нажать Enter. Комбинация клавиш Ctrl + M - действует так же, как и клавиша Enter, позволяя входить или выходить из директорий и запускать файлы.

В новой версии заметно легче выполнять операции копирования и переноса файлов при помощи мыши. Теперь необходимые файлы и каталоги можно выделить, поставив на них курсор мышки, и перетащить их для копирования с одной панели на другую, удерживая при этом в нажатом состоянии левую кнопку мыши. Если в процессе пе-

ретаскивания удерживать нажатой клавишу ALT, то файлы будут перемещены на новое место в другую директорию на другой панели. Кроме этого, файлы можно перетащить на кнопку панели управления - подсказку функциональных клавиш для вызова соответствующей файловой операции, например, копирования или переименования файлов и директорий.

Все эти улучшения и дополнения программы NC заметно повысили и расширили ее общие возможности, а также сделали более удобной в использовании на современных компьютерах.

ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ В NC

Основные сочетания клавиш в NC

Ниже приведен список функциональных клавиш, их комбинаций и назначение для выполнения различных команд и операций в программе NC. Под комбинацией клавиш типа Ctrl + F2 понимаются следующие действия: нужно нажать и удерживать первую из этих клавиш, а затем нажать и отпустить вторую и только после этого можно отпустить первую клавишу. Нажатие на любую клавишу должно быть кратким, в противном случае может включиться режим повтора символов или команд и вместо одного символа или действия будет выполнено несколько, что в некоторых случаях совершенно недопустимо.

F1	Вызов справки по NC. Может использоваться на любом этапе работы программы NC.
F2	Вызов меню пользователя, если оно создано в файле nc.mnu с помощью возможностей NC.
F3	Вызов встроенной в NC программы просмотра содержимого файлов.
F4	Вызов встроенного в NC редактора файлов.
F5	Вызов режима копирования файлов и директорий.
F6	Вызов режима переноса и переименования файлов и директорий.
F7	Создание новых директорий в позиции экранного курсора.
F8	Стирание выделенных файлов и директорий.
F9	Вызов Главного Меню NC.
F10	Выход из NC.
Ctrl + F1	Выключение и включение левой панели NC.
Ctrl + F2	Выключение и включение правой панели NC.
Ctrl + F3	Сортировка файлов на активной панели по имени в алфавитном порядке.
Ctrl + F4	Сортировка файлов и директорий на активной панели по типу в алфавитном порядке.
Ctrl + F5	Сортировка файлов и директорий на активной панели по времени создания.
Ctrl + F6	Сортировка файлов на активной панели по размеру.
Ctrl + F7	Несортированный показ файлов и директорий.
Ctrl + F8	Синхронизация каталогов на двух компьютерах.

Ctrl + F9	Печать выделенного файла.
Ctrl + F10	Деление (резка) на несколько частей и сборка файлов.
Alt + F1	Вызов списка дисков на левую панель.
Alt + F2	Вызов списка дисков на правую панель.
Alt + F3	Вызов программы быстрого просмотра содержимого файлов.
Alt + F4	Вызов альтернативной программы, например, NE для редактирования содержимого текстовых файлов.
Alt + F5	Создание архива файлов.
Alt + F6	Раскрытие выделенного архива файлов.
Alt + F7	Поиск файла по образцу.
Alt + F8	Вызов журнала команд для повтора ранее выполненной команды.
Alt + F9	Переключение различных режимов работы экрана.
Alt + F10	Вызов на экран дерева каталогов.

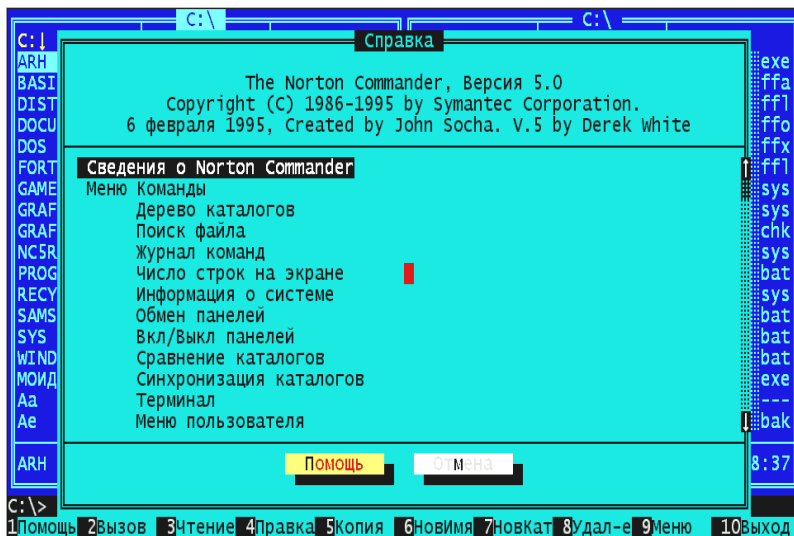


Рис.3. Оглавление справки NC.

Все функциональные клавиши могут нажиматься в комбинации с клавишей Shift, которая позволяет использовать дополнительные возможности по выбору файлов и каталогов. Многие из приведенных комбинаций клавиш будут далее рассмотрены более подробно с конкретным указанием последовательности действий, выполнять кото-

рые для получения требуемого результата нужно очень аккуратно. Неправильные, непоследовательные действия при выполнении операций в некоторых случаях могут привести к нежелательному стиранию файлов и каталогов или к порче содержимого текстовых и даже программных файлов.

На любом этапе работы с NC можно использовать клавишу F1 для получения краткой справки, часть оглавления которой при работе в главном каталоге показана на рис. 3. После нажатия в главном каталоге клавиши F1 появляется оглавление справки, а экранный курсор стоит на первой верхней позиции - на первом параграфе оглавления.

Клавишами управления курсором можно перемещаться по оглавлению и, нажав Enter в нужном месте, вывести на экран содержимое справки по выбранному разделу. Нажав клавишу Enter в начале справки на первом параграфе, можно последовательно выводить на экран содержимое ее разделов, постоянно нажимая Enter. Внутри окна справки по выбранному разделу можно листать текст клавишами со стрелками вверх и вниз. Эта операция оказывается доступна, если справа экрана будет показана линейка прокрутки, управлять которой можно и с помощью мышки.

Выделение группы файлов или каталогов

Выделить один файл или каталог можно, поставив на него экранный курсор. Для выделения группы файлов или каталогов нужно использовать клавишу Ins или Insert, которые последовательно выделяют группу объектов. Серые клавиши « + » и «-» на цифровой клавиатуре можно использовать для выделения или снятия выделения по маске. Эти клавиши выводят окно диалога, где надо задать выделяемые файлы по стандартному шаблону DOS - маске. Выделение будет выполнено в том каталоге, где находился экранный курсор. Серая звездочка “*” на цифровой клавиатуре инвертирует выделение, снимая выделение ранее выделенных файлов и выделяя те файлы, которые выделены не были.

Переключение дисков Alt + F1 или Alt + F2

Для переключения панелей NC между разными дисками, которые подключены на компьютере, используются команды, выполняемые комбинацией клавиш Alt + F1 или Alt + F2.

На левой или правой панели соответственно появляется список дисков, имеющихся на компьютере, а нужный из них выбирается клавишами со стрелками с последующим нажатием Enter. Вид экрана при включенном меню выбора диска на левой панели показан на рис.

4.

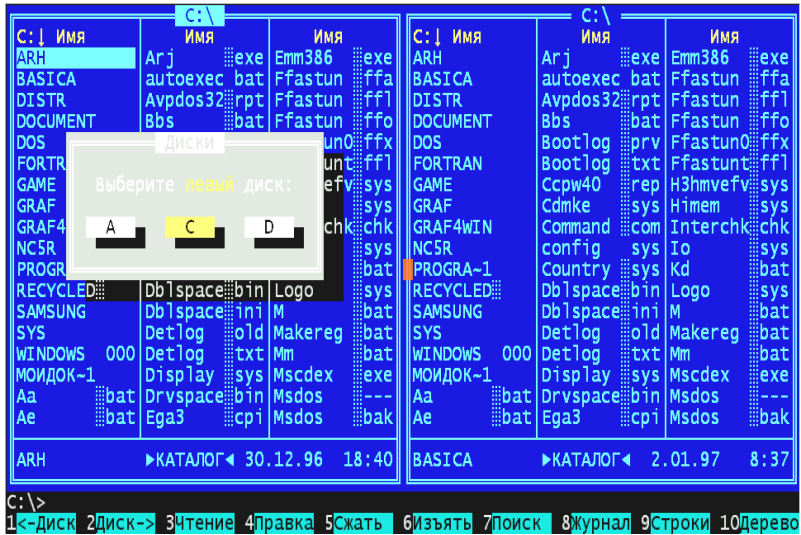


Рис.4. Вид панелей при выводе на левую панель списка дисков.

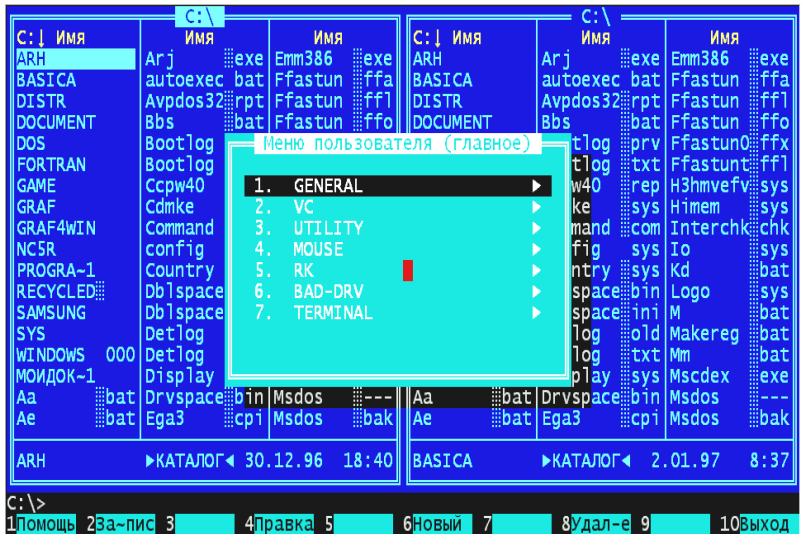


Рис.5. Вид панелей при вызове Меню пользователя клавишей F2.

Вызов меню пользователя - клавиша F2

Эта команда используется для создания, редактирования и поддержания определяемого пользователем **Меню** пользователя, доступ к которому осуществляется клавишей F2. При нажатии на эту клавишу, на экране появляется список заданных команд или подменю, выполнять которые можно, выбирая курсором нужную команду с последующим нажатием клавиши Enter.

Примерный вид такого **Меню** показан на рис.5. Эти меню могут создаваться, например, с помощью встроенного редактора NS. Поскольку использование **Меню** пользователя существенно облегчает работу с часто применяемыми программами, рассмотрим этот вопрос более подробно.

Нажмите клавишу F2 для вызова **Меню** пользователя и, когда его окно появится на экране, нажмите F6 для создания нового раздела меню и выберите нужный вариант - **Меню** или **Команда**. Выбор осуществляется нажатием пробела около желаемого варианта так, чтобы слева от него установилась точка.

Для создания команд выберите пункт **Команда** и нажмите Enter. В появившемся окне наберите, если нужно, номер команды и обязательно ее название, нажмите Enter и в новой строке укажите путь к файлу с нужной программой. Клавишей Tab перейдите на кнопку ОК внизу окна и нажмите Enter, чтобы вернуться к первоначальному окну **Меню** пользователя, в котором уже будет стоять набранное вами имя команды.

Теперь для сохранения сделанных изменений нужно нажать F2, а если это не сделано, то при нажатии на Esc или Enter будет выведено окно для подтверждения сохранения сделанных изменений.

Вместо номера команды можно использовать назначаемую клавишу, при нажатии на которую после вызова **Меню** будет выполняться заданная команда. После нажатия F2 для вызова **Меню** пользователя клавиша F4 позволяет редактировать заданные уже команды, а F8 стирать их. В любой момент можно нажать F1 для получения краткой справки о работе и создании этих Меню.

Вставить **Меню** или подменю в **Меню** пользователя можно точно так же, как и команду, выбрав пробелом пункт **Меню** после нажатия на клавишу F6. Нажмите далее Enter и введите имя подменю, перейдите клавишей Tab на кнопку ОК и снова нажмите Enter.

В **Меню** пользователя появится пункт с введенным названием подменю, справа от которого стоит треугольник, показывающий, что данный пункт не просто команда, а внутреннее **Меню** - подменю

пользователя. Для ввода команды в подменю нажмите клавишу Enter на его заголовке. При этом появится содержимое подменю, которое вначале будет пустым. Команды в это подменю можно вставить стандартным образом, описанным выше. В режиме редактирования команд в подменю также можно использовать любые функциональные клавиши, по которым внизу экрана приведена подсказка, в том числе клавишу F2 для сохранения сделанных изменений.

Существует два вида меню пользователя - **Главное**, файл которого хранится в каталоге, где находится NC, и **Локальное**, файл которого находится в текущем каталоге, т.е. там, где стоит курсор в момент нажатия F2. **Главное меню** пользователя вызывается, когда вы находитесь в главном каталоге, а **Локальное меню** - при нажатии F2 внутри подкаталога.

Таким образом, внутри любой директории можно создавать свое собственное меню пользователя. Вся информации о сделанных изменениях хранится в файле nc.mnu.

Просмотр содержимого файла - F3

При просмотре содержимого файла с помощью встроенной в NC программы просмотра надо выделить его и нажать клавишу F3. При нажатии Shift + F3 можно дополнительно выбирать, какой именно файл нужно просмотреть.

После выполнения команды просмотра на экране появляется содержимое файла, как показано на рис.6, перемещаться по которому можно клавишами стрелок или PgUp и PgDn. В режиме просмотра можно использовать клавишу F8 для выбора различных типов кодировок текста, что позволяет просматривать большое число файлов, сделанных разными редакторами.

Все утилиты просмотра, поддерживаемые Norton Commander версия 5.0, определены в файле NC.CFG. При необходимости пользователь может внести изменения в этот файл, добавив свои расширения файлов и связав их с определенными утилитами просмотра. Например, если какой-нибудь редактор создает файлы с расширением .DCC, то для использования утилиты просмотра текстовых файлов WPVIEW.EXE необходимо внести следующие изменения в соответствующий раздел файла NC.CFG:

```
~wpview.exe
$*.exe,p
$*.com,p
$*.dll,p
$*.dcc,p - строка, добавленная для обработки файлов *.dcc.
```

```

Текст: C:\CONFIG.SYS          Ст. 0          430 байт          100%
DEVICE = C:\WINDOWS.000\HIMEM.SYS
DEVICE = C:\WINDOWS.000\emm386.exe noems
rem d=64

DOS = HIGH,umb
rem FILES = 100

rem Samsung
rem DEVICEhigh=C:\SAMSUNG\SSCDROM.SYS /D:SSCD000 /v
DEVICEhigh=C:\SAMSUNG\SSCDROM.SYS /D:d

rem Panasonic
rem devicehigh=c:\sys\util\sound16a\drivers\cdmke.sys /d:d

devicehigh=C:\WINDOWS.000\COMMAND\display.sys con=(ega,,1)
country=007,866,c:\WINDOWS.000\COMMAND\country.sys
    
```

1Помощь 2>> 3 4Коды 5 6 7Поиск 8Формат 9Печать 10Выход

Рис.6. Вид экрана при просмотре содержимого файлов по клавише F3.

В режиме просмотра текста с помощью F3 можно выполнять следующие действия:

- ← - передвинуть курсор на позицию влево.
- - передвинуть на позицию влево вправо.
- ↑ - передвинуть на позицию влево вверх.
- ↓ - передвинуть на позицию влево вниз.
- PgUp - передвинуть вверх на страницу.
- PgDn - передвинуть вниз на страницу.
- HOME - начало файла.
- END - конец файла.
- F1 - вызов справки.
- F4 - переключение режима просмотра с формата ASCII на шестнадцатеричный и обратно.
- F7 - поиск по образцу.
- Shift + F7 - повтор поиска.
- F8 - переключить формат отображения текста на экране в форматах разных текстовых редакторов.
- F9 - печать файла.
- F10 или Esc - выход из режима просмотра и возврат в обычный режим работы.

Редактирование содержимого файла - F4

Для редактирования файла надо поставить на него курсор, т.е. выделить его и нажать клавишу F4. Если нужно явно задать имя файла, необходимо нажать Shift + F4 и в появившемся окне написать имя файла. Если файла с таким именем не существует, то после задания имени и нажатии Enter, он будет создан на жестком диске. Для редактирования файлов используется встроенный редактор NC, который находится в файле NCEDIT.EXE, а вид экрана показан на рис.7. В режиме редактирования при нажатии клавиши F1 появляется список клавиш, используемых для операций редактирования, который приведен ниже.

```

файл: C:\autoexec.bat      Строка 1      Столбец 1      63 985 свободно 64
@echo off
prompt $p$g
SET PATH=C:\WINDOWS.000;C:\WINDOWS.000\COMMAND;C:\WINDOWS;C:\WINDOWS\COMMAND;c:\

set sound=c:\util\sound16a
set galaxy=a220 i5 d1 k10 p530 t6
set blaster=a220 i5 d1 t4

lh turbokey
rem lh c:\sys\mouses\gmouse

rem Samsung
rem lh C:\samsung\MSCDEX.exe /D:SSCD000
lh C:\samsung\MSCDEX.exe /D:d

rem Panasonic
rem lh C:\WINDOWS.000\COMMAND\MSCDEX.EXE /D:MSCD0001 /M:30
rem lh C:\WINDOWS.000\COMMAND\MSCDEX.EXE /D:d

lh mode con codepage prepare=((866) C:\WINDOWS.000\COMMAND\ega3.cpi)
lh mode con codepage select=866
lh keyb ru, ,C:\WINDOWS.000\COMMAND\keybrd3.sys
lh c:\nc5r\nc
1Помощь 2Запись 3Блок 4Замен 5Копия 6Сдвиг 7Поиск 8Удал-е 9Печать 10Выход?

```

Рис.7. Вид экрана в режиме редактирования
файлов по F4.

Способы перемещения курсора:

- ← - передвинуть курсор влево на одну позицию.
- - передвинуть курсор вправо на одну позицию.
- ↑ - передвинуть курсор вверх на одну строку.
- ↓ - передвинуть курсор вниз на одну строку.
- ^← - передвинуть курсор влево на слово.
- ^→ - передвинуть курсор вправо на слово.

HOME - передвинуть курсор в начало строки.
END - передвинуть курсор в конец строки.
PgUp - передвинуть курсор вверх на страницу.
PgDn - передвинуть курсор вниз на страницу.
^HOME - передвинуть курсор в начало файла.
^END - передвинуть курсор в конец файла.

Здесь значок ^ обозначает клавишу Ctrl.

Удаление:

Back Space - удалить символ слева от курсора.
DEL - удалить символ над курсором.
^W - удалить слово слева от курсора.
^T - удалить слово справа от курсора.
^Y - удалить строку, на которой стоит курсор.
^K - удалить с позиции курсора до конца строки.
F8 - удалить выделенный блок или текущую строку.

Работа с блоками:

F3 - начало выделения блока. Затем переместите курсор на последнюю строку выделения и снова нажмите F3 для завершения выделения.

Shift + F3 - отмена выделения.
F5 - копирование выделенного блока в позицию курсора.
F6 - перемещение блока в позицию курсора.
Alt + F10 - Запись блока в конец файла.

Команды работы с файлами:

F1 - вызов справки редактора.
F2 - сохранение изменений, сделанных в файле.
Shift + F2 - сохранить изменения в другом файле.
F9 - печать содержимого файла.
Alt + F9 - включение и выключение резервного копирования файлов.

Alt + F5 - вставка содержимого другого файла в позицию курсора.

F10, Esc - выход из файла. Если при этом были сделаны изменения, то появится окно с сообщением о необходимости сохранения или не сохранения этих изменений.

Поиск и замена по образцу:

F4 - поиск заданного образа и замена на другой образ вперед, т.е. к концу файла от позиции курсора.

Shift + F4 - поиск и замена назад, т.е. к началу файла.

Alt + F4 - повторить поиск и замену.

F7 - поиск строки вперед.

Shift + F7 - поиск строки назад.

Alt + F7 - простор поиска строки.

Дополнительные команды:

Alt + F1 - отобразить текст в формате ANSI.

Alt + F2 - отобразить текст в формате ASCII.

Alt + F3 - вставить дату.

Alt + F6 - подсчет слов и строк.

Alt + F8 - переход к заданной строке.

Копирование файла или каталога - F5

Для копирования файла или каталога используются клавиши F5 или Shift+F5. На рис.8 показано, как выполняется копирование директории DOS из главного каталога на правой панели внутрь директории NC5R, содержимое которой выведено на левую панель.

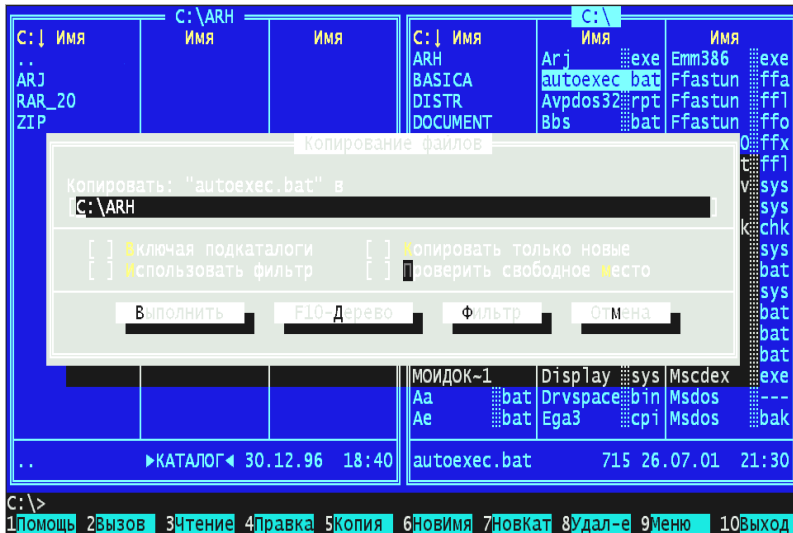


Рис.8. Вид экрана при копировании клавише F5.

Копировать можно несколько типов объектов:

1. Один выделенный объект - файл или директорию, на котором стоит курсор.
2. Выделенную клавишей Ins группу объектов - файлов или директорий.
3. Указанный явно объект в окне, которое появляется при нажатии комбинации клавиш копирования Shift + F5.

Каталог назначения, куда надо копировать, также можно задавать несколькими способами:

1. Установкой на неактивной (второй, куда копируют) панели нужного каталога или диска.
2. Явным заданием пути назначения в окне копирования с помощью режима **Дерева каталогов**.
3. Нажатием клавиш Shift + F5 также можно задавать путь копирования, как показано на рис.9.

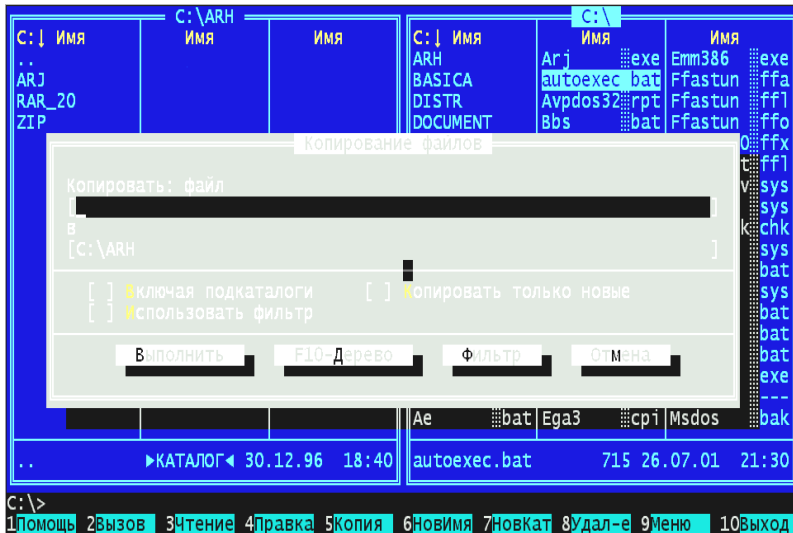


Рис.9. Вид экрана при копировании по клавише Shift + F5.

При копировании по клавише F5 экранный курсор должен находиться в каталоге источника, все объекты должны быть определены и путь копирования задан.

Кроме того, в окне копирования, которое появляется при нажатии F5, можно задавать следующие параметры:

1. **Включая подкаталоги** - для указания копирования файлов в подкаталогах.
2. **Использовать фильтр** - для указания файлов с указанной спецификацией.
3. **Копировать только новые** - для указания копирования новых файлов, которые отсутствуют по адресу назначения.
4. **Проверить свободное место** - для проверки свободного места для хранения файлов по адресу копирования.

Копирование в любой момент можно остановить клавишей Esc. При установке параметров копирования и переноса объектов в **Главном меню** NC, которое появляется на экране по клавише F9, может быть задан пункт **Подтверждение**, тогда перед копированием будет выводиться дополнительное окно для подтверждения операций.

Фильтры

Параметры **Фильтра** панелей, использовавшихся в Norton Commander 4.0, хранились в файле NC.CFG. В новой версии программы эти фильтры и фильтры операций хранятся в двоичном файле NC.FIL. Если пользователь мог вносить изменения в определения фильтров в версии 4.0, то эту операцию можно выполнить и в новой версии, используя встроенные возможности настройки фильтров.

Особенностью новой версии Norton Commander 5.0 является эффективная организация фильтрации необходимых файлов. Она позволяет не только отображать файлы в панелях в соответствии с определенными критериями, но и использовать настроенные необходимым образом **Фильтры** при выполнении операций копирования, перемещения, синхронизации, удаления и сжатия.

Фильтры операций позволяют пользователю отбирать файлы, соответствующие определенному имени, дате и времени создания, размеру и даже атрибутам. **Фильтры** панелей управляют тем, что показывается на панелях файлов. Такие расширенные возможности обычно были присущи специализированным программам поиска файлов, в то время, как новая пятая версия Norton Commander содержит их в самой основе программы, что позволяет считать Norton Commander одной из лучших программ управления файлами, которые работают на основе ОП DOS. Вид экрана NC при выводе окна **Фильтра** панелей показан на рис.10.

Команда **Фильтр** панелей доступна по клавише F9 из **Главного меню** NC в разделе **Левая - Правая**. Команда **Фильтр** операций по-

является во многих окнах различных операций, например, копирования. Кроме этого имеется ряд встроенных фильтров для использования в процессах поиска. Если выбрать пункт **Загрузить** окна **Фильтр**, то оказывается возможным отображение диалога **Фильтра** файлов, в котором имеются следующие разделы:

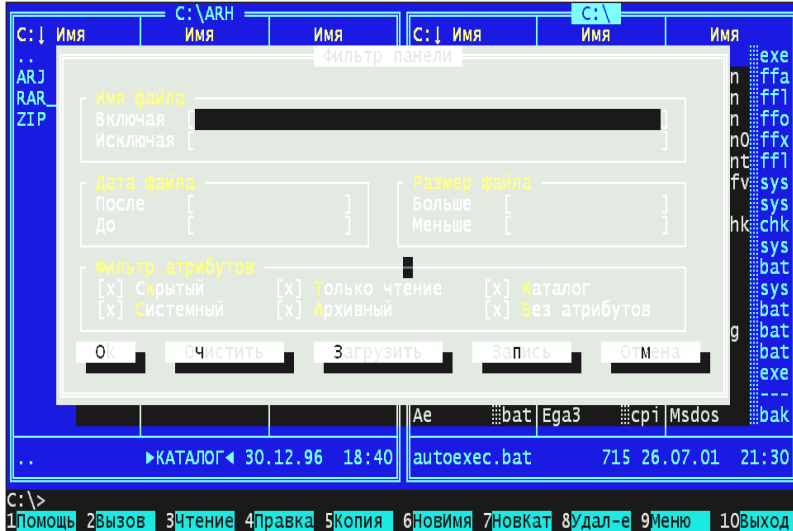


Рис.10. Вид экрана при установке фильтров панелей.

1. **Исполнение** - включает только программы и командные файлы.
2. **Файлы баз данных** - включают только базы данных.
3. **Электронные таблицы** - включают только файлы электронных таблиц.
4. **Документы** - включают только файлы разных текстовых редакторов.
5. **Графические** - включают только файлы с графикой.
6. **Архивы** - включают только файлы архивов типа ARC, ZIP, LZH, ZOO и некоторые другие.
7. **Пустые** - включают только файлы, которые могут быть созданы пользователем.

Переименование и перенос файла или каталога - F6

Эта команда позволяет перенести или переименовать файл или каталог и выполняется клавишей F6 или Shift + F6. Существует три

способа использования этой команды:

1. Переименование или перенос файла или каталога с помощью клавиши F6.
2. Перенос определенной выделенной группы файлов или каталогов с помощью F6.
3. Нажатием клавиш Shift + F6 можно задавать путь переноса выделенных объектов.

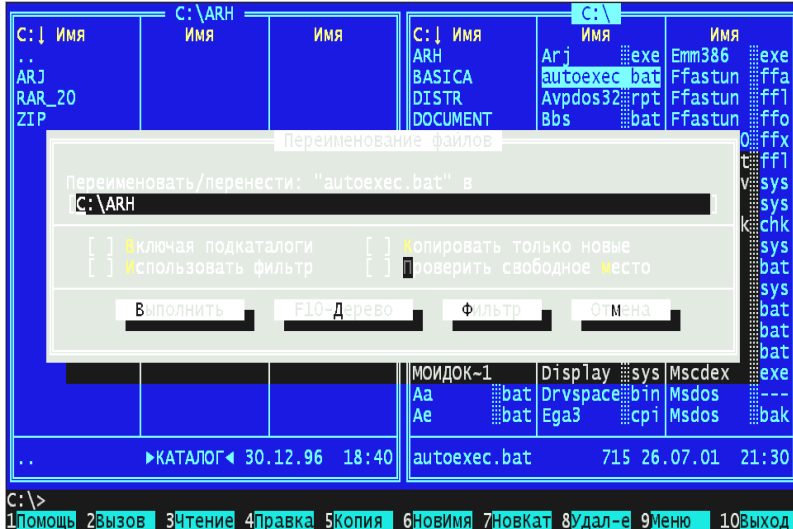


Рис.11. Вид экрана при переносе объектов по клавише F6.

Например, на рис.11 показан вариант переноса без переименования каталога DOS с правой панели во внутрь каталога NC5R, содержимое которого показано на левой панели.

На панели, которая вызывается нажатием F6, можно установить следующие параметры:

1. **Включая подкаталоги** - для указания переноса файлов в подкаталогах.
2. **Использовать фильтр** - для переноса файлов, удовлетворяющих спецификации фильтра, который также задается в этом окне.
3. **Копировать только новые** - для указания переноса только новых файлов.
4. **Проверить свободное место** - для проверки доступного свободного места на диске, куда перенесутся файлы.

Существует три способа для выбора того каталога, куда можно перенести файл:

1. При использовании Shift + F6 набрать имя каталога, куда надо перенести файл или каталог. Вид экрана показан на рис.12.
2. Нажать F10 для отображения дерева каталогов, из которого можно выбрать требуемый.
3. Выведите перед переносом на неактивной панели требуемый каталог.

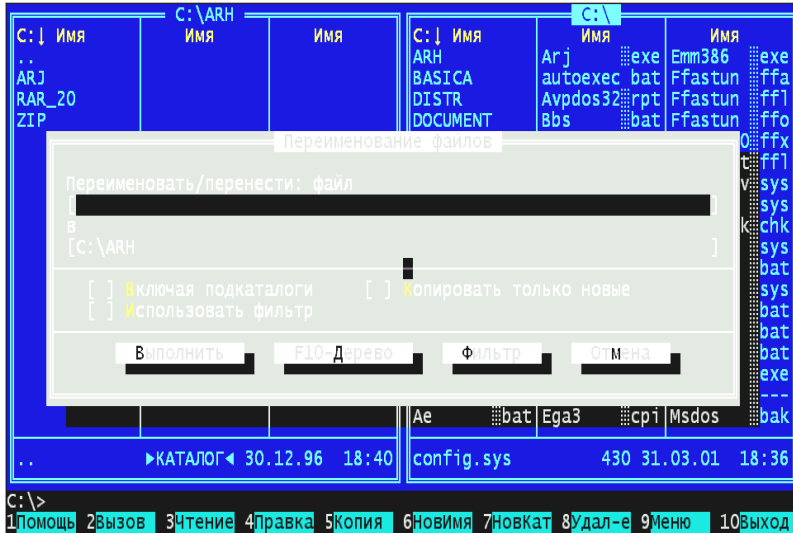


Рис.12. Вид экрана при нажатии комбинации клавиш Shift + F6.

Для переименования объектов в окне, которое появляется по F6, нужно задать новое имя и, если на неактивной панели установлен тот же каталог, что на активной произойдет переименование без переноса. Если на неактивной панели установлен другой каталог и задано новое имя объекта, произойдет перенос с переименованием. Если новое имя не задается, но каталоги на панелях различные, происходит только перенос. Для отмены переноса или переименования в любое время можно нажать клавишу Esc.

Если в установках параметров по F9 в меню **Команды - Конфигурация** задан пункт **Подтверждение**, то перед переносом будет выводиться дополнительное окно для подтверждения операции.

Например, на рис.13. показан вид экрана с переименованием директории DOS в DOS-1 без переноса, поскольку на обеих панелях

задан главный каталог.

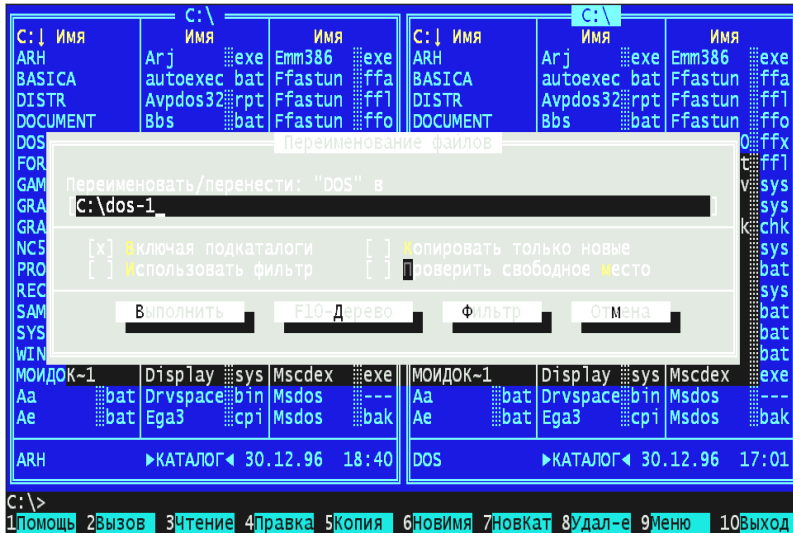


Рис.13. Вид экрана при переименовании файлов и каталогов по F6.

Создание каталога – клавиша F7

Эта команда выполняется клавишей F7 и позволяет создавать каталог в том месте, где перед нажатием на эту клавишу находился экранный курсор. Если курсор находился внутри каталога, то новый каталог будет подкаталогом этого каталога. При нажатии на F7 появляется окно, где надо задать имя каталога, которое может быть произвольным, но не более 8 символов и может иметь расширение, записанное через точку, но не более 3 произвольных символов.

Удаление файла или каталога - F8

Удалить файл или каталог, на котором стоит курсор, или выделенную группу можно нажатием клавиши F8. Существует три способа использования этой команды:

1. Можно удалить файл или каталог, на котором стоит экранный курсор.

2. Можно удалить выделенную клавишей Ins группу объектов - файлов, папок.
3. Нажатием Shift + F8 можно задавать группу удаления.

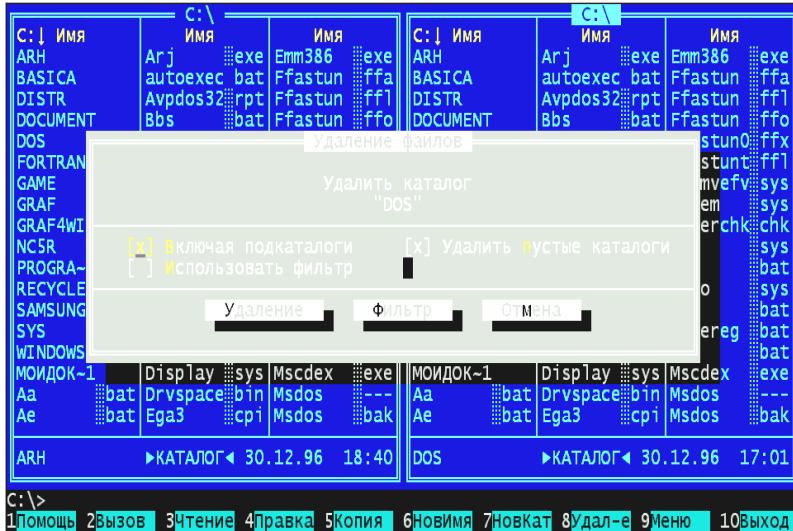


Рис. 14. Вид экрана при удалении объектов.

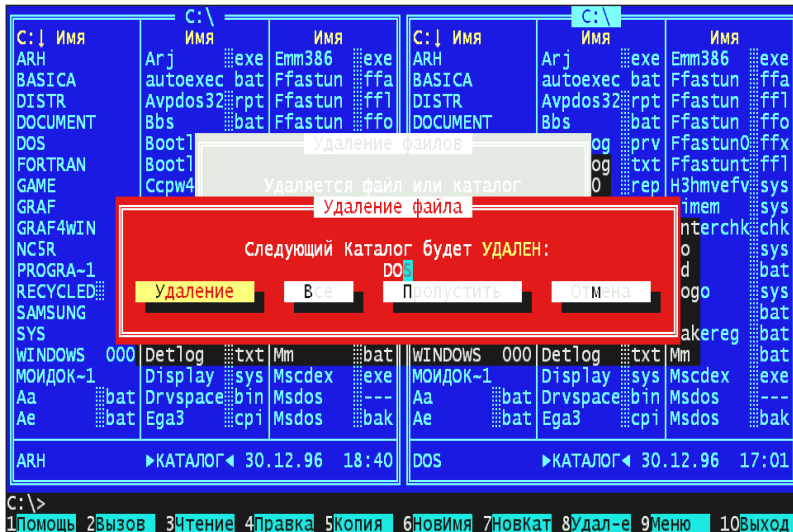


Рис. 15. Подтверждение на удаление объекта по клавише F8.

В окне, которое появляется при нажатии F8 и показано на рис.14, есть те же условия, которые имелись для копирования или переноса файлов - кроме одного: Удалить пустые каталоги - для удаления всех имеющихся пустых каталогов. Если удаляются файлы с параметром Только для чтения, то появляется дополнительное окно подтверждения удаления (рис.15).

Если удаляется группа файлов или каталог, то при нажатии Enter также появляется окно **Подтверждения**. Для подтверждения надо снова нажать Enter или выбрать пункт **Отмена** для отмены удаления. Пункт **Все** окна **Подтверждение** позволяет стереть все объекты, не выводя больше этого окна на экран. А пункт **Пропустить** позволяет не стирать один из выбранных ранее объектов.

Главное меню NC - клавиша F9

Главное Меню NC, вызываемое клавишей F9, обеспечивает доступ к ряду полезных команд. Все команды из этого меню описываются в отдельных экранах подсказки. В этом спускающемся меню имеется несколько групп команд:

1. **Команды** - содержит команды расширения возможностей для быстрой смены каталогов, поиска файлов, вызова последних команд, изменения числа строк на экране и получения информации о системе. Здесь же имеется пункт Конфигурация, который вызывает соответствующее окно диалога, позволяющее установить конфигурацию Norton Commander. Вид меню Команды показан на рис.16.

2. **Панели** (Левая или Правая) - эти команды воздействуют на **Левую** и **Правую** панели, меняя их местами, включая и выключая панели, сравнивая и синхронизируя каталоги на этих панелях. **Меню** раздела **Правая** показано на рис.17.

3. **Диск** - команды применяются для работы с диском, уборки диска, форматирования дискет и т.д., как показано на рис.18.

4. **Файл** - большая группа команд, позволяющая получать информацию о файлах и менять или устанавливать их атрибуты. Вид этого меню показан на рис.19.

Сравнение и синхронизация каталогов

Команда **Сравнение** доступна по клавише F9 в разделе **Команды** и позволяет сравнивать содержимое разных каталогов, находящихся на разных панелях NC. Если каталоги идентичны, NC выдает об этом сообщение. Различные файлы в этих каталогах выделяются желтым цветом. Команда сравнивает размер файла, дату и время создания, совпадения или различия имен файлов. Для синхронизации

каталогов можно использовать раздел меню **Синхронизация каталогов**, также находящуюся в **Главном меню Команды**. Команда **Синхронизации** делает два каталога идентичными.

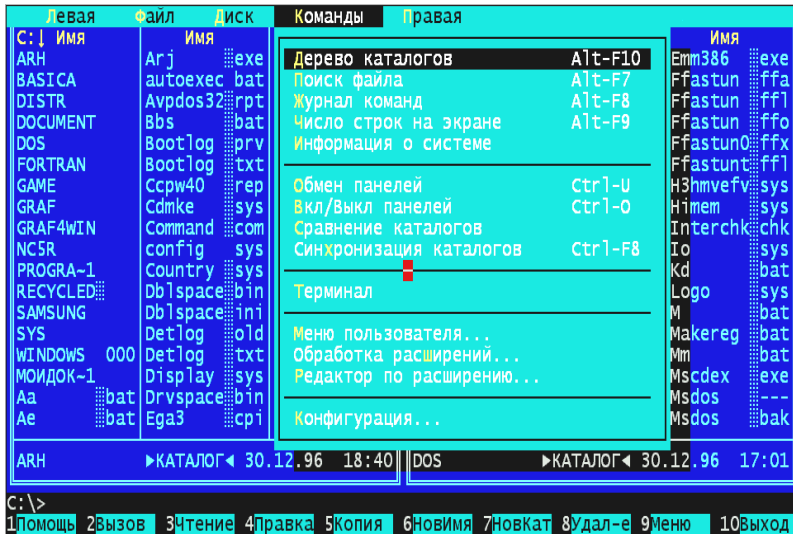


Рис.16. Вид меню **Команды** в Главном меню NC.

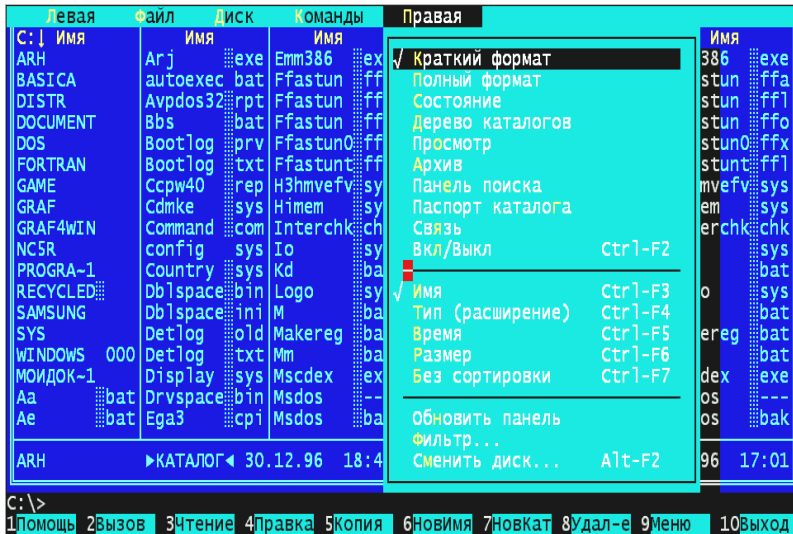


Рис.17. Вид меню **Панели** (левая - правая) в Главном меню.

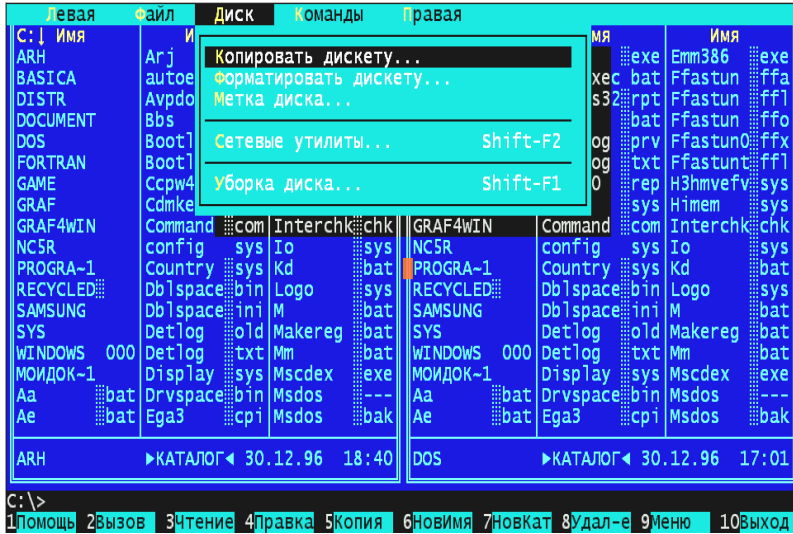


Рис.18. Вид меню Диск в Главном меню.

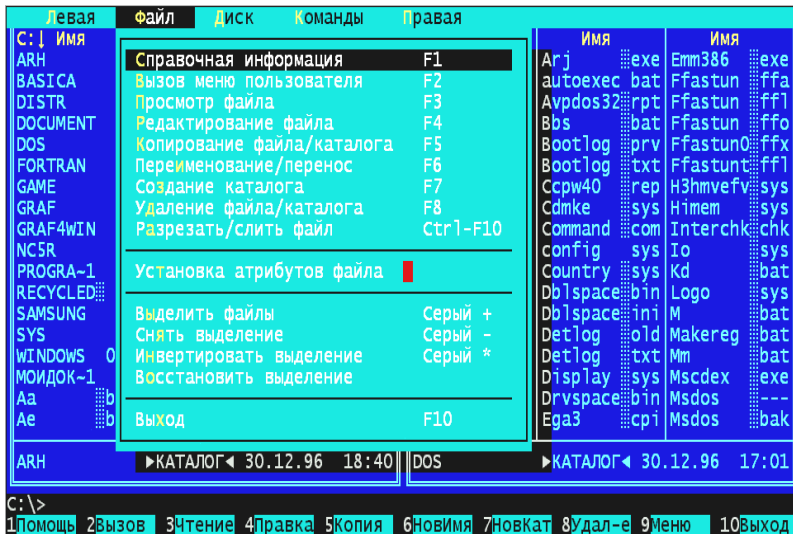


Рис.19. Вид Главного меню Файл по клавише F9.

Это бывает необходимо для получения двух одинаковых каталогов на основном и переносном компьютерах.

Работа с архивами Alt + F5 и Alt + F6

Для просмотра содержимого архива, т.е. архивного файла, на котором стоит курсор, выберете пункт меню **Архив**, появляющийся в **Главном меню Левая** или **Правая** при нажатии клавиши F9. На экране появится его содержимое, с которым можно работать, как с содержимым обычного каталога: просматривать текстовые файлы, копировать файлы в архив или из архива, можно удалять файлы и т.д., но нельзя запустить какое-либо приложение на исполнение, необходимо сначала извлечь его из архива.

Для сжатия - архивации файлов надо выделить их и нажать клавиши Alt + F5. Появится окно **Упаковка файлов** в архив, в котором надо указать любое допустимое имя архива и указать включать или не включать директории в архив, а после этого, для создания архива нажать клавишу Enter.

Для распаковки файлов из архива надо поставить курсор на этот файл и нажать Ctrl + F6. Появится окно **Распаковка архива**, в котором надо указать включать или не включать в число распакованных объектов директории. Распаковка или разархивация архива будет производиться на другую панель NC, если явно в окне распаковки не задан путь.

В Norton Commander версии 5.0 можно сжимать до 40000 файлов, используя различные программы сжатия, включая ZIP, ARC, ARJ, PAK, LHARC и ZOO. Можно просматривать содержимое файла, находящегося в архиве, с использованием клавиш F3 или Enter. В последнем случае можно редактировать содержимое архива, работая с ним, как с обычным каталогом - папкой.

Поиск файла - Alt + F7

Эта команда, вызываемая Alt + F7, проверяет текущий диск в поиске заданных файлов. Можно искать файлы по имени или файлы, содержащие заданный текст. Можно запускать программу поиска файлов Norton Commander из командной строки, набирая NCFF. Для получения информации об опциях и режимах работы этой команды, наберите в командной строке NCFF/? и нажмите Enter. Для поиска файла наберите в окне поиска, которое показано на рис.20, его имя (можно использовать символы шаблона DOS, описываемые ниже) и нажмите клавишу Enter.

По желанию можно ввести текстовую строку для поиска, т.е. строку с текстом, которую надо найти в каком-либо файле. Если это поле остается пустым, команда Поиск файла выполнит только файловый поиск по заданному имени. Например, для нахождения всех файлов с расширением DOC, в которых содержится слово

«Commander», можно выполнить поиск файла *.doc, содержащий текст - “Commander”, который задается в поле Содержащий.

В окне поиска можно задавать некоторые дополнительные опции (рис.20):

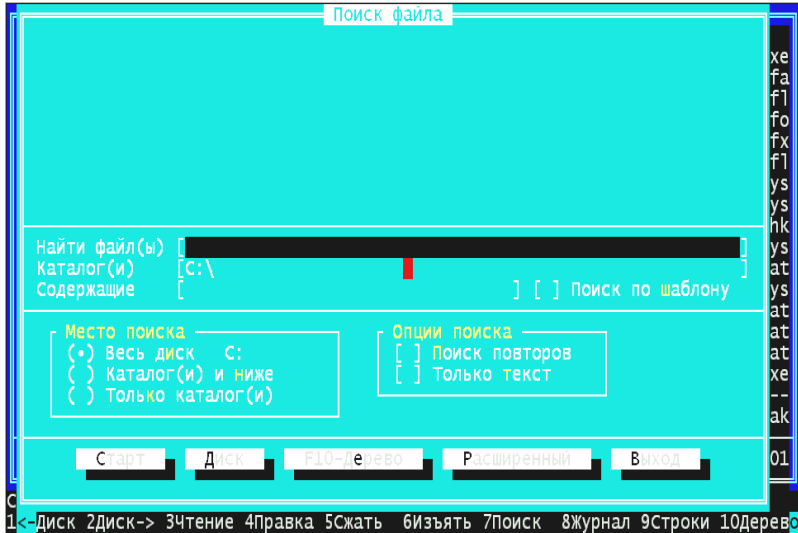


Рис.20. Вид панелей экрана для задания режима поиска файлов.

1. **Старт** - начало поиска.
2. **Диск** - задание диска для поиска.
3. **Дерево** - отображение дерева каталогов текущего диска.
4. **Расширенный** - вызов окна диалога “Улучшенный поиск”, где можно указать дальнейшие опции поиска.
5. **Выход** - выключение окна поиска. Или можно просто нажать клавишу Esc.

В процессе поиска можно остановить его кнопкой **Стоп**. После выполнения или остановки поиска на экране показываются его результаты. Окно результатов имеет дополнительные возможности (рис.21):

1. **Новый** - задание критериев для нового поиска.
2. **Просмотр** - просмотр выделенного файла, на котором стоит курсор.

3. **Перейти к** - переход в панель файлов NC, показывающую текущий каталог, с выбранным в окне результатов файлом.
4. **Панель** - пересылка списка файлов в панель “Поиска файлов”.
5. **Выход** - выключение окна поиска. Или можно просто нажать клавишу Esc.



Рис.21. Окно результатов поиска файлов.

Как уже говорилось, для поиска файлов можно применять шаблоны DOS, которые позволяют использовать следующие символы для поиска определенных типов файлов:

- * - Соответствует набору любых символов.
- *.* - Все файлы.
- *.bat - Все файлы с расширением .bat.
- a*.bat - Все файлы .bat, начинающиеся с “a”.
- ? - Соответствует одному произвольному символу в задаваемом имени файла.
- [] - Соответствует любому из символов или из диапазона символов, заданных между скобками. Например, [abcd]*.bat или [a-d]*.bat - находит файлы .bat, начинающиеся с a,b,c,d.
- ~ - соответствует заданному, за исключением файлов, следующих за символом “~”.

Журнал команд - Alt + F8

Эта команда вызывается комбинацией Alt + F8 и отображает список последних 16 команд DOS, выполненных во время текущего сеанса работы с NC. В простейшем случае выделите команду из списка на экране, поставив на нее курсор, и нажмите клавишу Enter для повторного выполнения этой команды.

Существует и краткая форма вызова **Журнала**: для вызова недавней команды начните набирать ее и нажмите клавиши Ctrl + Enter. Эта краткая форма вызывает самую последнюю команду, которая начинается с набранных букв. Правка списка команд **Журнала** возможна с помощью клавиши F4, вызывающей внутренний редактор NC. Окно **Журнала** команд показано на рис.22.

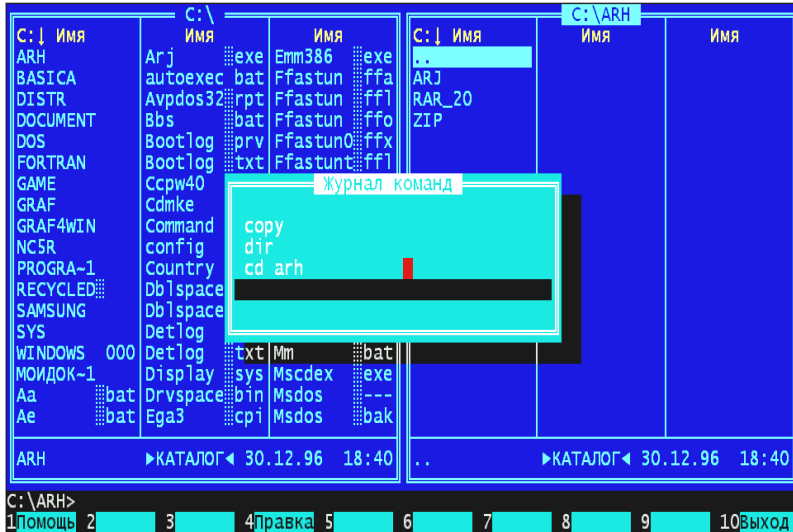


Рис.22. Окно **Журнала** команд оболочки NC.

Режим работы экрана - Alt + F9

Эта команда переключает режим отображения экрана с 25 - на 43 -строчковый (на EGA мониторах) или с 25 - на 50 - строчковый (на VGA мониторах) и обратно. Пятидесяти (или 43) строчный режим позволяет сразу выводить на панели больше информации о файлах и директориях, что в некоторых случаях бывает очень удобно, однако шрифт на экране становится сравнительно мелкими, что может за-

труднять работу с NC.

Дерево каталогов - Alt + F10

Эта команда вызывается нажатием комбинации клавиш Alt + F10 и отображает дерево каталогов, где можно быстро переключаться с каталога на каталог, создавать, удалять и переименовывать их. Управление курсором происходит клавишами со стрелками, а файловые операции с помощью функциональных клавиш, функции которых показаны в строке подсказки внизу экрана.

Дерево каталогов хранится на каждом диске в файле treeinfo.ncd, а возможный его вид показан на рис.23. Этот файл обновляется при создании, переименовании и удалении каталога с использованием возможностей Norton Commander.

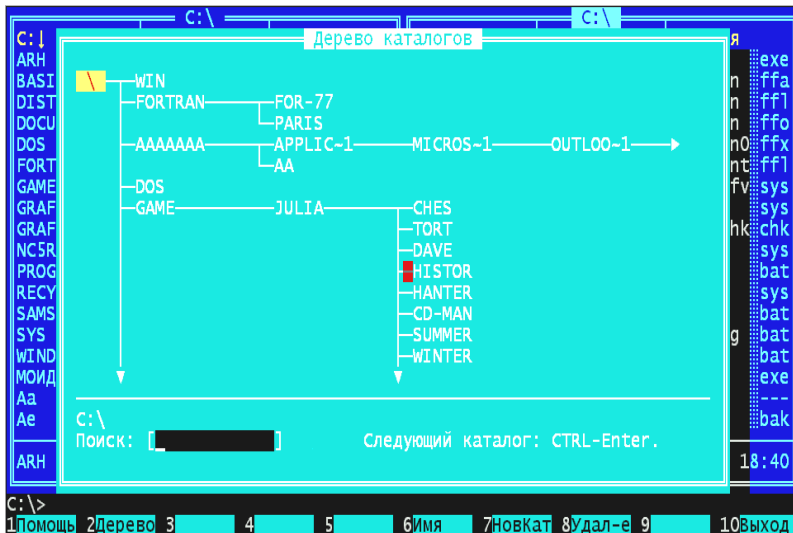


Рис.23. Вид экрана с Деревом каталогов.

Обработка расширений

Эта команда используется для создания файла расширений, который сообщает NC, какое приложение - программу нужно запускать вместе с файлом, на котором нажата клавиша Enter. Для создания файла расширений выберите пункт **Обработка расширений** в меню **Команды**, которое появляется при нажатии клавиши F9.

Если такого файла нет, появится пустое меню, в которое можно вставлять нужные команды. Нажмите F6 или Ins для вызова окна Из-

менение обработки расширений и введите расширение и выполняемую команду. Для правки уже созданных расширений нажмите F4 и в появившемся окне **Изменение обработки** расширений редактируйте существующие расширения и команды. Клавиша F8 удаляет выделенную в данный момент команду, а клавиша F2 сохраняет сделанные изменения (рис.24).

В качестве примера приведем строку записи расширений в том виде, как она хранится в файле nc.ext:

gif: viwpic !!

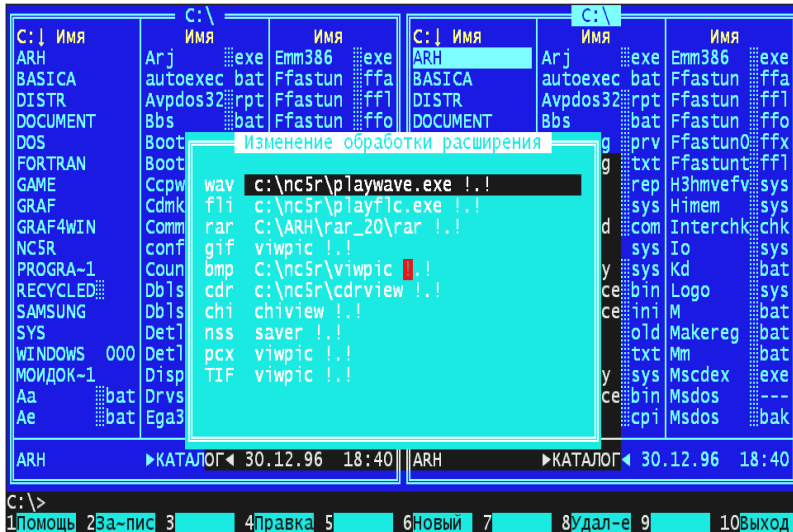


Рис.24. Окно обработки расширений.

Здесь первым стоит расширение графического файла gif, для которого необходимо использовать программу просмотра графических файлов viwpic.exe, а символы "!!" показывают, что при запуске этой программы должен загружаться файл, на котором стоит курсор и была нажата клавиша Enter.

Редактор расширений

Этот режим можно вызвать с помощью клавиши F9 и пункта **Команды Главного меню Редактор расширений**. Она используется для задания редактора, применяемого для работы с файлами с различными расширениями. В режиме редактирования изменение редакторов возможно клавишей F4. Для создания файла редакторов по

расширению выполните указанные выше действия и если такого файла нет, то на экране отображается пустое окно. Теперь можно **Вставить**, **Редактировать** или **Удалять** редакторы клавишами F6, F4 и F8 так же, как в предыдущем параграфе. Информация о редактировании файлов с заданными расширениями содержится в файле ncredit.ext, который входит в состав NC. В качестве примера приведем строку:

Doc: word !.!

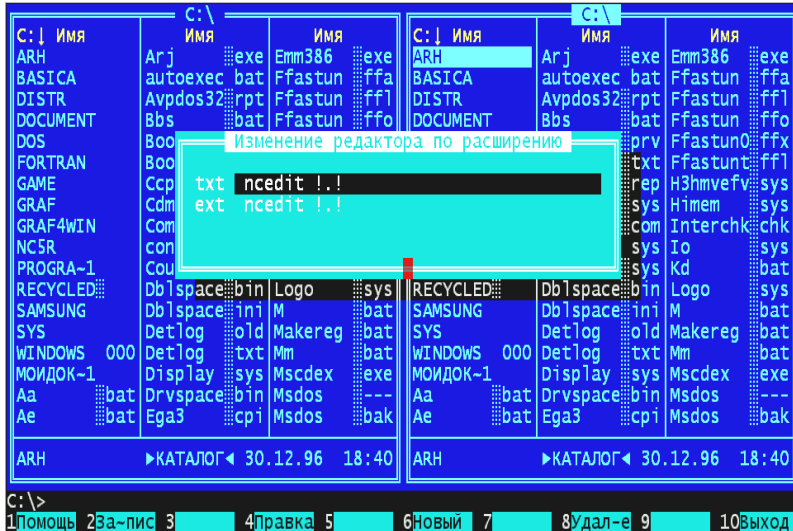


Рис.25. Окно редактора расширений.

Здесь первым стоит расширение файла, для которого необходимо использовать редактор Word (для DOS), а символы “!.!” показывают, что при запуске редактора должен загружаться файл, на котором находился курсор в момент нажатия клавиши Enter.

Быстрый поиск файла

Для быстрого поиска файла на панели NC нажмите Alt и начинайте набирать имя искомого файла. Внизу на панели файлов появится окно диалога быстрого поиска. Первый файл, удовлетворяющий указанной спецификации, выделяется подсветкой, т.е. на него перейдет экранный курсор. Для выхода из режима быстрого поиска нажмите Esc или Ctrl + Enter для продолжения поиска.

Панель инфо для жесткого диска - Ctrl + L

Для отображения на неактивной панели NC информации о постоянной и оперативной памяти компьютера нажмите Ctrl + L. Будет показано общее и свободное пространство в оперативной памяти, используемой NC, а также полное и свободное место на жестком диске компьютера.

Повторное нажатие этих клавиш выключает на экране компьютера информационную панель (рис.26).

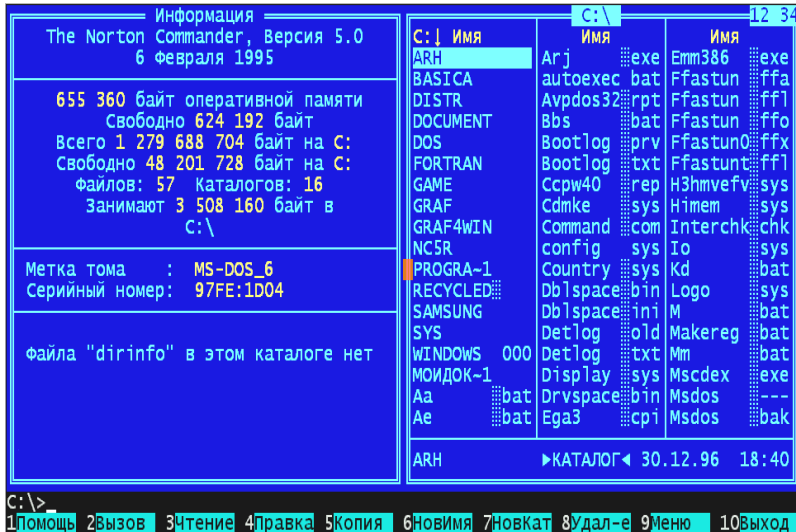


Рис.26. Панель Информации о состоянии жесткого диска.

Информация о файле или каталоге - Ctrl + Q или Ctrl + Z

Для получения краткой информации о файле, на котором стоит курсор, необходимо нажать Ctrl + Q. Эта комбинация позволяет выводить на другую панель NC информацию о размере и дате создания файла или каталога. Если в этом режиме нажать F3, то можно просмотреть содержимое этого файла.

Повторное нажатие этой комбинации выключает режим информации. Несколько иную, но аналогичную информацию дает комбинация клавиш Ctrl + Z (рис.27).

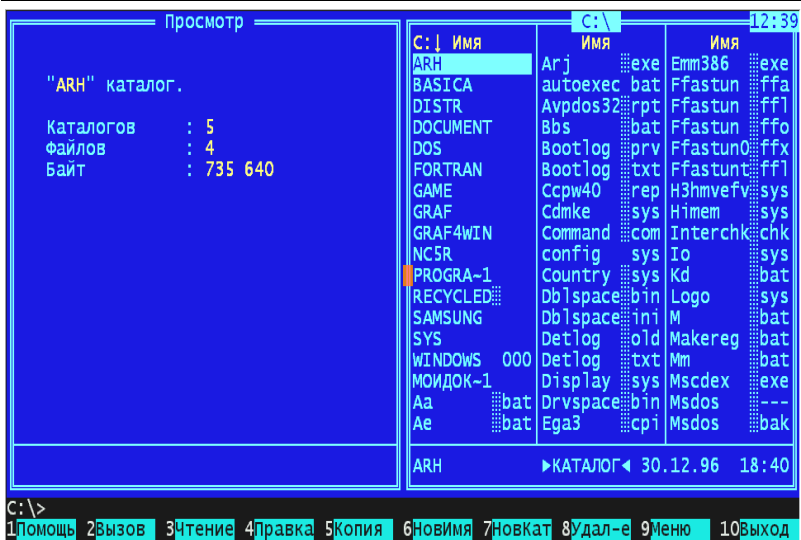


Рис.27. Информация о файле или каталоге.

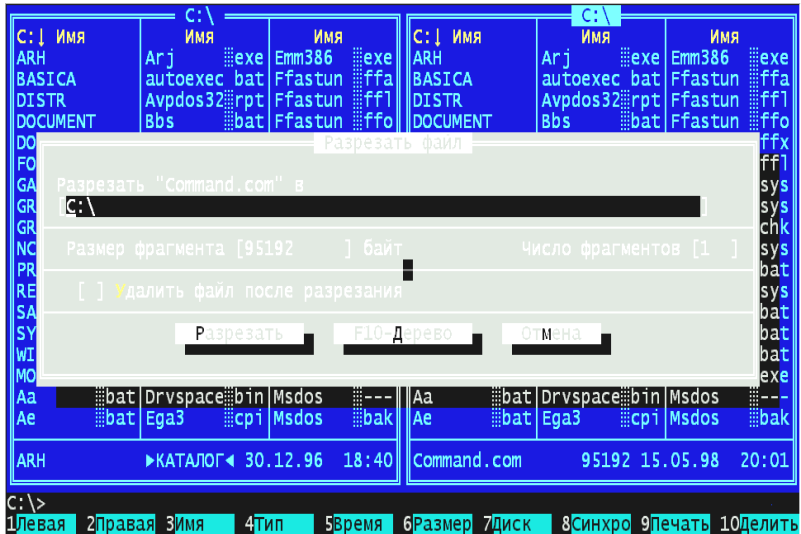


Рис.28. Вид экрана при резке и объединении файлов.

Разрезать и слить файл - Ctrl + F10

Команда Ctrl + F10 позволяет разбивать файл на сегменты или собрать из сегментов целый файл. Для расщепления файлов нужно задать размер сегментов, и при нажатии Enter число сегментов вычисляется автоматически. Можно, наоборот, задавать число сегментов, а размер каждого вычисляется автоматически. Если установить флажок **Удалить файл** после разрезания, то исходный файл будет удален (рис.28).

Для объединения файлов выделите те их них, которые надо объединить и нажмите Ctrl + F10. Укажите путь и имя нового файла, по умолчанию принимается имя исходного файла, и нажмите Enter для объединения.

Если установить флажок **Удалить сегменты** после слияния, то все части файла будут удалены.

Установка атрибутов файла

Эта команда позволяет устанавливать или изменять **Атрибуты** файла и доступна по клавише F9 в меню **Файл**. Вид окна установки атрибутов показан на рис.29. Возможны следующие атрибуты файла:

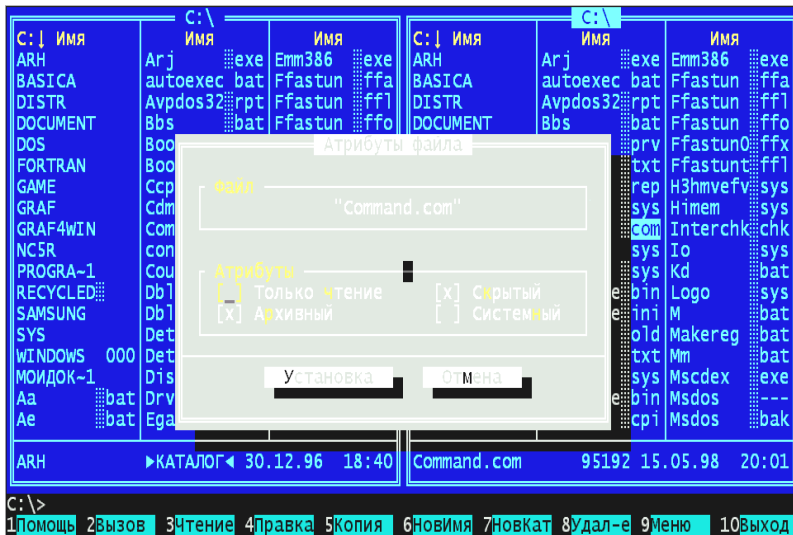


Рис.29. Вид экрана при установке **Атрибутов** файла.

1. **Только чтение** - файлы можно только читать, но нельзя редактировать и удаляются они клавишей **Удалить** только после дополнительного подтверждения.
2. **Архивный** - устанавливается каждый раз когда файл изменяется. Он обычно снимается после создания резервной копии файла.
3. **Скрытый** - скрытые файлы не видны на экране компьютера, если выключен режим показа скрытых файлов или при использовании DOS команды DIR.
4. **Системный** - системные файлы резервируются для некоторых файлов DOS.

При установке атрибутов возможен выбор следующих опций:

1. **Включая подкаталоги** - для задания изменения атрибутов всех файлов в подкаталоге.
2. **Использовать фильтр** - для выбора файлов по спецификации фильтра, которая также устанавливается в этом окне.

НЕКОТОРЫЕ УТИЛИТЫ MS

Утилита копирования дискет

Утилита доступна в подменю **Диск Главного меню**, а вид окна MS показан на рис.30. Эта команда создает идентичную копию дискеты и работает значительно проще и быстрее, чем аналогичная команда DOS. Можно даже создать файл - образ этой дискеты и сохранить его на жестком диске. Этот образ может быть использован для создания любого количества копий такой дискеты.



Рис.30. Вид экрана при копировании дискет.

При копировании дискет доступны следующие опции - параметры копирования:

1. **Пропускать плохие сектора** - для пропуска плохих секторов дискеты при копировании. Этот режим позволяет сохранить с плохой дискеты всю возможную информацию перед попыткой исправления плохих секторов на ней какой-либо программой. Опция настройки режима копирования - Пропускать плохие сектора - никогда не сохраняется при выходе из режима копирования и при запуске утилиты, если нужно, ее необходимо задавать снова. При запуске программы она всегда выключена.

2. **Проверять при копировании** - включает проверку исходной дискеты, что повышает надежность копирования.

Утилита воспроизведения звука

В состав Norton Commander 5.0 входит утилита playwav.exe для проигрывания звуковых файлов с расширением WAV. Т.к. эта программа не использует дополнительного оборудования, кроме встроенного в компьютер динамика, качество звука будет не очень хорошим, то ее можно использовать для предварительного проигрывания звуков, чтобы определить, какой звук находится в данном файле.

Утилита форматирования дискет

Команда доступна по клавише F9 в **Главном меню Диск** и позволяет быстро и эффективно форматировать дискету (Рис. 31) в следующих трех режимах работы:

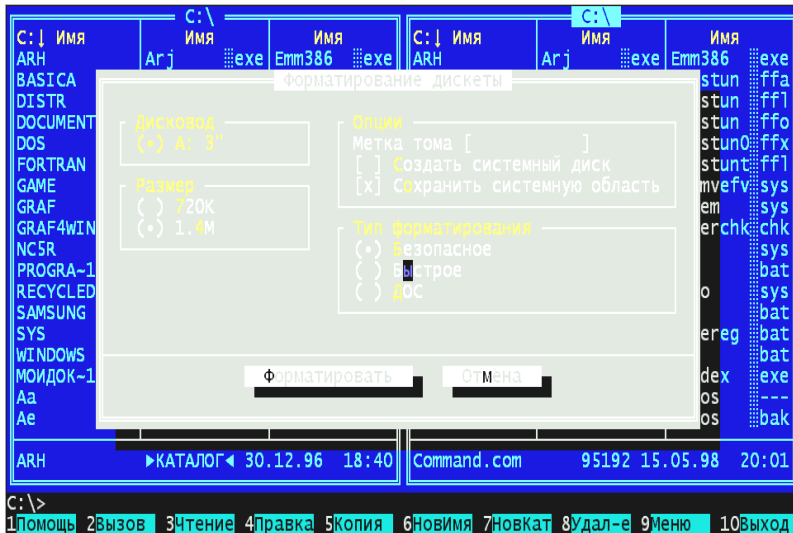


Рис.31. Окно форматирования дискет.

1. **Безопасный** - при переформатировании работает намного быстрее, чем программы форматирования DOS. Системная информация перезагружается так, что дискета кажется пустой, но при этом после форматирования все данные остаются в восстанавливаемом

виде при помощи программы Восстановления - Unformat из комплекта Norton Utilits. В случае первого форматирования дискеты этот режим работает так же, как в системе DOS.

2. **Быстрый** - занимает несколько секунд т.к. только перезагружает информацию на диске так, что он кажется пустым.

3. **DOS - формат** - DOS низкого уровня и предыдущая информация полностью стирается. Этот режим несколько быстрее, чем стандартная команда форматирования DOS.

4. При форматировании можно задавать размер диска, по умолчанию он принимается максимально возможным.

Возможны также следующие дополнительные опции:

1. **Метка тома** - можно вводить метку длиной до 11 символов.
2. **Сделать системный диск** - позволяет создать системный - загружаемый диск с переносом на него системных файлов.
3. **Сохранить системную область** - позволяет сделать копию системных файлов и сохранить их в файле IMAGE.DAT.

Утилита NC_EXIT

Эта полезная программа выгружает ядро Norton Commander, т.е. выключает его так же, как клавиша F10 без использования резидентных программ (TSR) и может быть использована для запуска дополнительных программ, например, Windows.

При выходе из Norton Commander возможно использование командной строки, например:

1. NC_EXIT - простой выход из NC без запуска - включения другой программы.
2. NC_EXIT <строка> - выход с запуском программы, указанной в последующей <строке>, которая может содержать до 11 символов, эквивалентных вводу с клавиатуры после выхода из Norton Commander.

Команду можно использовать в пакетных файлах или в наборе команд в пользовательском меню. Не требуется никаких специальных параметров и, если указана <строка>, то она обрабатывается, как ASCII последовательность символов. В силу внутренней структуры NC_EXIT должна выполняться последней в пакетных файлах или в наборе команд в пользовательском меню.

Как NC_EXIT, так и вызываемая <строкой> команда (программа или пакетный файл) должны находиться в каталогах, описанных в переменной PATH файла AUTUEXEC.BAT. В противном случае про-

грамма может не выполняться. В некоторых случаях имеет смысл поместить вызываемый пакетный файл в каталог Norton Commander, который должен быть описан в переменной PATH.

Для запуска Windows после выхода из Norton Commander, можно создать пункт пользовательского меню, вызываемого по F2:

```
W: Windows
nc_exit win
```

Предполагается, что каталог, содержащий WIN.COM, описан в переменной PATH. Таким образом, когда будет выбран этот пункт меню, то программа закроет Norton Commander и вызовет Windows из командной строки DOS. Надо отметить, что на правильно настроенных программах вызов Windows после выхода из NC по клавише F10 происходит автоматически.

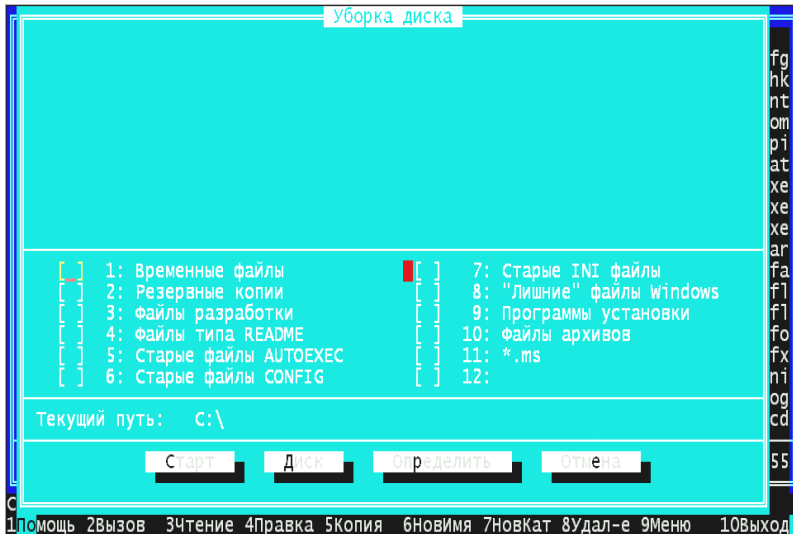


Рис.32. Окно Утилиты уборки диска.

Утилита уборки диска

Утилита доступна в Главном меню раздела Диск и позволяет очищать диски от ненужных файлов, например, ВАК, INI, TMP и т.д. При вызове утилиты нужно, передвигая курсор клавишами со стрелками, установить, нажимая клавишу пробел, "птички" около выбранных пунктов поиска. После выбора пункта Старт и нажатия клавиши Enter программа начинает поиск, выводя на экран его результаты.

Когда найдены все выбранные типы файлов, на экране показывается их список, передвигаться по которому можно с помощью клавиатуры управления курсором. Все файлы этого списка выделены желтой подсветкой, снять которую, в случае необходимости, можно, нажав клавишу Ins.

Когда определены файлы для стирания, следует выбрать пункт меню **Удалить** и нажать Enter, а после появления окна с предложением о подтверждении стирания, можно выбрать пункт Все и снова нажать Enter для стирания всех выбранных файлов. Окно подтверждения подобно такому же окну при обычном стирании объектов по F8, и возможности выбора файлов для стирания здесь полностью совпадают (Рис. 32).

Конфигурация NC

Вызов этого пункта меню по клавише F9 в разделе **Команды** выводит на экран дополнительное окно, показанное на рис.33, которое имеет следующие возможности для установок режимов работы NC:

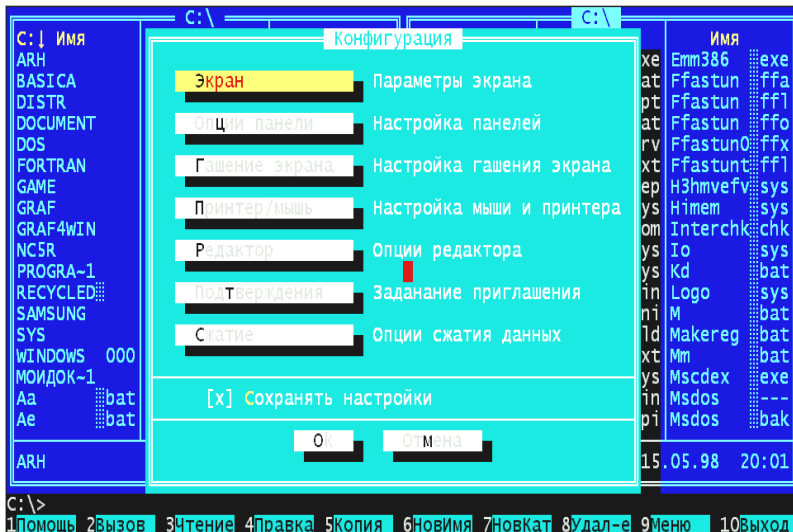


Рис.33. Общий вид окна меню **Конфигурация** по клавише F9.

1. **Экран** - выводит еще одно окно для дополнительных установок режимов работы экрана (рис.34):

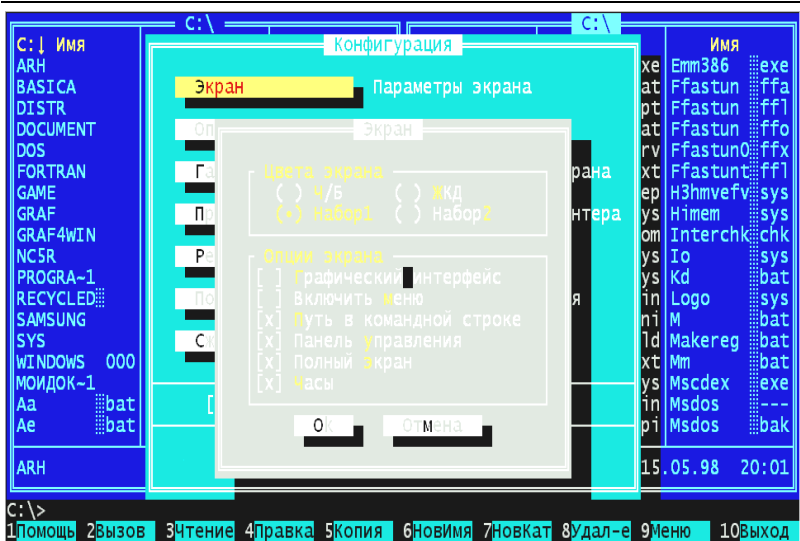


Рис.34. Вид окна Экран из меню Конфигурация.

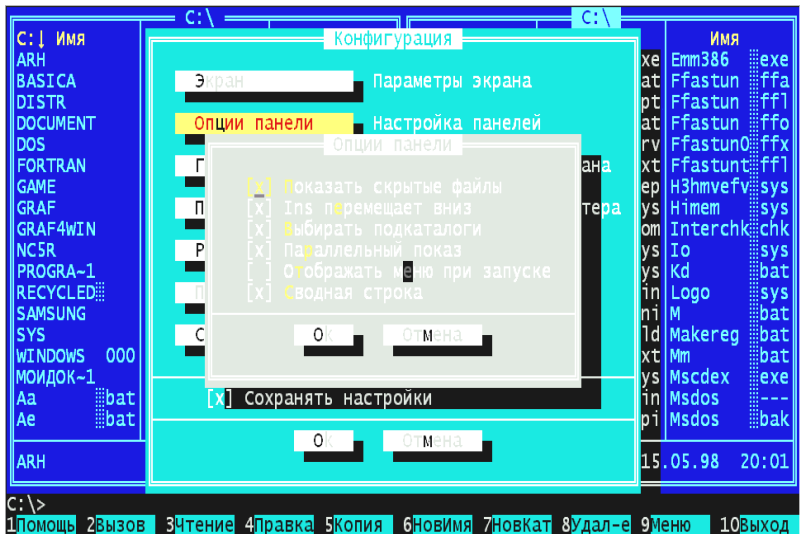


Рис.35. Вид окна Опции панели из меню Конфигурация.

- 1) Цвет экрана - можно выбрать нужную цветовую схему.

2) **Графический интерфейс** - позволяет отображать панели NC в графическом режиме.

3) **Включить меню** - для отображения сверху экрана линейки **Главного меню**.

4) **Путь в командной строке** - для показа полного пути DOS в командной строке NC.

5) **Панель управления** - для постоянного показа на экране панели управления.

6) **Полный экран** - для включения полного экрана NC, в противном случае NC занимает половину экрана.

7) **Часы** - для отображения текущего времени компьютера справа сверху экрана.

2. **Опции панели**, показанные выше на рис.35, - выводят на экран дополнительное окно, в котором имеются следующие возможности:

1) **Показывать скрытые файлы** - для показа на панелях NC скрытых файлов, которые отмечаются специальным значком между именем и расширением.

2) **Ins перемещает вниз** - для задания перемещения курсора при использовании клавиши Insert.

3) **Выбрать подкаталоги** - для указания, что нужно выделять каталоги при использовании шаблона “*.*”.

4) **Параллельный показ** - чтобы указать, что выделенный каталог автоматически выбирался, а его содержимое показывалось на другой панели.

5) **Отображать меню при запуске** - чтобы меню пользователя отображалось при запуске NC.

6) **Сводная строка** - чтобы внизу экрана отображалась сводная строка.

3. **Гашение экрана** - программы сохранения экрана (рис.36) выводят окно с множеством пунктов для выбора хранителя экрана. Нужно также указать, через какое время должен включаться хранитель экрана.

4. **Принтер/Мышь** - позволяет задавать следующие возможности:

1) **Порт принтера** - для выбора порта, обычно LPT1, при работе с принтером.

2) **Тип принтера** - для указания типа принтера.

3) **Мыши для левши** - для перемены местами кнопок мыши.

4) **Коррекция работы мыши** - для установки программной коррекции и ускорения работы последовательной мыши.



Рис.36. Вид окна **Гашение экрана** из меню **Конфигурация**.

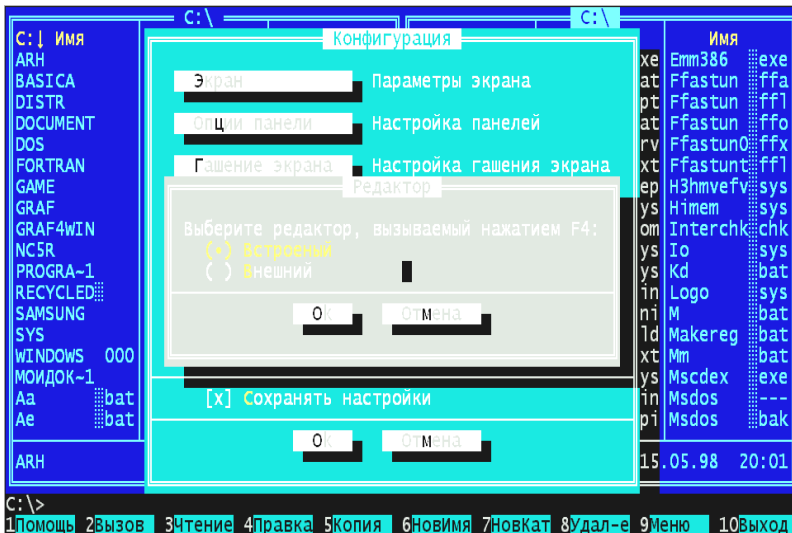


Рис.37. Окно выбора редактора.

5. **Редактор** - программа NC позволяет определять, какой редактор текста будет вызываться при нажатии на клавишу F4. Если выбрать Встроенный редактор, то будет использоваться редактор NC. При выборе Внешнего редактора необходимо задать его имя (рис.37).

6. **Подтверждение** - позволяет установить вывод на экран дополнительных подтверждений при выполнении различных операций, как показано на рис.38.

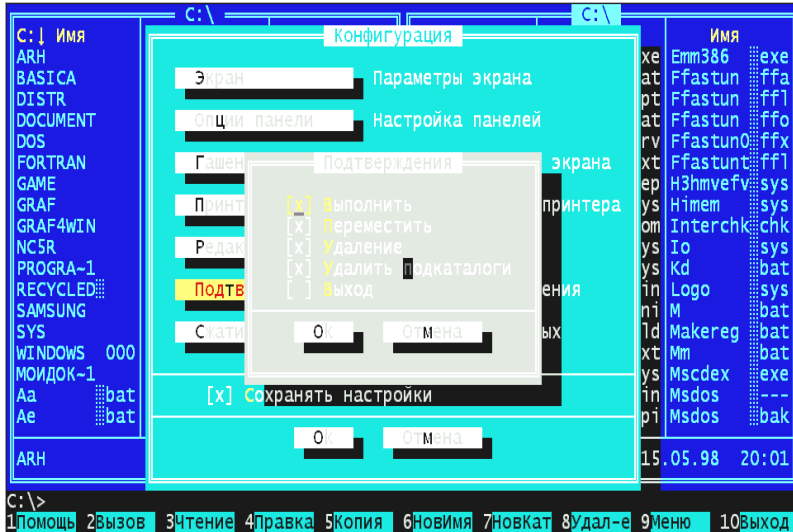


Рис.38. Вид окна **Подтверждение**.

7. **Сжатие** - позволяет выбрать метод сжатия - архивации файлов, а также выбрать нужна ли защита паролем сжатых файлов (рис.39). Можно выбрать несколько методов сжатия:

1) **Автоматический выбор лучшего метода** - позволяет NC самостоятельно определить метод сжатия.

2) **Минимизировать размер** - дает максимальное сжатие, но работает несколько медленнее.

Оптимизировать скорость - позволяет увеличить скорость сжатия при большем объеме.

Если внизу окна **Конфигурация** установить флажок **Сохранять настройки**, то при выходе из **Главного меню** изменения будут сохраняться.

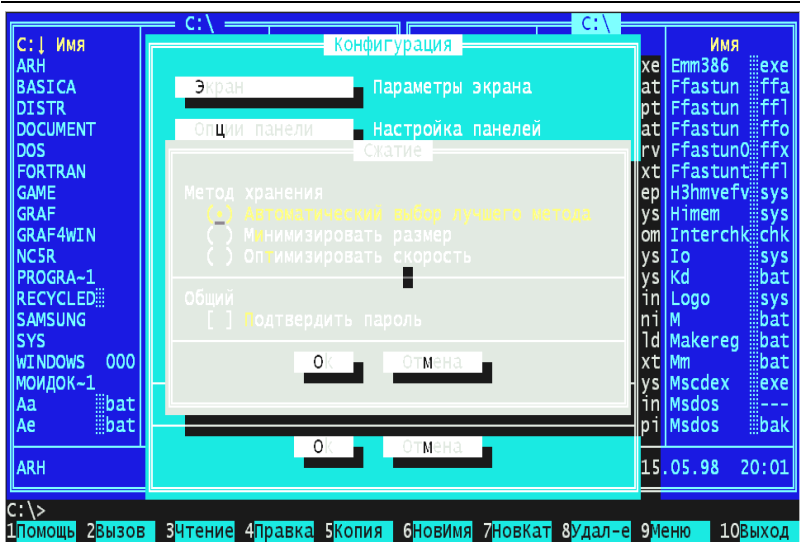


Рис.39. Метод сжатия, архивации файлов.

Режим эмуляции Терминала

Режим эмуляции терминала вызывается из **Главного меню Команды** и является программой работы с модемом при соединении двух компьютеров по телефонной сети для взаимной пересылки файлов между ними. В данной версии NC используется **Терминал - Telex - 3.0**, который несколько отличается от более ранней версии Term-95 из Norton Commander - 4. Новая программа **Терминала** полностью русифицирована и позволяет выполнять:

1. Набор телефонный номер и установить связь с удаленным компьютером по телефонной линии. Работает режим автодозвона с заданным числом попыток дозвониться при занятой линии.
2. Эмулировать различные кодировки - TTY, ANSI, Hex и другие после установки соединения.
3. Посылать и принимать файлы с использованием протоколов Xmodem, Zmodem, ASCII, Kermit и другие.
4. Интерпретировать полученные или переданные данные с помощью кодовых таблиц ASCII, Windows и КОИ-8.

Экран **Терминала** состоит из трех частей: области ввода - вывода данных, главного меню и индикатора состояния. В области ввода - вывода можно набирать символы для отправки абоненту и видеть символы, приходящие с модема от выбранного абонента (рис.40).

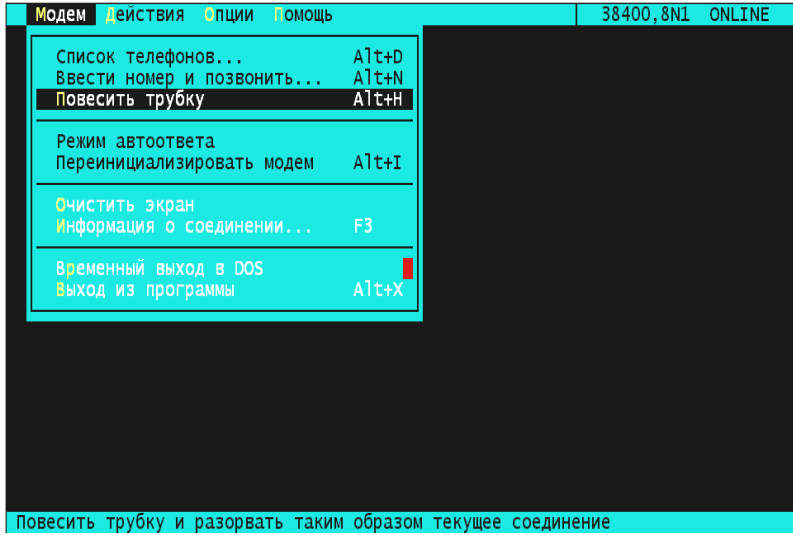


Рис.40. Окно Терминала.

Главное меню Терминала является панелью управления программой, причем для работы с ним можно использовать клавиатуру или мышь. При использовании клавиатуры для его вызова надо нажать F9 или Alt, а затем выбрать из меню нужный пункт и нажать Enter. **Главное меню** содержит несколько пунктов - **Модем**, **Действия**, **Опции** и **Помощь**, которые раскрываются в систему ниспадающих меню.

ПРОГРАММИРОВАНИЕ НА TURBO BASIC

Третий раздел книги содержит сравнительно полное описание алгоритмических языков программирования Quick Basic и Turbo Basic (в дальнейшем просто Basic), работающих на основе стандартной операционной системы MS - DOS. Эти программы могут так же работать и на основе ОС Windows. Описываемые здесь версии языка Basic представляют собой наиболее мощную реализацию этого языка для персональных ЭВМ типа IBM PC. Они имеют намного более широкие возможности по сравнению со старыми версиями языка Basic A или Basic Q. Рассматриваемые версии по своим функциональным возможностям очень похожи на хорошо известный алгоритмический язык Fortran версии Дубна или 77.

ЭЛЕМЕНТЫ ЯЗЫКА

Строки программы на языке Basic в общем случае имеют следующий формат:

[метка] [LET] BASIC оператор [:BASIC оператор] ...

В одной строке программы может располагаться более чем один оператор, но при этом каждый оператор в строке отделяется от других операторов с помощью символа ":" - двоеточие. Строка с операторами языка может содержать не более 255 символов. Оператор присваивания LET вводить в начале строки не обязательно, т.е. для того, чтобы присвоить переменной значение выражения, достаточно знака равенства.

Метка строки или ее номер, также не обязательные элементы программы и могут использоваться только для задания номера метки строки в операторах типа GOTO, THEN, GOSUB и т.д.

Константы

Существует два типа констант - строковые и числовые. Строковая константа представляет собой последовательность до 255 символов, заключенных в кавычки.

Пример:

```
"HELLO"  
"-25,00.00"  
"A+B*200"
```

Числовые константы представляют собой положительные и отрицательные числа. Числовые константы в Basic не могут содержать запятой, а десятичная часть числа отделяется с помощью знака “.” - точка.

Существует 5 типов числовых констант:

1. Целые константы - представляют собой целые положительные и отрицательные числа в диапазоне от -32768 до +32767 и не могут содержать десятичную точку.

2. Константы с фиксированной точкой - положительные или отрицательные вещественные числа, т.е. числа с десятичной точкой.

3. Константы с плавающей точкой - числа, имеющие экспоненциальную форму. Эти константы состоят из целого или вещественного числа (мантиссы) и буквы E (или D - двойная точность), за которой следует целое число (экспонента). Если необходимо, мантисса и экспонента могут иметь знак. Константы с плавающей точкой имеют диапазон представления от $10E-38$ до $10E+38$.

4. Шестнадцатеричные константы - представляются шестнадцатеричным числом с префиксом &H.

5. Восьмеричные константы - представляются восьмеричным числом с префиксом & или &O.

Числовые константы могут быть обычной или двойной точности. Константы обычной точности представляются 7 цифрами, а константы двойной точности 16 цифрами.

Константой обычной точности является любая числовая константа, которая имеет один из следующих признаков:

1. 7 или меньше цифр.
2. Экспоненциальная форма с использованием E и мантиссой не более 7 знаков.
3. Заканчивается восклицательным знаком.

Константой двойной точности является любая числовая константа, у которой есть один из следующих признаков:

1. Восемь и более цифр, но не больше 16.
2. Экспоненциальная форма с использованием D и мантиссой не больше 16 цифр.
3. Заканчивается символом "#".

Примеры числовых констант:

1. 645 - целая константа обычной точности.

2. -1.09E-06 - константа обычной точности с плавающей точкой.
3. 22.5! - константа обычной точности с фиксированной точкой.
4. 1.09432D-05 - константа двойной точности с плавающей точкой.
5. 345692811 - целая константа двойной точности.
6. &H32F - шестнадцатеричная константа.
7. &O347 - восьмеричная константа.

Переменные

Переменные представляют собой символы (буквы или комбинации букв и цифр), используемые для обозначения величин, которые присутствуют в программе. Значение переменной может быть непосредственно присвоено программистом, либо оно может определяться в результате вычислений в программе. До того, как переменной будет присвоено значение, она принимается равной нулю.

Имена переменных в Basic могут быть любой длины, однако распознавание переменной осуществляется по первым 40 символам. Для обозначения имени переменной можно использовать буквы, цифры и десятичную точку, причем, первым символом имени должна быть буква. Кроме того, допустимо использовать специальные символы объявления типа, о которых будет сказано ниже. Имя переменной не может совпадать с зарезервированным именем. К зарезервированным именам в Basic относятся имена стандартных директив, операторов, функций и операций.

Переменные могут обозначать либо числовые величины, либо строки. Имя строковой или текстовой переменной должно заканчиваться символом \$. Символ \$ в данном случае является символом объявления, указывающим на то, что переменная является строковой - текстовой и состоит из символов текста. Имена числовых переменных могут обозначать целые числа, числа обычной и двойной точности.

Если необходимо, то для задания типа числовой переменной используются следующие символы объявления типа, всегда помещаемые в конце символа переменной:

1. Знак % - целая переменная, которая не содержит десятичную точку (знак % можно не набирать, достаточно отсутствия в числе десятичной точки).
2. Знак ! - переменная обычной точности содержит точку (знак ! можно не писать, достаточно присутствия десятичной точки).

3. Знак # - переменная двойной точности (знак # или описание переменной оператором DEFDBL обязательны).

4. Знак \$ - строковая переменная (знак \$ или описание переменной оператором DEFSTR обязательны). Например, ABC\$ = "A" - значение строковой переменной A заключается в скобки.

По умолчанию числовые переменные имеют представление с обычной точностью.

Примеры имен переменных:

1. A\$ - имя строковой переменной.
2. PI# - имя переменной двойной точности.
3. MINIMUM! - имя переменной обычной точности.
4. LIMIT% - имя целой переменной.
5. ABC - имя переменной обычной точности.

В Basic существует и другой способ объявления типа переменной. Для этого используются операторы DEFINT - целая, DEFSTR - строковая, DEFSNG - обычной точности и DEFDBL - двойной точности, которые будут описаны ниже. Для размещения в памяти целой переменной необходимо - 2 байта, для переменной обычной точности - 4 байта, а для переменной двойной точности - 8 байт. Для размещения строковой переменной в памяти необходимо 3 байта плюс количество байт, равное количеству символов в строке.

Массивы переменных

Массив - это группа чисел, ассоциированных с одним именем. Каждый элемент массива связан с переменной массива, которая имеет индекс, определяющийся целой величиной или целым выражением. Размерность массива определяется числом индексов переменной.

Например, A(10) - одномерный массив, каждый элемент которого A(I) имеет определенную величину, а индекс I может принимать значения от 0 до 10. B(20,30) - двумерный массив, который, если не использовать нулевые элементы массива можно представить в виде таблицы с 20 строками и 30 столбцами.

Преобразование типов

В Basic существует возможность преобразования типов числовых констант. Если числовая константа одного типа присваивается числовой переменной другого типа, то переменной будет присвоена величина, соответствующая типу переменной. Если строковой пере-

менной присваивается числовое значение или числовой переменной - строковое, то выдается сообщение об ошибке "Type mismatch".

Пример:

```
A% =23.42  
PRINT A%
```

RUN - запуск программы и вывод на экран результата.

23

Таким образом, величина A будет иметь целое значение. Во время вычисления выражения все команды в арифметических операциях или операциях отношения преобразуются к одному типу представления - типу наибольшей точности. Результат арифметической операции, также будет иметь этот тип представления. При выполнении логических операций значения операндов преобразуются до целых, и результату присваивается целая величина.

Значения операндов должны находиться в пределах от -32768 до +32767, в противном случае выдается сообщение об ошибке "Overflow".

Когда число с дробной частью преобразуется в целую величину, выполняется округление. Если переменной двойной точности присваивается величина обычной точности, то эта величина преобразуется в представление двойной точности. Абсолютная погрешность между прежним и новым значением преобразованной величины не превышает $1.0E-7$.

Пример:

```
A=2.04  
B# =A  
PRINT A;B#
```

RUN - запуск программы и вывод на экран результата.

2.04 2.039999961853027

Выражения и операции

Выражением может быть просто строковая или числовая константа или переменная. Им может быть комбинация констант, переменных, функций и операций.

Операции в Basic можно разделить на четыре категории:

1. Арифметические операции.
2. Операции отношения.
3. Логические операции.
4. Строковые операции.

Рассмотрим последовательно все эти типы операций:

Арифметические операции

Арифметические операции в Basic имеют порядок старшинства, приведенный ниже:

1. Возведение в степень - X^Y .
2. Отрицание - X .
3. Умножение и деление - $X*Y$; X/Y .
4. Целочисленное деление - $X \setminus Y$.
5. Модуль числа - $X \text{ MOD } Y$.
6. Сложение и вычитание - $X+Y$; $X-Y$.

Для изменения порядка выполнения операций используются круглые скобки. Операции внутри скобок выполняются в порядке старшинства первыми. При выполнении операции деления целых чисел и присвоении результату целого числа, он округляется до целых значений (они должны быть в пределах от -32768 до 32767), перед выполнением целочисленного деления частное усекается до целого (не округляется, а берется только целая часть числа).

Примеры целочисленного деления:

$$10 \setminus 4 = 2$$
$$23.68 \setminus 6.99 = 3$$

Результатом операции модуля числа, также является остаток целочисленного деления.

Примеры:

$$10 \text{ MOD } 4 = 2$$
$$23.68 \text{ MOD } 6.99 = 3$$

Если во время вычисления выражения в Quick Basic происходит деление на ноль, выдается сообщение об ошибке "Division by zero", результату деления присваивается максимально возможное число со знаком числителя и выполнение программы продолжается. Если ноль возводится в отрицательную степень, выдается сообщение об ошибке "Division by zero", результату присваивается максимально возможное

положительное число и выполнение программы продолжается. При возникновении переполнения выдается сообщение об ошибке "Overflow", результату присваивается максимально возможное число с соответствующим знаком и выполнение продолжается.

В Turbo Basic при таких операциях после выдачи соответствующего сообщения обычно происходит "зависание" программы и требуется перезагружать компьютер.

Операции отношения

Операции отношения используются для сравнения двух величин. Результатом сравнения является "истина" (-1) или "ложь" (0):

1. = - знак равно, определяющий равенство величин - $X=Y$.
2. <> - знак не равно - $X\neq Y$.
3. < - знак меньше - $X<Y$ - величина X меньше, чем Y.
4. > - знак больше - $X>Y$ - величина X больше, чем Y.
5. <= - меньше или равно - $X\leq Y$.
6. >= - больше или равно - $X\geq Y$.

Если арифметические операции и операции отношения объединены в одно выражение, то арифметические операции всегда выполняются первыми.

Например, выражение:

$$X \wedge Y < (T - 1) / Z$$

является истинным, если величина X в степени Y меньше чем величина T - 1 деленная на Z.

Логические операции

Результатом логической операции является либо "истина" (не ноль), либо "ложь" (ноль). В выражениях логические операции выполняются после арифметических операций и операций отношения.

При выполнении логических операций операнды преобразовываются в шестнадцати битовое целое число в пределах от -32768 до +32767. Если значение операнда не находится в этих пределах, то результат будет ошибочным.

Строковые операции

Строки могут быть соединены с помощью операции конкатенации (обозначается знаком "+").

Пример:

```
A$ = "FILE"  
B$ = "NAME"  
PRINT A$ + B$  
PRINT "NEW " + A$ + B$
```

RUN - запуск программы и вывод на экран результата.

```
FILENAME  
NEW FILENAME
```

Для сравнения строк используются обычные операции отношения: =, <, >, <=, >=.

Сравнение строк осуществляется путем последовательного сравнения (слева направо) кодов символов строк. Если все коды совпадают, то строки равны. Если коды разные, то та строка, чей код символа больше, считается большей. Если при сравнении строк обнаружен конец одной из строк, то это означает, что короткая строка меньше. Следует помнить, что пробелы в начале и в конце строк являются значащими и влияют на результаты сравнения.

Примеры:

```
"AA" < "AB"  
"FILENAME" = "FILENAME"  
"X&" > "X#"  
"AB" < "ABC"
```

Все строковые константы, используемые в операциях отношения, должны быть заключены в кавычки. Можно предварительно текстовым переменным присвоить их значение и операции сравнения проводить с самими переменными.

Примеры:

```
A$="AA"  
B$="AB"
```

Тогда при сравнении:

```
A$<B$
```

Если:

```
A$="AC"
```

B\$="AB"

Тогда при сравнении будет обратный результат:

A\$>B\$

Если:

A\$= "FILE"

B\$= "FILE"

Тогда при сравнении получим равенство величин:

A\$=B\$

ОСНОВНЫЕ ОПЕРАТОРЫ BASIC

Написание любой программы на языке Basic выполняется с помощью последовательности определенных операторов, выполняющих те или иные математические и логические действия. Программа языка состоит также из операций присваивания переменным, и массивам их значений. Приведем далее краткие описания основных операторов, которые расположены в алфавитном порядке и некоторые примеры их использования.

Оператор CALL

Оператор CALL используется для вызова подпрограмм на Basic, находящихся в том же файле.

Синтаксис:

```
CALL <имя>[(<список аргументов>)]
```

где:

1. Параметр <имя> - задает имя подпрограммы.
2. Параметр <список аргументов> - представляет собой имена переменных или констант, разделенных запятыми, которые передаются в вызываемую подпрограмму.

Пример:

```
G=5: H=1: CALL ABC(G,H,E)
D=E: PRINT D
END
SUB ABC(R,T,Y)
Y=R+T
END SUB
```

Если необходимо получить результаты работы подпрограммы в основную программу, следует включить в <список аргументов> имена переменных, которым в подпрограмме будут присвоены возвращаемые значения.

Оператор CHAIN

Оператор CHAIN вызывает программу или подпрограмму из другого файла и передает ей переменные из текущей программы.

Синтаксис:

CHAIN <имя файла>

где параметр <имя файла> - задает имя файла для вызываемой подпрограммы, заключенное в кавычки.

Имя файла есть допустимое для MS - DOS алфавитно - цифровое имя с расширением. Если расширение опущено - предполагается программы с расширением .BAS, которая должна быть написана на Basic в компактном или текстовом формате. Если задано расширение .EXE - программа должна быть скомпилирована Basic в исполняемый файл. В Turbo Basic допустим вызов подпрограмм, находящихся в другом файле только из исполняемых файлов.

Пример:

```
CHAIN "PROG1"
```

Вызывающая программа обращается к файлу PROG1, в котором должна находиться подпрограмма, написанная на Basic. Эта подпрограмма проводит необходимые, запрограммированные в ней вычисления и передает их результат в основную программу.

Оператор CLOSE

Оператор CLOSE предназначен для завершения действий по вводу/выводу при работе с файлами на диске и закрывает файлы с указанными номерами.

Синтаксис:

```
CLOSE [[#] <номер файла> [, [#] <номер файла> ...]]
```

где параметр <номер файла> - представляет собой номер, под которым файл был открыт с помощью оператора OPEN.

Оператор CLOSE без параметров закрывает все файлы. Связь между файлом и его номером прерывается после выполнения оператора CLOSE. Это значит, что под тем же номером с помощью оператора OPEN может быть открыт другой файл. Оператор END всегда автоматически закрывает все файлы на диске. Оператор STOP прерывает работу программы, но файлы не закрывает.

Операторы COMMON и SHARED

Оператор COMMON передает переменные программе, вызываемой оператором CHAIN. Оператор SHARED передает переменные

программе, вызываемой оператором CALL. Переменным присписывается статус "общие".

Синтаксис:

```
COMMON [/имя-группы] <переменные> [(размерность)] [AS  
тип][,...]
```

или

```
SHARED [/имя-группы] <переменные> [(размерность)] [AS  
тип][,...]
```

где:

1. Параметр "имя-группы" - определяет группу переменных (длина имени до 40 байтов).
2. Конструкция "AS тип" - обязывает, чтобы переменная была одного из следующих типов: INTEGER, LONG, SINGLE, DOUBLE, STRING или типа, определенного пользователем.
3. Список переменных задается через запятую.

Операторы могут находиться в любом месте программы, хотя рекомендуется размещать их в начале. Не допускается использование одной и той же переменной в нескольких операторах COMMON.

SHARED означает, что переменные используются совместно во всех процедурах - подпрограммах программного модуля, находящегося в одном файле. SHARED может использоваться только внутри процедур - подпрограмм FUNCTION (только для Quick Basic) и SUB. Обозначения переменных, стоящих в операторе должны совпадать с использованными с основной программе.

Массивы переменных обозначаются добавлением символа () - скобки к имени переменной.

Пример:

```
COMMON A, B, C, D( ), GS
```

Оператор DATA

Оператор DATA заносит в память числовые и строковые константы, которые считываются оператором READ.

Синтаксис:

```
DATA <список констант>
```

где параметр <список констант> - может содержать числовые константы любого типа: с фиксированной точкой, плавающей точкой или целые.

Строковые константы в операторе DATA должны заключаться в кавычки только в том случае, если они содержат запятые, двоеточия или пробелы в начале или в конце строки. В других случаях кавычки необязательны. Тип переменной (числовой или строковой), задаваемой в операторе READ, должен согласовываться с соответствующей константой в операторе DATA.

Оператор DATA может располагаться в любом месте программы. Количество констант в операторе DATA ограничивается только длиной строки (256 символов). Константы должны разделяться запятыми. В программе может быть любое количество операторов DATA. Оператор READ считывает данные последовательно в соответствии с номерами строк операторов DATA. Таким образом, константы, задаваемые несколькими операторами DATA, можно рассматривать, как последовательный список данных, формируемый в зависимости от длины <списка констант> операторов DATA и от того, какие номера имеют строки этих операторов.

Константы, задаваемые операторами DATA, могут быть прочитаны вновь с начала оператором READ после применения оператора RESTORE.

Оператор DEF FN

Оператор DEF FN определяет некоторую функцию, описываемую пользователем.

Синтаксис:

DEF FN<имя> [(<список параметров>)] = <описание функции>

где:

1. Тип любой из перечисленных переменных может быть INTEGER, LONG, DOUBLE или STRING.

2. Параметр <имя> - должен быть допустимым в Basic именем переменной. Это имя с предшествующими буквами FN и будет именем функции.

3. Параметр <список параметров> - включает имена переменных для описания функции, которые при вызове функции будут заменены соответствующими значениями. Параметры в списке должны разделяться запятыми.

4. Параметр <описание функции> - представляет собой выражение, описывающее функцию.

Имена переменных, которые включаются в выражение, используются только для описания функции и не влияют на переменные в программе, имеющие такие же имена. Переменные, используемые в описании функции, могут либо указываться, либо не указываться в списке параметров. Если переменная включена в список, то ее значение приводится во время вызова функции. Если же переменной в списке параметров нет, то используется текущее значение переменной. Функция, описываемая пользователем, может быть числовой или строковой. Если тип указан в имени функции, то значение выражения, описывающего функцию, преобразуется к этому типу. Если тип, указанный в имени функции, и тип аргумента не соответствуют, то выдается сообщение об ошибке "Type mismatch".

Оператор DEF FN должен располагаться прежде, чем произойдет вызов функции, определяемой им. Если функция вызывается раньше, то выдается сообщение об ошибке "Undefined user function".

Пример:

```
DEF FNAB(X,Y)=X^3/Y^2
```

```
.
```

```
.
```

```
T = FNAB(I,J)
```

Операторы DEFINT/LNG/SNG/DBL/STR

Операторы DEFINT/LNG/SNG/DBL/STR объявляют тип переменных, как целый, длинный, обычной точности, двойной точности или строковый.

Синтаксис:

```
DEF<тип> <диапазон(ы) букв>
```

Здесь <тип> - это INT, LNG, SNG, DBL или STR.

Оператор DEF<тип> объявляет, что имена переменных, начинающихся с определенных букв, соответствуют указанному типу переменных. Если тип переменных не объявлен, то все переменные имеют обычную точность представления.

Примеры:

1. DEFDBL L-P - все переменные, начинающиеся с букв L, M, N, O и P, будут иметь двойную точность представления.

2. DEFSTR A - все переменные, начинающиеся с буквы A, будут строковыми переменными.

3. DEFINT I-N, W-Z - все переменные, начинающиеся с букв I, J, K, L, M, N, W, X, Y, Z, будут целыми переменными.

Оператор DO ... LOOP

Операторы DO ... LOOP организуют циклическое выполнение группы операторов ограниченных телом цикла DO ... LOOP пока (WHILE) или до тех пор, пока (UNTIL) условие продолжение цикла истинно.

Синтаксис:

```
DO
....
<операторы>
....
LOOP {WHILE | UNTIL} <условие продолжения цикла>
```

или

```
DO {WHILE | UNTIL} <условие продолжения цикла>
.....
<операторы>
.....
LOOP
```

где <условие продолжения цикла> - организуется логическим оператором, а <операторы> - список выполняемых операторов программы.

Пример:

```
DO
X=X+1
PRINT X;
LOOP WHILE X<10

END
```

RUN - запуск программы и выдача на экран.

1 2 3 4 5 6 7 8 9 10

Операторы DIM и REDIM

Оператор DIM определяет максимальное значение индексов переменных массива и отводит необходимую массиву память.

Синтаксис:

```
[RE]DIM <список переменных с индексами [AS тип]>
```

Для каждой переменной списка в скобках указывается верхняя граница индекса. Сами переменные записываются через запятую. Тип массива может быть любым допустимым в Basic.

Пример для одномерных массивов:

```
DIM A(20), B(50), C(100)
```

Пример для двумерных массивов:

```
DIM A(20,10), B(20,50), C(5,100)
```

Если переменная массива не была описана оператором DIM, то максимальное значение индексов принимается равным 10. Если значение индекса превышает максимальное установленное значение, то выдается сообщение "Subscript out of range". Минимальное значение индекса всегда равняется нулю, кроме того случая, когда оно изменяется с помощью оператора OPTION BASE.

Оператор DIM устанавливает все элементы описываемого числового массива равными нулю. Приставка RE переопределяет ранее определенный массив.

Оператор END

Оператор END завершает выполнение программы, закрывает все файлы и возвращает пользователя на командный уровень компилятора языка Basic. Также оператор END определяет окончание различных синтаксических конструкций, таких как DEF, FUNCTION, IF, SELECT, SUB или TYPE.

Синтаксис:

```
END [{DEF | FUNCTION | IF | SELECT | SUB | TYPE}]
```

Операторы END могут располагаться в любом месте программы с целью завершения выполнения программы. Оператор END в конце программы является необязательным.

Basic всегда возвращается на командный уровень после выполнения оператора END.

Пример:

```
IF K>1000 THEN END ELSE GOTO 20
```

Оператор ERASE

Оператор ERASE предназначен для удаления из программы массивов или их очистки. Если массив имеет атрибут STATIC, то все его элементы обнуляются. Массивы с атрибутом DYNAMIC уничтожаются и освобождают занимаемую ими память (только для Quick Basic). Синтаксис:

```
ERASE <список переменных массивов>
```

После удаления массивов с помощью оператора ERASE имеется возможность заново задать их размерность или использовать отведенную ранее под массив память для других целей. В случае, если производится попытка заново задать размерность массива, который не был удален оператором ERASE, выдается сообщение об ошибке: "Redimensioned array".

Пример:

```
50 ERASE A, B  
60 DIM B(99)
```

Оператор EXIT

Оператор осуществляет выход из программных блоков, определенных в конструкциях DEF, циклах DO...LOOP и FOR...NEXT, а также из определенных пользователем функций FUNCTION и процедур SUB до их логического завершения. Синтаксис:

```
EXIT {DEF | DO | FOR | FUNCTION | SUB}
```

В случае циклов управление передается на следующий оператор непосредственно после цикла. В случае процедур и функций управление передается на следующий оператор после точки вызова.

Операторы FOR и NEXT

Операторы FOR и NEXT предназначены для организации циклического выполнения последовательности некоторых операторов языка Basic заданное число раз.

Синтаксис:

```
FOR <переменная> = x TO y [STEP z]  
.....  
<операторы>  
.....  
NEXT
```

где <переменная> - используется в качестве счетчика. Первое числовое выражение (x) задает начальное значение счетчика. Второе числовое выражение (y) определяет конечное значение счетчика.

Строки программы, следующие за оператором FOR, будут выполняться до тех пор, пока не будет встречен оператор NEXT. После этого к значению счетчика будет прибавлена величина, определяемая выражением (z) опции STEP - шаг. Затем осуществляется проверка с целью определения того, не превысило ли значение счетчика его конечного значения (y). Если этого не случилось, компилятор Basic осуществляет переход назад на оператор, который располагается сразу за оператором FOR и процесс вычисления повторяется. В случае, если значение счетчика превысило конечное значение, выполнение программы продолжается с оператора, следующего непосредственно за оператором NEXT. В случае, если опция STEP не задана, приращение принимается равным единице. Если в опции STEP задано отрицательное значение, то конечное значение счетчика цикла должно быть меньше начального. При этом значение счетчика каждый раз будет уменьшаться, и цикл будет выполняться до тех пор, пока значение счетчика не станет меньше конечного значения. В случае, если начальное значение счетчика, умноженное на знак приращения, превышает его конечное значение, умноженное на знак приращения, цикл игнорируется.

Циклы FOR ... NEXT могут быть вложенными, т.е. один цикл FOR ... NEXT может быть помещен в контексте другого цикла FOR ... NEXT. При вложении циклов каждый цикл должен иметь собственное, отличное от других имя счетчика цикла. Оператор NEXT вложенного цикла должен располагаться в программе до оператора NEXT, соответствующего внешнему циклу. В случае, если вложенные циклы имеют одну и ту же точку окончания, для них можно задавать один оператор NEXT.

Переменные в операторе NEXT могут быть опущены. В этом случае оператор NEXT будет соответствовать последнему оператору FOR. Если оператор NEXT располагается в программе раньше соответствующего ему оператора FOR, то выдается сообщение об ошибке "NEXT without FOR", и выполнение программы прекращается.

Пример:

```
10 J=5
20 FOR I=1 TO J+5
30 PRINT I;
40 NEXT
```

RUN - запуск программы и выдача на экран результата.

```
1 2 3 4 5 6 7 8 9 10
```

В примере цикл выполняется десять раз и на печать выводится значение счетчика цикла.

Оператор FUNCTION

Этот оператор в Quick Basic (в Turbo Basic такого оператора нет) объявляет функцию, определенную пользователем.

Синтаксис вызова функции:

```
....
A = <имя >[( <список переменных >)]
```

....
Синтаксис функции:

```
FUNCTION <имя>[( <переменная 1>[()] [AS <тип>] [,...])]
[STATIC]
```

```
....
<имя>=<выражение>
```

```
....
END FUNCTION
```

где:

1. Параметр <имя> - любое допустимое для Basic имя, которое будет именем функции.

2. Параметр STATIC - позволяет сохранить значения локальных переменных функции между вызовами. Количество переменных - аргументов неограниченно и могут быть использованы любые допустимые в языке типы данных.

В теле функции обязательно должен быть оператор присваивания значения имени функции, которое передается в вызывающую программу.

Операторы GOSUB и RETURN

Операторы GOSUB и RETURN предназначены для перехода на подпрограмму и возвращения из нее:

Синтаксис вызова подпрограммы:

```
GOSUB <номер строки 1>|<метка 1>
```

Например, номер метки - 1.

Синтаксис самой подпрограммы:

```
1 REM Подпрограмма для .....
```

```
.....
```

```
<операторы>
```

```
.....
```

```
RETURN [<номер строки 2>|<метка 2>]
```

где <номер строки 1> или <метка 1> - представляют собой номер первой строки подпрограммы, или ее метку, <номер строки 2>|<метка 2> - номер строки или метка возврата в программу. Если она не задана, то возврат происходит на следующий после GOSUB оператор.

Подпрограмма может быть вызвана из программы произвольное число раз. Кроме того, подпрограмма может быть вызвана из другой подпрограммы. Такая вложенность подпрограмм ограничивается только объемом доступной памяти.

Подпрограмма может содержать несколько операторов RETURN, каждый из которых будет осуществлять выход из подпрограммы в определенной точке. Подпрограммы могут располагаться в различных местах программы, однако рекомендуется, чтобы подпрограммы были легко отличимы от основной программы. Для того, чтобы предотвратить случайный вход в подпрограмму, перед ней можно расположить оператор STOP, END, GOTO или расположить ее в конце программы.

Пример:

```
GOSUB 4  
PRINT "STOP"  
END  
4 PRINT "1234567890"  
RETURN
```

RUN - запуск программы со следующей выдачей на экран.

1234567890

STOP

Оператор GOTO

Оператор GOTO осуществляет безусловный переход из обычной последовательности операторов программы на строку программы с заданным номером.

Синтаксис:

GOTO <номер строки>|<метка>

В случае, если <номер строки> или <метки> относится к выполняемому оператору, то выполняется этот оператор и последующие за ним. Если же это невыполняемый оператор, то выполнение программы продолжается с первого выполняемого оператора, встреченного после <номера строки>.

Операторы IF...THEN [...ELSE] и IF...GOTO

Операторы IF...THEN [...ELSE] и IF...GOTO предназначены для изменения порядка выполнения программы в зависимости от значения выражения, следующего за опцией IF.

Синтаксис:

IF <выражение> THEN <оператор(ы)> [ELSE <оператор(ы)>]

или

IF < выражение > GOTO < номер строки|метка > [ELSE <оператор(ы)> | GOTO <номер строки|метка>]

Если значение <выражения> истинно, то выполняется предписание THEN или GOTO. За THEN может следовать либо номер строки, на которую следует осуществить переход, либо один или более выполняемых операторов. За GOTO всегда следует номер строки. Если значение <выражения> ложно, т.е. условие не выполняется, то предписание THEN или GOTO игнорируются и выполняется предписание ELSE, если оно задано. Перед THEN допускается наличие запятой. Допускается вложенность операторов IF ... THEN ... ELSE. Вложенность ограничивается только длиной строки, т.е. до 256 символов.

Например, оператор:

```
IF X>Y THEN PRINT "GREATER" ELSE IF Y>X THEN PRINT  
"LESS THAN" ELSE PRINT "EQUAL"
```

является допустимым оператором. Если оператор состоит из разного количества предписаний ELSE и THEN, то каждому предписанию ELSE соответствует ближайшее THEN, у которого нет пары ELSE.

Например, оператор:

```
IF A=B THEN IF B=C THEN PRINT "A=C" ELSE PRINT "A<>C"
```

не будет печатать "A<>C", когда A<>B.

Когда в операторе IF анализируется значение выражения, являющегося результатом вычислений в формате с плавающей запятой, следует помнить о том, что внутреннее представление величины может быть не точным. Следовательно, проверка должна производиться в пределах точности, в которых может изменяться величина. Например, для того, чтобы сравнить вычисленное значение переменной A с числом 1.0, воспользуемся оператором:

```
IF ABS(A-1.0)<1.0E-6 THEN...
```

Эта проверка возвратит значение "истина" в случае, если значение A совпадает с 1.0 с относительной ошибкой, меньшей, чем 1.0E-6.

Другой пример:

```
IF(I<20)*(I>10) THEN D=100-I GOTO 3  
PRINT "OUT OF RANGE"  
....  
3 REM
```

В этом примере проверяется, имеет ли переменная I значение, большее 10 и меньшее 20. В случае, если значение I находится в этом интервале, то производится вычисление D и осуществляется переход на строку 3. В противном случае выполнение программы продолжается со строки PRINT.

Оператор INPUT

Оператор INPUT позволяет в процессе выполнения программы вводить данные с клавиатуры.

Синтаксис:

```
INPUT[;] [<"наводящая строка">{; | ,}] <список переменных>
```

При достижении оператора INPUT выполнение программы приостанавливается и на экране выводится знак вопроса, который указывает на то, что программа ожидает ввода данных. Если в оператор включена <"наводящая строка">, то она печатается перед знаком вопроса. После этого с клавиатуры производится ввод необходимой информации. Вместо точки с запятой после <"наводящей строки"> можно использовать запятую, которая подавляет печать вопросительного знака.

Введенные данные присваиваются переменным, заданным в <списке переменных>. Число вводимых элементов данных должно соответствовать числу переменных в списке. Элементы данных в списке переменных разделяются запятыми. В качестве имен переменных в списке могут быть заданы переменные с индексами и строковые переменные. Тип каждого вводимого элемента данных должен соответствовать типу, задаваемому именем переменной. Строковые данные для оператора INPUT не нужно заключать в кавычки.

В случае, если в режиме оператора INPUT было задано меньше или больше данных, чем необходимо, или тип данных неверный (числовые вместо строковых и т.п.), выдается сообщение "?Redo from start". До тех пор, пока не будет дан правильный ответ на запрос оператора INPUT, никаких присвоений входных значений не выполняется.

Пример:

```
INPUT "ЗНАЧЕНИЕ X";X  
PRINT X  
END
```

RUN - запуск программы для работы и выдача на экран.

ЗНАЧЕНИЕ X ? 7

Здесь число 7 вводится пользователем с клавиатуры.

Оператор INPUT #

Оператор INPUT# предназначен для считывания элементов данных из открытого последовательного файла на диске и присвоения их переменным программы.

Синтаксис:

```
INPUT#<номер файла>, <список переменных>
```


где:

1. Параметр <номер файла> - представляет собой номер, присвоенный файлу оператором OPEN, который открыл файл для ввода/вывода.

2. Параметр <список переменных> - состоит из имен переменных, которым будут присвоены элементы данных, хранящиеся в файле. (Тип переменной должен соответствовать типу, который задается именем переменной.) В отличие от оператора INPUT оператор INPUT# не выводит на экран знак вопроса.

Элементы данных в файле должны иметь такой же порядок, как если бы они вводились в ответ на запрос оператора INPUT.

Для числовых значений лидирующие коды пробела, возврата каретки и перевода строки игнорируются. Первый встречный символ, отличный от указанных выше считается первым символом числа. Число отделяется от других данных пробелом, возвратом каретки, переводом строки или запятой.

В случае, если компилятор Basic осуществляет поиск строкового элемента в последовательном файле, лидирующие коды пробела, возврата каретки и перевода строки также игнорируются. Первый встречный символ, отличный от указанных считается началом строки. Если этот первый символ является кавычками ("), то строковый элемент данных будет состоять из всех символов, расположенных между первым и вторым символами кавычек. Следовательно, заключенная в кавычки строка символов не может содержать символа кавычек.

В случае, если первый символ строки не является символом кавычек, то это означает, что строка не заключена в кавычки, и она отделяется от других элементов данных запятой, возвратом каретки или переводом строки (кроме того, считывание строкового элемента прекращается после того, как были прочитаны 255 символов.)

Оператор LINE INPUT

Оператор LINE INPUT вводит с клавиатуры целую строку до 254 символов и присваивает ее строковой переменной без использования ограничителей.

Синтаксис:

```
LINE INPUT[;][<"наводящая строка">;] < строковая переменная>
```

где:

Параметр <наводящая строка> - представляет собой строку символов, которая печатается на экране дисплея перед вводом числовых данных. Знак вопроса печатается только, если он является частью текста <наводящей строки>.

Параметр <строковая переменная> - имя переменной, которой присваивается вся введенная информация.

Тем не менее, если была встречена последовательность кодов перевода строки, возврата каретки (именно в этом порядке), то оба эти символа повторяются на экране дисплея, при этом возврат каретки игнорируется, а перевод строки заносится в <строковую переменную>, после чего вывод данных продолжается.

Оператор LINE INPUT#

Оператор LINE INPUT# предназначен для считывания из последовательного файла, расположенного на диске, полной строки до 254 символов и присвоения ее строковой переменной.

Синтаксис:

LINE INPUT#<номер файла>, <строковая переменная>

где:

1. Параметр <номер файла> - представляет собой номер, под которым этот файл был открыт с помощью оператора OPEN.

2. Параметр <строковая переменная> - имя переменной, которой присваивается строка.

Оператор LINE INPUT# считывает из последовательного файла все символы вплоть до кода возврата каретки - Enter. Затем осуществляется пропуск последовательности кодов возврата каретки и перевода строки, а при выполнении следующего оператора LINE INPUT# будут считаны все символы вслед до следующего кода возврата каретки. В случае, если будет встречена последовательность кодов перевода строки, возврата каретки, то эта последовательность сохраняется.

Особенно удобно использовать оператор LINE INPUT# в тех случаях, когда каждая строка файла данных разбита на поля или, когда программа на языке Basic в коде ASCII считывается, как данные другой программой.

Пример:

```
OPEN "O",1,"LIST"  
LINE INPUT"CUSTOMER INFORMATION?";C$
```

```
PRINT#1,C$: CLOSE 1
```

```
OPEN "I",1,"LIST"  
LINE INPUT#1, A$  
PRINT A$: CLOSE 1
```

RUN - запуск программы и результат выдачи на экран.

CUSTOMER INFORMATION?

Ввод с клавиатуры:

JOHN JONES 234,4 MEMPHIS

Вывод на экран:

JOHN JONES 234,4 MEMPHIS

Оператор MOD

Арифметическая операция MOD дает в качестве результата остаток от деления двух числовых выражений.

Синтаксис:

<числовая переменная>=<делимое> MOD <делитель>

Пример:

A = 10 MOD 4

Результат действия оператора MOD:

A = 2

Операторы LPRINT и LPRINT USING

Оператор LPRINT и LPRINT USING предназначены для вывода данных на устройство печати.

Синтаксис:

LPRINT [<список выражений>] [{;}]

или

LPRINT USING <строковое выражение>; <список выражений> [{{,}}].

Эти операторы аналогичны операторам PRINT и PRINT USING за исключением того, что выдача информации производится на печатающее устройство, а не на экран монитора. Оператор LPRINT подразумевает, что ширина печатающего устройства составляет 132 символа.

Операторы ON...GOSUB и ON...GOTO

Операторы ON...GOSUB и ON...GOTO осуществляют переход на одну из заданных строк, номер которой определяется в зависимости от значения вычисленного выражения.

Синтаксис:

ON <выражение> GOSUB <список номеров строк>

или

ON <выражение> GOTO <список номеров строк>

Значение <выражения> определяет, какой из номеров строк в списке, записанном через пробелы, будет использован при переходе. Например, в случае, если значение <выражения> равно трем, в качестве адреса перехода будет взят третий номер в списке. Если значение <выражения> не является целым числом, производится округление дробной части.

В операторе ON...GOSUB каждый номер строки в списке должен представлять собой номер первой строки подпрограммы. В случае, если значение <выражения> равно нулю или превышает число элементов списка (но не больше 255), то выполнение программы будет продолжено со следующего оператора. Если значение <выражения> больше 255, то выдается сообщение об ошибке "Illegal function call".

Оператор OPEN

Оператор OPEN обеспечивает возможность выполнения операций ввода/вывода в расположенный на диске файл.

Синтаксис:

OPEN <режим 1>, [#]<номер файла>, <имя файла> [<длина записи>]

или

OPEN <имя файла> [FOR <режим 2>] [ACCESS <доступ>] [<использование> AS [#] <номер-файла>] [LEN=<длина записи>]

где:

1. Параметр <режим 1> - представляет собой строковое выражение, первым символом которого является один из перечисленных ниже символов:

O - определяет последовательный метод доступа при выводе информации в файл.

I - определяет последовательный метод доступа при вводе информации из файла.

R - определяет прямой метод доступа при вводе и выводе.

B - определяет доступ к двоичному файлу, который может содержать любые из 256 символов кода ASCII.

A - последовательный доступ на вывод после последней уже существующей записи.

2. Параметр <Режим 2> - также представляет собой строковое выражение, определяющее режим ввода/вывода, который совпадает с <режимом 1>, но в отличие от него должен записываться полностью, как приведено ниже:

O - OUTPUT.

I - INPUT.

R - RANDOM.

B - BINARY.

A - APPEND.

3. Параметр <Номер файла> - является целым выражением, значение которого лежит в диапазоне от 1 до 15. В дальнейшем этот номер связан с файлом до тех пор, пока файл открыт, и используется для обращения к файлу при помощи других операторов ввода/вывода.

4. Параметр <Имя файла> - представляет собой строковое выражение, содержащее имя файла (заключенное в кавычки), которое согласуется с правилами относительно имен файлов, принятыми в MS DOS.

5. Параметр <Доступ> - строковое выражение, определяющее условие доступа к файлу:

READ - только на чтение.

WRITE - только на запись.

READ WRITE - и на чтение и на запись (только для RANDOM, BINARY и APPEND).

6. Параметр <Использование> - строковое выражение, определяющее использование файла другими (параллельными) процессами:

SHARED - все процессы могут иметь доступ к файлу.

LOCK READ - доступ к файлу по чтению не доступен процессам.

LOCK WRITE - доступ к файлу по записи не доступен процессам.

LOCK READ WRITE - файл вообще недоступен другим процессам.

7. Параметр <Длина записи> - является целым выражением, которое (если оно задано) устанавливает длину записи для файлов с прямым методом доступа. По умолчанию длина записи принимается равной 128 байтам. Файл может быть открыт для организации последовательного ввода или для прямого доступа под несколькими номерами одновременно. Однако для последовательного вывода файл может быть открыт только под одним номером.

Для того, чтобы стало возможным выполнение операций ввода/вывода с расположенным на диске файлом, этот файл должен быть открыт с помощью оператора OPEN. Этот оператор назначает буфер для выполнения операций ввода/вывода в файл и определяет режим доступа к буферу.

Пример:

```
OPEN "1", 2, "DEMO.DAT"
```

Оператор OPTION BASE

Оператор OPTION BASE предназначен для установки минимального значения индексов массивов.

Синтаксис:

```
OPTION BASE n
```

где n принимает значение 0 или 1. По умолчанию минимальное значение индексов равняется нулю.

В случае, если выполняется оператор

OPTION BASE 1

минимальным значением индексов будет единица.

Оператор PRINT

Оператор PRINT предназначен для вывода информации на дисплей - экран компьютера.

Синтаксис:

PRINT [<список выражений>]

В случае, если <список выражений> отсутствует, печатается пустая строка. Если <список выражений> задан, то на дисплее распечатываются значения выражений. Выражения в списке могут быть числовыми и (или) строковыми. Строки должны быть заключены в кавычки. Позиция каждого печатаемого элемента определяется пунктуацией, используемой для разделения элементов в списке. Компилятор Basic разделяет строку на зоны печати, каждая из которых состоит из 14 позиций. Задание запятой в списке выражений приводит к тому, что следующее значение будет напечатано в начале следующей зоны. Если же задана точка с запятой, то следующее значение выражения будет распечатано непосредственно после предыдущего значения. Задание одного или нескольких пробелов между выражениями приводит к такому же результату, что и задание точки с запятой.

Если запятая или точка с запятой стоят в конце списка выражений, то следующий оператор PRINT будет осуществлять печать значений выражений в той же строке, что и предыдущий оператор PRINT. В случае, если список выражений не заканчивается запятой или точкой с запятой, то в конце каждой строки на печать будет выдаваться код возврата каретки. Если печатаемая строка длиннее, чем длина строки дисплея, то осуществляется переход на следующую физическую строку и печать продолжается. Перед положительным числом и после любого числа всегда помещается пробел. Перед отрицательным числом ставится знак минус.

Числа с обычной точностью, которые могут быть представлены семью и менее цифрами в формате без показателя степени с такой же точностью, что и в формате с показателем степени, выводятся на печать в формате без показателя степени.

Числа с двойной точностью, которые могут быть представлены шестнадцатью и менее цифрами в формате без показателя степени с

такой же точностью, что и в формате с показателем степени, выводятся на печать в формате без показателя степени.

В операторе PRINT вместо ключевого слова PRINT можно использовать вопросительный знак.

Пример:

```
X=5  
PRINT X+5,X-5,X^5  
END
```

RUN - запуск программы и результат на экране.

```
10      0      25
```

В приведенном примере запятые в операторе PRINT указывают на то, что каждое значение должно печататься в начале следующей зоны печати.

Пример:

```
INPUT X  
PRINT X "SQUARED IS" X^2 "AND";  
PRINT X "CUBED IS" X^3
```

RUN - запуск программы и выдача результата на экран.

```
? 9
```

Число 9 вводится с клавиатуры. Тогда на экране будет показан следующий результат:

```
9 SQUARED IS 81 AND 9 CUBED IS 729
```

В этом примере точка с запятой в конце строки после слова "AND" привела к тому, что оба оператора PRINT распечатывают данные в одной строке.

Пример:

```
FOR X=1 TO 3  
J=J+5; K=K+10  
?J;K;  
NEXT X
```

RUN - запуск программы и результат.

5 10 10 20 15 30

В этом примере две точки с запятой в операторе PRINT указывают на то, что каждое значение переменных должно выводиться на печать непосредственно за предыдущим значением. Не надо забывать, что за числом всегда следует пробел, а перед положительными числами также ставится пробел.

Оператор PRINT USING

Оператор PRINT USING осуществляет печать строк или чисел в определенном формате.

Синтаксис:

PRINT USING <строковое выражение>; <список выражений>

где:

1. Параметр <список выражений> - состоит из строковых или числовых выражений, которые необходимо вывести на печать, разделенных точками с запятой или запятыми.

2. Параметр <строковое выражение> - представляет собой строковую константу (или переменную), которая включает в себя специальные символы задания формата.

Эти символы задания формата (смотрите ниже) определяют поле и формат выводимых на печать строк или чисел. В случае, если оператор PRINT USING применяется для вывода строк, то для задания формата поля строки может быть использован один из следующих форматов:

1. Знак "!" - указывает, что на печать необходимо вывести только первый символ заданной строки.

2. Знак "\n пробелов" - указывает, что следует распечатать 2+n символов заданной строки. Если между обратной косой чертой нет пробелов, на печать выводятся два символа строки. Один пробел приведет к выводу трех символов и т.д. В случае, если строка длиннее поля, все дополнительные символы игнорируются. Если поле длиннее строки, то выполняется левостороннее выравнивание строки в поле и дополнение ее пробелами справа.

3. Знак "&" - указывает на то, что строка должна быть распечатана точно в том же виде, в каком она была введена.

Пример:

```
A$="FIVE"  
B$="SEVEN"  
PRINT USING"!";A$;B$  
PRINT USING"\\";A$;B$  
PRINT USING"&";A$;B$
```

RUN - запуск программы и результат на экране.

```
FS  
FIVSEV  
FIVESEVEN
```

Если оператор PRINT USING применяется для печати числовой информации, то для задания формата поля числа используются следующие специальные символы:

1. Знак номера (#) - обозначает позицию каждой цифры. Если выводимое на печать число имеет меньше цифр, чем задано цифровых позиций, то осуществляется правостороннее выравнивание числа в поле (числу будут предшествовать пробелы).

2. Точка (.) - может размещаться в любой позиции поля. В случае, если строка формата указывает на то, что перед десятичной точкой должна находиться цифра, то эта цифра всегда выводится на печать (ноль, если необходимо). В случае необходимости производится округление числа.

3. Знак плюс (+), помещенный в начале или в конце строки задания формата - указывает на то, что перед или после числа необходимо напечатать его знак (плюс или минус);

4. знак минус (-), расположенный в конце поля задания формата - указывает на то, что отрицательные числа должны печататься со знаком минус, расположенным после числа.

5. Два знака звездочка (**), помещенные в начале строки задания формата - указывают на то, что предшествующие пробелы в поле числа должны быть заполнены звездочками. Кроме того, звездочки резервируют позиции для двух дополнительных символов.

6. Два знака \$\$ - приводят к тому, что непосредственно перед числом будет напечатан знак \$. Эти знаки резервируют место для двух символов, одним из которых является \$. Экспоненциальный формат не может быть использован совместно с \$\$. Отрицательные числа могут использоваться только в том случае, если знак минус располагается справа от числа.

7. Символы **\$ в начале строки задания формата - приводят к тому, что предшествующие числу пробелы будут заполнены звездоч-

ками и перед числом будет напечатан знак \$. Символы **\$ задают три дополнительные позиции, одну из которых занимает \$.

8. Запятая (,), расположенная слева от десятичной точки в строке задания формата - приводит к тому, что слева от десятичной точки после каждой третьей цифры слева будет печататься запятая. Запятая, расположенная в конце строки задания формата, выводится на печать, как часть строки. Запятая задает еще одну цифровую позицию. В случае, если запятая используется вместе с экспоненциальным форматом (^^^), она игнорируется.

9. Четыре стрелки вверх (^^^), размещенные после символов цифровых позиций - используются для задания экспоненциального формата. Они отводят место для печати в виде E+xx. Может быть задана произвольная позиция десятичной точки. Осуществляется левостороннее выравнивание значащих цифр и определяется экспонента. Если не задан предшествующий числу знак плюс или следующие за числом знаки плюс или минус, то одна цифровая позиция слева от десятичной точки выделяется для печати числа или знака минус.

10. Знак подчеркивания (_) в строке определения формата - указывает на то, что следующий символ будет выведен на печать, как символьная константа. Для задания символа подчеркивания в качестве строковой константы необходимо в строке определения формата вывода указать два таких символа.

Если число, которое требуется распечатать, превышает размеры отведенного для него числового поля, то перед числом будет напечатан знак процента (%). В случае, если в результате округления число выходит за пределы числового поля, перед округленным числом также будет присутствовать знак процента.

Если число цифровых позиций в строке задания формата превышает 24, то выдается сообщение об ошибке "Illegal function call" ("Недопустимое обращение к функции").

Примеры использования оператора PRINT USING для вывода информации на экран:

```
1) PRINT USING "##.##";.78
```

RUN - запуск программы и результат.

0.78

```
2) PRINT USING "###.##";987.654
```

RUN - запуск программы и результат.

987.65

3) PRINT USING "###.## " ;10.2,5.3,66.789,.234

RUN - запуск программы и результат.

10.20 5.30 66.79 0.23

В последнем примере в конце строки задания формата помещены три пробела, которые предназначены для разделения печатаемых значений в строке.

Операторы PRINT# и PRINT# USING

Операторы PRINT# и PRINT# USING предназначены для записи данных в последовательный файл, расположенный на диске.

Синтаксис:

PRINT#<номер файла>, [USING <строковое выражение>:] <список выражений>

где:

1. Параметр <номер файла> - представляет собой номер, присвоенный файлу в тот момент, когда файл был открыт для вывода с помощью оператора OPEN.

2. Параметр <строковое выражение> - состоит из символов задания формата.

3. Выражения в <списке выражений> - представляют собой числовые и (или) строковые выражения, которые будут записаны в файл.

Оператор PRINT# не производит уплотнения данных в файле на диске. Данные записываются на диск в таком же виде, в каком данные распечатываются на экране дисплея с помощью оператора PRINT.

Поэтому необходимо разграничивать данные на диске таким образом, чтобы было обеспечено их правильное считывание с диска. В списке выражений числовые выражения должны быть разделены точкой с запятой.

Например:

PRINT#1, A; B; C; X; Y; Z

В случае, если в качестве разделителей используются запятые, то дополнительные пробелы, которые помещаются между печатаемыми числами, также будут записываться на диск.

Строковые выражения в списке должны быть разделены точками с запятой. Для того, чтобы на диске строковые выражения были представлены в требуемом формате, следует использовать в списке выражений явно заданные ограничители.

Например:

```
A$="CAMERA": B$="93604-1"
```

тогда оператор

```
PRINT #1, A$, B$
```

запишет на диск строку

```
CAMERA93604-1.
```

В связи с тем, что ограничители отсутствуют, эта информация не может быть введена обратно, как две отдельные строки. Для того, чтобы такая ситуация не возникла, следует ввести в оператор PRINT# явно заданные разделители:

```
PRINT#1, A$, "; "; B$
```

В результате выполнения этого оператора на диск будет записана следующая информация:

```
CAMERA, 93604-1
```

которая может быть прочитана обратно, как значения двух строковых переменных.

Оператор RANDOMIZE

Оператор RANDOMIZE инициирует генератор случайных чисел.

Синтаксис:

```
RANDOMIZE [<выражение>]
```

Если <выражение> опущено, компилятор языка Basic приостанавливает выполнение программы и запрашивает ввод значения с

клавиатуры, печатая при этом на экране дисплея следующее сообщение: Random Number Seed (-32768 to 32767)? ("Начальное значение для генератора случайных чисел (от -32768 до 32767)?").

В случае, если генератор случайных чисел не инициализирован, функция RND всякий раз при выполнении программы будет возвращать одну и ту же последовательность случайных чисел. Для того, чтобы изменить последовательность случайных чисел, генерируемых при каждом выполнении программы, необходимо поместить в начало программы оператор RANDOMIZE и при каждом запуске программы на выполнение изменять его аргумент.

Оператор READ

Оператор READ предназначен для считывания данных, задаваемых оператором DATA, и присваивания их переменным.

Синтаксис:

READ <список переменных>

Оператор READ должен всегда использоваться совместно с оператором DATA. Операторы READ устанавливают взаимно однозначное соответствие между переменными и данными, задаваемыми операторами DATA. Переменные в операторе READ могут быть, как числовыми, так и строковыми, но считываемые значения должны соответствовать заданным типам переменных. Если же такого соответствия нет, то выдается сообщение "Syntax error". Один оператор READ может обращаться к одному или нескольким операторам DATA, и наоборот, несколько операторов READ могут обращаться к одному и тому же оператору DATA. В случае, если число переменных в <списке переменных> превышает число элементов в операторе (операторах) DATA, выдается сообщение "Out of data" ("Данные исчерпаны"). Если заданное число переменных меньше, чем число элементов в операторе (операторах) DATA, то последующие операторы READ начнут считывание данных с первого непрочитанного элемента. Если же последующих операторов DATA нет, то непрочитанные данные игнорируются.

Для того, чтобы повторить сначала считывание данных, заданных операторами DATA, следует воспользоваться оператором RESTORE.

Пример:

```
DATA "ABC,", XYZ, 802111
READ B$,C$,Z
PRINT B$,C$,Z
```

RUN - запуск программы и результат.

ABC, XYZ 802111

Оператор REM

Оператор REM позволяет вводить в программу любые комментарии.

Синтаксис:

REM <примечание>

или

' <примечание>

Операторы REM не выполняются, однако при выдаче листинга программы воспроизводятся точно в том же виде, в котором они были введены в программу. Допускается переход на операторы REM с помощью операторов GOTO и GOSUB. После этого выполнение программы будет продолжено с первого выполняемого оператора, следующего за оператором REM. Комментарии могут добавляться в конце программной строки. Вместо оператора REM перед комментарием можно помещать знак апострофа ('). Однако, комментарии нельзя размещать в конце операторов DATA, поскольку они будут восприниматься как данные.

Пример:

```
REM ВЫЧИСЛЕНИЕ СРЕДНЕЙ СКОРОСТИ
FOR I=1 TO 20
SUM=SUM+V(I)
NEXT I
```

или

```
'ВЫЧИСЛЕНИЕ СРЕДНЕЙ СКОРОСТИ
FOR I=1 TO 20
SUM=SUM+V(I)
NEXT I
```

Оператор RESET

При выполнении этого оператора производится закрытие всех активных (открытых) файлов.

Синтаксис:

RESET

Оператор RESTORE

Оператор RESTORE обеспечивает повторное считывание данных из операторов DATA, начиная с указанной строки.

Синтаксис:

RESTORE [<номер строки>]

После выполнения оператора RESTORE следующий оператор READ считывает первый элемент первого оператора DATA программы. В случае, если задан <номер строки>, следующий оператор READ обращается к первому элементу оператора DATA в указанной строке.

Оператор RUN

Оператор RUN выполняет программу, обрабатываемую в данный момент компилятором.

Синтаксис:

RUN [{<метка строки> | <имя файла>}]

Если оператор имеет параметры, то работа программы начинается с оператора, заданного <меткой строки> или выполняется программа на языке Basic, находящаяся в файле, специфицированном <именем файла>. Если <имя файла> указано без расширения то подразумевается стандартное расширение .BAS.

Оператор STOP

Оператор STOP предназначен для прекращения выполнения программы и возвращения на командный уровень.

Синтаксис:

STOP

Операторы STOP могут размещаться в любых местах программы с целью прекращения ее выполнения. В отличие от оператора END, оператор STOP не закрывает файлы. После выполнения оператора STOP компилятор Basic всегда возвращается на командный уровень.

Выполнение программы может быть продолжено при помощи директивы CONT (только для Quick Basic).

Оператор SUB

Оператор описывает процедуру - подпрограмму, которая может быть откомпилирована отдельно или вместе с головным модулем. В случае раздельной компиляции имеется возможность поместить процедуру в общую библиотеку.

Синтаксис:

```
SUB <имя процедуры> [(<список параметров>)] [STATIC]  
...  
...  
END SUB
```

Чтобы использовать библиотечную процедуру, созданную ранее и помещенную в библиотеку с расширением QLB в интегрированном окружении компилятора следует запустить компилятор Quick Basic с ключом: QB/L и на запрос: "QB QLB file not found" ввести полное имя требуемой библиотеки, которая может находиться в любой директории.

Операторы WHILE ...WEND

Операторы WHILE...WEND предназначены для циклического повторения последовательности операторов до тех пор, пока заданное условие истинно.

Синтаксис:

```
WHILE <выражение>  
...  
[<операторы цикла>]  
...  
WEND
```

В случае, если <выражение> не равно нулю (т.е. истинно), осуществляется выполнение <операторов цикла> до тех пор, пока не будет встречен оператор WEND. Затем происходит возврат к оператору WHILE и выполняется проверка <выражения>. Если оно, по-прежнему, истинно, то <операторы цикла> выполняются вновь. Если же оно не истинно, выполнение программы продолжается с оператором, который непосредственно следует за оператором WEND.

Допускается произвольное число уровней вложенности циклов WHILE.....WEND. Каждый оператор WEND соответствует последнему оператору WHILE. Если у оператора WHILE нет соответствующего ему оператора WEND, выдается сообщение об ошибке "WHILE without END". Если же наоборот оператору WEND не соответствует оператор WHILE, то выдается сообщение "WEND without WHILE".

Оператор WRITE

Оператор WRITE предназначен для вывода информации на дисплей, также как PRINT.

Синтаксис:

WRITE [<список выражений>]

Если <список выражений> не задан, то выводится пустая строка. В случае, если <список выражений> задан, на дисплей выводятся значения выражений. В списке могут быть строковые и числовые выражения, которые должны быть разделены запятыми.

Выводимые строковые выражения заключаются в кавычки. После того, как был распечатан последний элемент <списка выражений>, выдается последовательность кодов возврата каретки и перевода строки.

Оператор WRITE выводит числовые значения в том же формате, что и оператор PRINT, но разделяет их запятыми, а строки заключаются в кавычки.

Пример:

```
A=80: B=90  
C$="THAT'S ALL": WRITE A,B,C$
```

RUN - запуск программы и результат.

```
80, 90, "THAT'S ALL"
```

Оператор WRITE#

Оператор WRITE# осуществляет запись информации в последовательный файл.

Синтаксис:

WRITE# <номер файла>, <список выражений>

где <номер файла> - представляет собой номер, под которым файл был открыт с помощью оператора OPEN для работы в режиме последовательного доступа при вводе.

Входящие в список выражения представляют собой строковые или числовые выражения, разделенные запятыми. Различие между операторами WRITE# и PRINT# заключается в том, что оператор WRITE# по мере записи отдельных элементов на диск вставляет между ними запятые и заключает строки в кавычки. Следовательно, пользователю не надо вводить в список явно заданные ограничители. После записи на диск последнего элемента списка осуществляется запись кодов возврата каретки и перевода строки.

Например, пусть:

```
A$="CAMERA": B$="93604-1"
```

Тогда оператор

```
WRITE#1, A$, B$
```

запишет на диск следующую информацию:

```
"CAMERA", "93604-1"
```

Последующий оператор

```
INPUT#1, A$, B$
```

присвоит переменной A\$ символы

```
"CAMERA"
```

а переменной B\$ - символы

```
"93604-1"
```

ВСТРОЕННЫЕ ФУНКЦИИ

В этом разделе описываются встроенные функции компилятора языка Basic. К этим функциям можно обращаться из любой программы, написанной на языке Basic без их предварительного определения. Аргументы функций всегда заключаются в круглые скобки. При описании синтаксиса функций используются следующие обозначения их аргументов:

1. X и Y - произвольные числовые выражения.
2. I и J - целые выражения.
3. X\$ и Y\$ - строковые выражения.

В случае, если вместо целого выражения задано выражение, представленное в формате с плавающей точкой, то его значение округляется и затем используется полученное целое число. Функция обычно возвращает - выдает в качестве результата работы, только целые значения или значения обычной точности.

Функция ABS

Функция ABS возвращает беззнаковое абсолютное значение числа X. Синтаксис:

ABS (X)

Пример:

```
PRINT ABS(7*(-5))
```

RUN - запуск программы и результат.

35

Функция ATN

Функция ATN возвращает заданное в радианах значение арктангенса X, которое находится в пределах от $-\pi/2$ до $\pi/2$.

Синтаксис:

ATN(X)

Выражение X может иметь любой числовой тип.

Вычисление арктангенса выполняется с обычной точностью для аргументов INTEGER и SINGLE, и с двойной точностью для DOUBLE.

Пример:

```
INPUT X  
PRINT ATN(X)
```

RUN - запуск программы.

? 3

Ввод числа 3 с клавиатуры и результат.

1.249046

Функция CINT

Функция CINT преобразует значение X в целое число путем округления дробной части.

Синтаксис:

```
CINT(X)
```

Если значение X не лежит в пределах от -32768 до 32767, то возникает ошибка "Overflow".

Пример:

```
PRINT CINT(45.67)
```

RUN - запуск программы и результат.

46

Функция COS

Функция COS возвращает значение косинуса X, заданного в радианах.

Синтаксис:

```
COS(X)
```

Вычисление значения COS(X) выполняется с обычной точностью.

Пример:

```
X=2*COS(0.4)
PRINT X
```

RUN - запуск программы и результат.

1.842122

Функция EXP

Функция EXP возвращает результат возведения числа $e=2.718282$ в степень X.

Синтаксис:

```
EXP(X)
```

Значение X не должно превышать 87.3365. В случае, если при возведении в степень происходит переполнение, выдается сообщение об ошибке "Overflow" и в качестве результата принимается максимально возможное число, и выполнение программы продолжается. В Turbo Basic после выдачи сообщения об ошибке работа программы останавливается.

Пример:

```
X=5
PRINT EXP(X-1)
```

RUN - запуск программы и результат.

54.59815

Функция INPUT\$

Функция INPUT\$ возвращает строку, состоящую из X символов, которые считываются с клавиатуры или из файла с номером Y.

Синтаксис:

```
INPUT$(X[, [#]Y)
```

Если клавиатура используется для ввода, символы на экране не отображаются и все управляющие символы игнорируются.

Пример:

```
100 PRINT "ВВЕДИТЕ Р ДЛЯ ПРОДОЛЖЕНИЯ ИЛИ S ДЛЯ  
ПРЕКРАЩЕНИЯ"  
110 X$=INPUT$(1)  
120 IF X$="P" THEN 140  
130 IF X$="S" THEN END ELSE 100  
140 ....
```

Функция INT

Функция INT возвращает максимальное целое число, не превышающее X, т.е. округляет X в меньшую сторону.

Синтаксис:

INT(X)

Примеры:

1) PRINT INT(99.89)

RUN - запуск программы и результат.

99

2) PRINT INT(-12.11)

RUN - запуск программы и результат.

-12

Функция LOG

Функция LOG возвращает значение натурального логарифма числа X.

Синтаксис:

LOG(X)

Значение X должно быть больше нуля.

Пример:

PRINT LOG(45/7)

RUN - запуск программы и результат.

1.860752

Функция RND

Функция RND возвращает случайное число в диапазоне от 0 до 1.

Синтаксис:

RND[(X)]

Всякий раз при запуске программы на выполнение функция RND будет генерировать одну и ту же последовательность случайных чисел, если только не будет инициализирован генератор случайных чисел RANDOMIZE. Если аргумент X имеет отрицательное значение, функция RND будет всегда выдавать одну и ту же последовательность случайных чисел при различных X. Если аргумент X не задан или $X > 0$, будет выполнена генерация следующего случайного числа из последовательности случайных чисел. В случае, если $X = 0$, происходит повторение последнего сгенерированного случайного числа.

Пример:

```
FOR I=1 TO 5  
PRINT INT(RND*100);  
NEXT
```

RUN - запуск программы и результат.

24 30 31 51 5

Функция SIN

Функция SIN возвращает значение синуса аргумента X, заданного в радианах.

Синтаксис:

SIN(X)

Функция SIN(X) вычисляется с обычной точностью. Кроме того:

$\text{COS}(X) = \text{SIN}(X + 3.14159265/2)$.

Пример:

```
PRINT SIN(1.5)
```


RUN - запуск программы и результат.

0.9974951

Функция SQR

Функция SQR возвращает значение квадратного корня аргумента X.

Синтаксис:

SQR(X)

Значение X должно быть положительным.

Пример:

```
FOR X=10 TO 25 STEP 5  
PRINT X, SQR(X)  
NEXT X
```

RUN - запуск программы и результат.

```
10 3.162278  
15 3.872984  
20 4.472136  
25 5
```

Функция TAN

Функция TAN возвращает тангенс аргумента X, заданного в радианах.

Синтаксис:

TAN(X)

Значение TAN(X) вычисляется с обычной точностью. Если при вычислении функции TAN произошло переполнение, выдается сообщение об ошибке "Overflow", и в качестве результата возвращается максимально возможное число с соответствующим знаком, и выполнение программы продолжается. В Turbo Basic после такой диагностики работа программы прекращается.

ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

В этом разделе описываются дополнительные возможности компилятора языка Basic. К ним относятся операторы и функции, предназначенные для установления режима работы дисплея, работы с графикой и звуковым устройством, и других целей.

Оператор DATES\$

Оператор DATES\$ используется для установки или изменения текущей даты.

Синтаксис:

DATE\$ = <строковое выражение>

или

<строковое выражение> = DATE\$

где <строковое выражение> представляет собой строковую константу или выражение.

Первая форма оператора DATES\$ позволяет устанавливать текущую дату, а вторая форма предназначена для определения текущей даты. Текущая дата определяется и присваивается строковой переменной, если DATE\$ входит в выражение, используемое в операторе LET или PRINT. Если DATE\$ находится слева от знака равенства в операторе присваивания, то происходит установка текущей даты.

Если используется первая форма оператора, то строковое выражение может быть представлено в одном из следующих форматов:

1. "МММ-ДД-ГГ"
2. "ММ/ДД/ГГ"
3. "ММ-ДД-ГГГ"

Пример:

```
DATE$="01-01-81": PRINT DATE$  
RUN - запуск программы и результат.
```

```
01-01-81
```

При использовании второй формы оператора, переменной присваивается 10 символов в формате

"мм-дд-гггг"

где мм - месяц (от 01 до 12), дд - день (от 01 до 31) и гггг - год (от 1980 до 2099).

Если <строковое выражение> является неправильным, то выдается сообщение об ошибке "Type mismatch", и сохраняется прежнее значение.

Оператор TIMES

Оператор TIMES\$ используется для установки или изменения текущего времени.

Синтаксис:

TIMES\$ = <строковое выражение>

или

<строковое выражение> = TIMES\$

где параметр <строковое выражение> - представляет собой строковую константу или переменную.

Первая форма оператора TIMES\$ применяется для установки текущего времени, а вторая форма - для определения текущего времени. Если TIMES\$ входит в выражение, используемое в операторе LET или PRINT, то определяется текущее время и присваивается соответствующей строковой переменной.

При использовании оператора первой формы строковое выражение может быть задано в одном из следующих видов:

1. "чч" - установка часа. Минуты и секунды по умолчанию принимаются равными 00.
2. "чч:мм" - установка часа и минут. Секунды по умолчанию принимаются равными 00.
3. "чч:мм:сс" - установка часа, минут и секунд.

Пример:

```
TIMES$="08:00"  
PRINT TIMES$
```

RUN - запуск программы и результат.

08:00:04

При использовании второй формы переменной присваивается 8 символов в формате:

"чч:мм:сс"

где чч - час (от 00 до 23), мм - минуты (от 00 до 59) и сс - секунды (от 00 до 59).

При неправильном задании значений выдается сообщение об ошибке "Illegal Function Call" и сохраняется установленное ранее время.

Оператор COLOR

Оператор COLOR устанавливает цвет для отображения информации, а также цвета фона и границ экрана дисплея. Синтаксис оператора зависит от текущего режима работы дисплея, который устанавливается оператором SCREEN.

В алфавитно - цифровых режимах работы дисплея синтаксис оператора COLOR:

COLOR [<отображение>][,<фон>][,<границы>]],

где:

1. Параметр <отображение> - целое беззнаковое число от 0 до 31, устанавливающее цвет отображаемых символов.
2. Параметр <фон> - целое беззнаковое число от 0 до 15, определяющее цвет фона.
3. Параметр <границы> - целое беззнаковое число от 0 до 15, устанавливающее цвет границ.

Цвета задаются следующим образом:

1. 0 - черный.
2. 1 - синий.
3. 2 - зеленый.
4. 3 - бирюзовый.
5. 4 - красный.
6. 5 - сиреневый.
7. 6 - коричневый.
8. 7 - светло серый.

9. 8 - темно серый.
10. 9 - светло синий.
11. 10 - светло зеленый.
12. 11 - светло бирюзовый.
13. 12 - светло красный.
14. 13 - светло сиреневый.
15. 14 - желтый.
16. 15 - белый.

Любые значения параметров оператора COLOR, лежащие вне указанных пределов, приводят к ошибке "Illegal Function Call" и сохраняются прежние значения параметров.

Значения параметра отображения в диапазоне от 0 до 15 устанавливают цвет, а значения от 16 до 31 устанавливают соответствующие цвета от 0 до 15 и вводят режим мигания символов.

Любой из параметров оператора COLOR может быть опущен. В этом случае значения не заданных параметров сохраняются прежними.

Например:

```
COLOR 15,0,0
```

Устанавливает белый цвет отображения информации на экране, черный цвет границ и фона.

Оператор CLS

Оператор CLS предназначен для очистки активной страницы экрана дисплея.

Синтаксис:

```
CLS [<{0|1|2}>]
```

Если дисплей в данный момент времени находится в алфавитно - цифровом режиме работы, то активная страница экрана после очистки будет иметь цвет фона.

Если дисплей находится в графическом режиме работы (со средней или высокой разрешающей способностью), то после очистки экран будет черным.

Выполнение операторов SCREEN и WIDTH может привести к очистке экрана, если устанавливаемый режим работы дисплея отличается от текущего режима работы. Совместно с оператором CLS могут быть использованы уточняющие параметры (только для Quick Basic):

1. Параметр CLS 0 - полная очистка экрана (и текст и графика).
2. Параметр CLS 1 - если исполнялся оператор VIEW, то графическое окно стирается, иначе стирается весь экран.
3. Параметр CLS 2 - стирается только текстовое окно. Последняя строка экрана не изменяется.

Оператор SCREEN

Оператор SCREEN устанавливает режим работы дисплея.

Синтаксис:

```
SCREEN [<режим>] [,<отображение>] [,<страница 1>] [,<страница 2>]]],
```

где:

1. Параметр <режим> - числовое выражение, значение которого должно быть равно 0, 1 или 2. Эти значения определяют режим работы дисплея:

- 1) 0 - алфавитно - цифровой режим с текущей шириной экрана (40 или 80 символов).
- 2) 1 - графический режим работы со средней разрешающей способностью 320x200 точек (цветная графика).
- 3) 2 - графический режим работы с высокой разрешающей способностью 640x350 точек.

2. <отображение> - числовое выражение. Если его значение равно нулю, то цвета подавляются, и информация отображается в черно - белом режиме. Ненулевое значение разрешает цветное отображение.

3. <страница 1> - используется только в алфавитно - цифровом режиме. Числовое выражение должно быть целым числом в диапазоне от нуля до 7 для экрана шириной в 40 символов или в диапазоне от нуля до 3 для экрана шириной в 80 символов. Этот параметр выбирает страницу экрана, на которую осуществляется запись информации.

4. <страница 2> - используется только в алфавитно - цифровом режиме.

Если все параметры заданы правильно, то устанавливается новый режим работы дисплея и происходит очистка экрана. В качестве цвета отображения выбирается белый цвет, а в качестве цвета фона и границ экрана - черный цвет.

Если установленный оператором SCREEN режим работы дисплея совпадает со старым, то ничего не происходит. Если значения параметров оператора SCREEN выходят за указанные границы, то выдается сообщение "Illegal Function Call", и сохраняются прежние значения.

Оператор PALETTE

Оператор PALETTE выбирает палитру цветов.

Синтаксис:

PALETTE [<атрибут>,<цвет>]

или

PALETTE USING <имя массива> [(<индекс>)]

где:

1. Параметр <атрибут> - определяет номер палитры - 1 (темная) или 2 (светлая).
2. Параметр <цвет> - номер цвета. Для VGA и MGA мониторов требуется использовать длинное целое число при режимах SCREEN 11-13.
3. Параметр <имя массива> - имя массива содержащего номера цветов.
4. Параметр <индекс> - определяет элемент массива цветов для установки палитры.

Оператор KEY

Оператор KEY позволяет присвоить одной или всем десяти функциональным клавишам 15 - ти байтную строку.

Несколько вариантов синтаксиса:

1. KEY <номер клавиши>, <строковое выражение>
2. KEY LIST
3. KEY ON
4. KEY OFF
5. KEY(n) <{ON | OFF | STOP}>

Первая форма оператора KEY позволяет присвоить строковое выражение указанной функциональной клавише. <Строковое выра-

жение> может иметь длину от одного до 15 символов. Если его длина больше, символы после пятнадцатого игнорируются.

Оператор KEY ON распечатывает в 25 - ой строке экрана дисплея значения клавиш. Когда ширина экрана равняется 40 символам, на экране распечатываются значения пяти клавиш.

При ширине экрана в 80 символов распечатываются значения всех десяти клавиш. При использовании оператора KEY OFF удаляется 25 - ая строка на экране дисплея. Оператор KEY LIST распечатывает на экране значения всех десяти функциональных клавиш.

Если значение выражения <номер клавиши> выходит за пределы диапазона от 1 до 10, то выдается сообщение "Illegal Function Call". При этом сохраняются прежние значения функциональных клавиш.

Примеры:

1. KEY(n) ON - задействует функциональную клавишу n.
2. KEY(n) OFF - исключает заданную клавишу.
3. KEY(n) STOP - приостанавливает реакцию на клавишу n до ее повторной активизации по KEY(n) ON.

При этом все нажатия этой клавиши в состоянии приостановки запоминаются и становятся последовательно доступными при активизации. Здесь n - условный номер клавиши:

1. 1 - 10 клавиши F1 - F10.
2. 11 клавиша UP.
3. 12 клавиша LEFT.
4. 13 клавиша RIGHT.
5. 14 клавиша DOWN.
6. 15 - 25 клавиши определяемые пользователем.
7. 30 - 31 клавиши F11 - F12 на стандартной 101 - клавишной клавиатуре.

Оператор BEEP

Оператор BEEP позволяет генерировать звуковой сигнал частотой 800 Гц и длительностью 1/4 секунды.

Синтаксис:

BEEP

Пример:

IF X<20 THEN BEEP

Оператор SOUND

Оператор SOUND предназначен для генерации звуковых сигналов.

Синтаксис:

SOUND <частота, длительность>

где:

1. Параметр <частота> - частота в герцах, представляющее собой целое выражение, значение которого должно быть в диапазоне от 37 до 32767.

2. Параметр <длительность> - длительность в тактах, представляет собой целое выражение, значение которого должно быть от 0 до 65535.

В одной секунде 18.2 такта. Если задана нулевая длительность, то любой выполняемый оператор SOUND прекращает свою работу.

Пример:

SOUND RND*1000+37,2.

Оператор PLAY

Оператор PLAY предоставляет дополнительные возможности в использовании встроенного в ЭВМ звукового устройства.

Синтаксис:

PLAY <строковое выражение>

<Строковое выражение> состоит из следующих обозначающих команды символов:

1. A - G[#|+,-] - играть ноту. Символ "#" или "+" после буквы, определяющей ноту, означает диэз, а символ "-" означает бемоль.

2. L<n> - длительность. Устанавливает длительность каждой ноты. L4 означает четверть ноты, L1 - целую ноту и т.д. Значение n должно быть в пределах от 1 до 64. Длительность может также следовать за определенной нотой, в случае, если необходимо изменить длительность только этой ноты. В этом случае A16 эквивалентно L16A.

3. MF - означает, что каждая последующая нота или звук, генерируемые операторами PLAY или SOUND, не будут звучать до тех

пор, пока не закончится звучание предыдущей ноты или звука. Этот режим устанавливается по умолчанию.

4. MB - означает, что каждая нота или звук помещается в буфер, позволяя продолжать выполнение программы во время звучания музыки.

В буфер можно поместить до 32 нот (или пауз):

1. MN - каждая нота играется $7/8$ времени, заданного числом L (длительность).

2. ML - каждая нота звучит полный период времени, заданный числом L.

3. MS - каждая нота играется $3/4$ времени, заданного L.

4. N<n> - играть ноту n. Значение n должно быть в диапазоне от 0 до 84. В семи возможных октавах существует 84 ноты, причем $n=0$ означает отсутствие звучания.

5. O<n> - октава. Устанавливает текущую октаву. Существует 7 октав (0...6).

6. P<n> - пауза. Значение n может изменяться от 1 до 64.

7. T<n> - темп. Устанавливает число четвертей ноты (L4) в секунду. Значение n может изменяться от 32 до 255. По умолчанию оно равняется 120.

8. X<строка> - выполнить подстроку.

Точка после каждой ноты приводит к тому, что нота будет играть $3/2$ периода времени, определяемого произведением L (длительность) и T (темп). За нотой может следовать многоточие. При этом период времени звучания ноты масштабируется соответствующим образом. Так A. приводит к увеличению периода звучания в $3/2$ раза; A.. - в $9/4$ раза, A... - в $27/8$ раза и т.д.

Точки могут также помещаться после P (пауза) и при этом длительность паузы масштабируется, как описано выше. Числовой аргумент <n> может быть числовой константой или последовательностью символов "<имя переменной>".

Пример:

```
A$="BB - C": B$="O4XA$;"  
C$="L1CT50N3N4N5N6"  
PLAY"P2XA$;XB$;XC$;"
```

Приведем и некоторые другие варианты записи оператора:

1. PLAY(n) - возвращает количество нот в музыкальной очереди.

2. PLAY ON - разрешает воспроизведение музыки.
3. PLAY OFF - запрещает воспроизведение музыки.
4. PLAY STOP - приостанавливает воспроизведение музыки.
5. ON PLAY (n) GOSUB <метка> - вызывает переход на метку, когда воспроизведены ноты из буфера с n по n - 1.

ПРИМЕРЫ ПРОГРАММ НА BASIC

Вычисление значений функции

Выберем в качестве функции простое выражение:

$$f(x)=ax+b$$

где a и b параметры, определяющие вид кривой этой функции.

Выполним вычисление значений функции в 11 точках, включая нулевую на интервале 0 - 1. Примем, например, для параметров следующие значения: $a=2$ и $b=3$.

Ниже приведен текст программы для вычисления такой функции. Операторы программы даны заглавными буквами, а пояснения к ним - строчными.

REM ---- Программа для вычисления значений функции ---

CLS - оператор очистки экрана.

DEFDBL A-Z - оператор двойной точности.

DIM V(100) - оператор массива.

B=1 - конечное значение интервал.

A=0 - начальное значение интервал.

N=10 - число шагов.

H=(B-A)/N - величина шага.

AA=2 - величина a .

BB=3 - величина b .

S=0

FOR I=0 TO N - оператор цикла.

X=H*I - текущее значение величины x .

V(I)=AA*X+BB - текущее значение функции.

PRINT X,V(I) - оператор печати.

NEXT - конец цикла.

END - конец программы.

В результате работы программы на экран выводится два столбика - в первом величина аргумента, во втором самой функции:

0.0 3.0

0.1	3.2
0.2	3.4
0.3	3.6
0.4	3.8
0.5	4.0
0.6	4.2
0.7	4.4
0.8	4.6
0.9	4.8
1.0	5.0

Не трудно проверить, что получены правильные значения функции. В частности, при $x=0$ функция равна 3, при $x=1$ функция равна 5.

Вычисление суммы значений функции

Процедура вычисления суммы значений функции в N точках записывается в виде:

$$S = \sum_{i=0}^N f(x_i) ,$$

где N число шагов при вычислении суммы, $f(x_i)$ - значения функции в точке i и $x_i = h*i$ - величина аргумента в точке i . Шаг вычисления h определяется в виде $(b-a)/N$, где a и b верхняя и нижняя границы интервала вычисления функции.

В качестве функции выберем $\sin(x)$ и будем считать сумму на интервале $0 - 1$ с числом шагов 101, включая ноль. Ниже приведен текст программы для нахождения такой суммы:

```
REM --- Программа для вычисления суммы значений функции --
CLS
DEFDBL A-Z
DIM V(100)
V=1 - конечное значение интервала.
A=0 - начальное значение интервала.
N=100 - число шагов.
H=(V-A)/N - шаг вычисления.

S=0
```

```
FOR I=0 TO N
X=H*I
V(I)=SIN(X)
S=S+V(I)
NEXT

PRINT S

END
```

В результате вычислений суммы получим величину:

46.39012182353973.

Запись и считывание информации на диске

Запись и считывание информации производится оператором OPEN с различными параметрам, определяющими его режим работы. Ниже приведена программа, в которой проводится вычисление массива значений функции с последующей записью в файл на диск и считывание этого массива с последующим выводом его на экран. В качестве функции выбран Sin(x), а вычисления проводятся в 11 точках на интервале от 0 до π .

```
REM ----- Программа записи и считывания с диска -----
```

```
DEFDBL A-Z
DIM U(100), V(100)
CLS
```

```
G$="C:\BASICA\TB\FAIL.BAS"
```

```
A=0 - начальное значение интервала.
B=3.141592653589793 - конечное значение интервала, равное  $\pi$ .
N=10 - число шагов.
H=(B-A)/N - шаг вычисления.
```

```
OPEN "O",1,G$
```

```
FOR L=0 TO N
R=L*H
U(L)=SIN(R)
PRINT#1, USING " +#.#####^ ^ ^ ^ ";R,U(L)
```

```

NEXT
CLOSE

OPEN "I",1,G$
FOR I=0 TO N
INPUT#1, R,V(I)
PRINT USING " +#.#####^ ^ ^ ^ ";R,V(I)
NEXT

END

```

В результате на экран выводятся значения аргумента (левый столбец) и функции, которые были считаны с диска:

```

+0.000000E+00 +0.000000E+00
+3.141593E-01 +3.090170E-01
+6.283185E-01 +5.877853E-01
+9.424778E-01 +8.090170E-01
+1.256637E+00 +9.510565E-01
+1.570796E+00 +1.000000E+00
+1.884956E+00 +9.510565E-01
+2.199115E+00 +8.090170E-01
+2.513274E+00 +5.877853E-01
+2.827433E+00 +3.090170E-01
+3.141593E+00 +8.979318E-11

```

Не сложно проверить, что величина функции при $R = \pi/2 = 1.570796E+00$ равна 1, а при $R = \pi = 3.141593E+00$ должна быть равна нулю. Последнее условие выполняется с абсолютной точностью примерно 10^{-10} .

Вычисление интеграла

Рассмотрим теперь метод и программу для вычисления интеграла функции, заданной массивом переменных в N точках с использованием метода Симпсона. Такой интеграл может быть представлен в виде суммы вида:

$$\int_a^b f(x)dx = \frac{b-a}{3N} \left[\{f(x_0) + f(x_N)\} + 4 \sum_{i=1}^{N-1} f(x_i) + 2 \sum_{i=2}^{N-2} f(x_i) \right],$$

где:

1. Величина N - число шагов интегрирования, a и b - пределы интервала интегрирования.

2. Запись $f(x_i)$ - обозначает значения функции на i -том шаге, т.е. при $x_i = h \cdot i$. Здесь h - шаг интегрирования, который определяется в виде $h = (b-a)/N$.

Первая сумма выражения вычисляется по нечетным, а вторая по четным значениям i . В качестве функции выбрано $2\sin^2(x)$ на интервале интегрирования $0 - \pi$. Вычисления проводятся в 101 точке, т.е. $N = 100$. Значение этого интеграла известно и равно $\pi = 3,141592653589793$.

Программа наглядно демонстрирует, как можно повысить точность вычислений при использовании оператора двойной точности. В этом случае вполне можно получить результат с точностью до 16 значащих чисел, т.е. до 15 чисел после запятой при не очень большом значении N . А выключение этого оператора приводит к точности только в 7 значащих чисел.

REM ----- Программа для вычисления интеграла методом Симпсона -----

CLS - оператор очистки экрана перед выдачей результатов.

DEFDBL A-Z - оператор задания двойной точности.

DIM V(1000) - оператор задания размерности массива.

PI=3.141592653589793 - присвоение значения числу π .

N=100 - определение числа шагов интегрирования.

H=PI/N - вычисление шага интегрирования.

FOR I=0 TO N - цикл для вычисления значений функции.

X=H*I - вычисление значений аргумента.

V(I)=2*(SIN(X))^2 - вычисление значений функции.

NEXT - конец цикла.

A=0: B=0 - обнуление начальных значений переменных.

FOR II=1 TO N-1 STEP 2 - цикл вычисления первой суммы.

B=B+V(II)

NEXT II - конец цикла.

FOR JJ=2 TO N-2 STEP 2 - цикл вычисления второй суммы

A=A+V(JJ)

NEXT JJ - конец цикла.

SINT=H*(V(0)+V(N)+2*A+4*B)/3 - окончательное нахождение интеграла.

PRINT SINT - печать результата.

END - конец программы.

Если вычисления проводить с двойной точностью, то в результате получим значение: 3.141592653589793, что полностью совпадает с приведенной выше величиной π . Если же убрать оператор двойной точности, то результат будет: 3.141592502593994. Тем самым видно, что совпадение наблюдается только у 6 чисел после запятой.

Конечно, на точность результатов влияет и величина N, но выбранный значение N и применение оператора DEFDBL позволяют правильно передать до 15 чисел после запятой.

Заметим, что без оператора двойной точности DEFDBL даже при N=1000 получаем: 3.141592741012573, т.е. и в этом случае правильно передаются только 6 знаков.

Вычисление произведения двух матриц

Для нахождения матричных элементов произведения квадратных матриц размерности n

$$A=BC$$

используется выражение:

$$a_{ij} = \sum_{k=1}^N b_{ik} c_{kj}$$

где a_{ij} , b_{ik} и c_{kj} - матричные элементы.

Программа вычисления такого произведения приведена ниже:

REM ----- Программа вычисления произведения матриц -----

DEFDBL A-Z

DIM A(20,20), B(20,20), C(20,20)

B(1,1)=1: B(1,2)=2: B(1,3)=3: B(1,4)=4: B(2,1)=2: B(2,2)=2

B(2,3)=2: B(2,4)=2: B(3,1)=3: B(3,2)=2: B(3,3)=3: B(3,4)=2

B(4,1)=4: B(4,2)=2: B(4,3)=2: B(4,4)=4

C(1,1)=1: C(1,2)=3: C(1,3)=5: C(1,4)=7: C(2,1)=3: C(2,2)=2

$C(2,3)=2$: $C(2,4)=2$: $C(3,1)=5$: $C(3,2)=2$: $C(3,3)=3$: $C(3,4)=3$
 $C(4,1)=7$: $C(4,2)=2$: $C(4,3)=3$: $C(4,4)=4$

N=4

```
FOR I=1 TO N
FOR J=1 TO N
FOR K=1 TO N
A(I,J)= A(I,J)+B(I,J)*C(K,J)
NEXT K
NEXT J
NEXT I
```

```
PRINT "МАТРИЦА B"
FOR I=1 TO N
FOR J=1 TO N
PRINT B(I,J)
NEXT
NEXT
```

```
PRINT "МАТРИЦА C"
FOR I=1 TO N
FOR J=1 TO N
PRINT C(I,J)
NEXT
NEXT
```

```
PRINT "МАТРИЦА A"
FOR I=1 TO N
FOR J=1 TO N
PRINT A(I,J)
NEXT J
NEXT I
```

END

В результате работы программы получаем искомую матрицу:

МАТРИЦА B

```
1 2 3 4
2 2 2 2
3 2 3 2
4 2 2 4
```

МАТРИЦА С

1 3 5 7
 3 2 2 2
 5 2 3 3
 7 2 3 4

МАТРИЦА А

50 21 30 36
 32 18 26 32
 38 23 34 42
 48 28 42 54

Результат можно проверить вручную, используя известные правила - элемент, стоящий на I строке и J столбце искомой матрицы равен сумме произведения соответствующих элементов I строки матрицы В и элементов J столбца матрицы С.

Поиск корня функции

Корнем функции называется величина ее аргумента, при котором функция равна нулю. Возьмем простую функцию $\text{Cos}(x)$ - известно, что ее первый корень при $x > 0$ находится при $x = \pi/2$.

Равенство нулю какой-то величины в численной математике означает, что эта величина меньше заранее заданного числа ϵ - точности вычисления корня. Ниже приведена программа для нахождения корня такой функции. На печать выводится не x , а величина $2x$, равная π . Это позволяет легко сравнить полученную в результате расчетов величину с известным значением числа π , которое приводится в одном из предыдущих примеров и равно 3.141592653589793.

Пределы поиска корня (SKN - верхний и SKV - нижний) задаются в радианах. Начальный шаг поиска - HC выбран произвольно, поскольку его величина практически не влияет на результаты. Точность вычислений - EP задана максимальная даже для режима двойной точности.

REM -----Программа поиска корня функции -----

DEFDBL A-Z
 CLS

SKN=0 - нижний предел поиска.

SKV=2 - верхний предел поиска.

HC=.01 - шаг поиска.
EP=1.0E-16 - точность поиска корня.
A2=SKN
DK=A2
GOSUB 1000
D12=DD

B2=A2+HC
51 DK=B2

GOSUB 1000
D11=DD

IF D12*D11>0 GOTO 4
3 A3=A2
B3=B2
11 C3=(A3+B3)/2
IF (ABS(A3-B3))<EP GOTO 151

DK=C3
GOSUB 1000
F2=DD

IF D12*F2>0 GOTO 14
B3=C3
D11=F2
GOTO 15
4 A3=C3
D12=F2
15 IF ABS(F2)>EP GOTO 11

151 CO=C3

GOTO 7
4 IF ABS(D11*D12)<EP GOTO 3
A2=A2+HC
B2=B2+HC
D12=D11
IF B2-SKV<+0.1 GOTO 51
YS=SKV
GOTO 8
7 YS=NC
8 REM

```
PRINT "2*KOR-FUN=";2*CO
PRINT "VEL-FUN=";F2
```

```
END
```

```
1000 REM
DD=COS(DK) - подпрограмма вычисления функции.
RETURN
```

В результате работы программы на экран выдается следующий результат:

```
2*KOR-FUN=3.141592653589793
VEL-FUN= 6.1257422745431E-017
```

Двойная величина корня функции найдена точно при значении π . Значит, корень находится при $\pi/2$. Величина самой функции, когда ее аргумент равен корню, оказывается меньше 10^{-16} .

Приведем пример дугой программы для поиска корня, который использует метод Ньютона:

```
REM -- Программа поиска корня функции методом Ньютона ----
```

```
DEFDBL A-Z
CLS
```

```
X=2 - начальное значение аргумента.
```

```
E=1.0E-16 - точность вычислений.
```

```
30 CALL FUN(X,F)
```

```
L=F
```

```
X=X+E
```

```
CALL FUN(X,F)
```

```
L=E*L/(F-L)
```

```
X=X-L-E
```

```
IF ABS(L)>E THEN 30
```

```
PRINT "2*X=";2*X,
```

```
CALL FUN(X,F)
```

```
PRINT "FUN=" ;F
```

```
END
```

```
SUB FUN(X,F)
```

```
F=COS(X)
```

```
END SUB
```

В результате получаем:

$2 * X = 3.141592653589793$
 $FUN = 6.1257422745431E-017$

Двойное значение корня, как и в предыдущем случае точно равно числу π , а значение самой функции меньше 10^{-16} . В данном случае и в предыдущем примере не нулевое значение функции обусловлено пределом точности вычислений, который обеспечивает только 16 значащих цифр.

Нахождение минимума функции

Ниже приведена программа для нахождения минимума функции одного аргумента. В качестве функции выбран $\sin(x)$. Известно, что его минимум, равный -1 , находится при $3/2 \pi$. Эта величина выводится в первом столбце результатов. Во втором столбце дается значение аргумента x , при котором найден минимум функции и в третьем - значение самой функции в минимуме.

Вычисление значений функции здесь организовано через подпрограмму GOSUB, но эта процедура может быть записана и через оператор DEF FN.

REM ----- Программа нахождения минимума функции -----

CLS
DEFDBL A-Z

H=.1 - шаг поиска минимума.
Z=.2 - начальное значение аргумента.
E=1E-16 - точность поиска.

115 X=Z
GOSUB 200
V=F

X=Z-H
GOSUB 200
W=F

X=Z+H
GOSUB 200
U=F

```
T=W*(2*Z+H)-4*V*Z+U*(2*Z-H)
T=T/(W-2*V+U)/2
IF ABS(T-Z)<E THEN 180
Z=T
GOTO 115
```

```
180 PRINT 3*3.141592653589793/2,T,V
```

```
STOP
```

```
200 REM
F=SIN(X)
RETURN
```

В результате работы программы на экране получаем:

```
4.71238898038469      4.71238898038469      -1
```

Видно, что при заданной точности удается правильно определить 15 значащих чисел, а величина функции в минимуме находится абсолютно правильно.

Определение детерминанта матрицы

Для расчета детерминанта выбрана треугольная матрица третьего порядка, в которой элементы выше главной диагонали равны нулю. В данном случае, вместо нулей используются очень маленькие величины.

REM ----- Программа для вычисления детерминанта матрицы методом Гаусса с выбором главного элемента -----

```
CLS
DEFDBL A-Z
DIM A(3,3),AA(3,3)
```

```
N=3
```

```
AA(1,1)=1: AA(1,2)=1E-16: AA(1,3)=1E-16
AA(2,1)=1: AA(2,2)=1:   AA(2,3)=1E-16
AA(3,1)=1: AA(3,2)=1:   AA(3,3)=1
```

```
PRINT "ПРОГРАММА ДЛЯ ВЫЧИСЛЕНИЯ ДЕТЕРМИНАНТА  
МАТРИЦЫ МЕТОДОМ ГАУССА С ВЫБОРОМ ГЛАВНОГО ЭЛЕ-  
МЕНТА "
```

```
PRINT
```

```
PRINT "ИСХОДНЫЙ ВИД МАТРИЦЫ ДЛЯ ВЫЧИСЛЕНИЯ  
ДЕТЕРМИНАНТА "
```

```
PRINT
```

```
FOR I=1 TO N  
FOR J=1 TO N  
A(I-1,J-1)=AA(I,J)  
PRINT AA(I,J),  
NEXT J  
PRINT  
NEXT I
```

```
Z=1  
D=1  
FOR K=0 TO N-2  
E=1  
FOR I=K TO N-1  
FOR J=K TO N-1  
IF ABS(E)=ABS(A(I,J)) THEN 90  
E=A(I,J)  
B=I  
C=J  
NEXT J  
NEXT I
```

```
90 IF K=B THEN 120
```

```
FOR J=K TO N-1  
S=A(K,J)  
A(K,J)=A(B,J)  
A(B,J)=S  
NEXT J
```

```
Z=-Z  
120 IF K=C THEN 150  
FOR I=K TO N-1  
S=A(I,K)  
A(I,K)=A(I,C)  
A(I,C)=S  
NEXT I
```


Z=-Z

```
150 FOR I=K+1 TO N-1
G=A(I,K)/A(K,K)
```

```
FOR J=K TO N-1
A(I,J)=A(I,J)-G*A(K,J)
NEXT J
NEXT I
NEXT K
```

```
FOR I=0 TO N-1
D=D*A(I,I)
NEXT I
D=D*Z
PRINT
PRINT " ВЕЛИЧИНА ДЕТЕРМИНАНТА =";D
```

STOP

В результате работы программы будем иметь на экране следующие результаты:

ПРОГРАММА ДЛЯ ВЫЧИСЛЕНИЯ ДЕТЕРМИНАНТА МАТРИЦЫ
МЕТОДОМ ГАУССА С ВЫБОРОМ ГЛАВНОГО ЭЛЕМЕНТА

ИСХОДНЫЙ ВИД МАТРИЦЫ ДЛЯ ВЫЧИСЛЕНИЯ ДЕТЕРМИ-
НАНТА

1	1E-016	1E-016
1	1	1E-016
1	1	1

ВЕЛИЧИНА ДЕТЕРМИНАНТА= 0.9999999999999998

Точная величина детерминанта такой матрицы известна и равна 1. Видно, что найденное значение отличается от точной величины на $2.0 \cdot 10^{-16}$.

Решение системы линейных уравнений

Система линейных уравнений вида:


```
D(I)=B(I)
NEXT I

PRINT "ПЕЧАТЬ МАТРИЦЫ КОЭФФИЦИЕНТОВ УРАВНЕ-
НИЯ "A" И "B""
FOR I=1 TO N
FOR J=1 TO N
PRINT USING " +#.#####^ ^ ^ ^ "; A(I,J);
C(I,J)=A(I,J)
NEXT J
PRINT USING "          +#.#####^ ^ ^ ^ ";B(I)
NEXT I

FOR I=1 TO N-1
FOR J=I+1 TO N
A(J,I)=-A(J,I)/A(I,I)
FOR K=I+1 TO N
A(J,K)=A(J,K)+A(J,I)*A(I,K)
NEXT K
B(J)=B(J)+A(J,I)*B(I)
NEXT J
NEXT I
X(N)=B(N)/A(N,N)
FOR I=N-1 TO 1 STEP -1
H=B(I)
FOR J=I+1 TO N
H=H-X(J)*A(I,J)
NEXT J
X(I)=H/A(I,I)
NEXT I

PRINT " ВЫЧИСЛЕННЫЕ КОРНИ СИСТЕМЫ X(I)"

FOR I=1 TO N
PRINT "X("I")=";
PRINT USING " +#.#####^ ^ ^ ^ "; X(I)
NEXT I

PRINT "ПРОВЕРКА ПРАВИЛЬНОСТИ НАЙДЕННОГО РЕ-
ШЕНИЯ. ЕСЛИ ВЕЛИЧИНЫ B(I) БЛИЗКИ К НУЛЮ - РЕШЕНИЕ
ПРАВИЛЬНОЕ "
```

```
FOR I=1 TO N
S=0
```

```
FOR J=1 TO N
S=S+C(I,J)*X(J)
NEXT J
Y(I)=S-D(I)
PRINT USING " +#.#####^ ^ ^ ";Y(I)
NEXT I
```

END

В результате решения системы на экране получим:

ВВЕДИТЕ ЧИСЛО УРАВНЕНИЙ N? 2

ВВОД КОЭФФИЦИЕНТОВ ЛЕВОЙ ЧАСТИ УРАВНЕНИЯ
A(I,J)

```
A( 1 1 )=? 1
A( 1 2 )=? 2
A( 2 1 )=? 3
A( 2 2 )=? 4
```

ВВОД КОЭФФИЦИЕНТОВ ПРАВОЙ ЧАСТИ УРАВНЕНИЯ
B(I)

```
B( 1 )=? 5
B( 2 )=? 6
```

ПЕЧАТЬ МАТРИЦЫ КОЭФФИЦИЕНТОВ УРАВНЕНИЯ "A" И
"B"

```
+1.00000E+00 +2.00000E+00 +5.00000E+00
+3.00000E+00 +4.00000E+00 +6.00000E+00
```

ВЫЧИСЛЕННЫЕ КОРНИ СИСТЕМЫ X(I)

```
X( 1 )= -4.00000E+00
X( 2 )= +4.50000E+00
```

ПРОВЕРКА ПРАВИЛЬНОСТИ НАЙДЕННОГО РЕШЕНИЯ.
ЕСЛИ ВЕЛИЧИНЫ Y(I) БЛИЗКИ К НУЛЮ - РЕШЕНИЕ ПРА-
ВИЛЬНОЕ

```
+0.00000E+00
+0.00000E+00
```

Из результатов видно, что для такой системы получено абсолютно точное решение, поскольку элементы $Y(I)$ строго равны нулю.

Приведем еще одно решение для другой системы 3 - го порядка:

ПЕЧАТЬ МАТРИЦЫ КОЭФФИЦИЕНТОВ УРАВНЕНИЯ "А" И "В"

+9.00000E+00	+7.00000E+00	+5.00000E+00	+5.00000E+00
+3.00000E+00	+1.00000E+00	+4.00000E+00	+3.00000E+00
+2.00000E+00	+6.00000E+00	+8.00000E+00	+7.00000E+00

ВЫЧИСЛЕННЫЕ КОРНИ СИСТЕМЫ

$X(1) = -1.70455E-02$
 $X(2) = +2.32955E-01$
 $X(3) = +7.04545E-01$

ПРОВЕРКА ПРАВИЛЬНОСТИ НАЙДЕННОГО РЕШЕНИЯ.
ЕСЛИ ВЕЛИЧИНЫ $Y(I)$ БЛИЗКИ К НУЛЮ - РЕШЕНИЕ ПРАВИЛЬНОЕ

+0.00000E+00
+0.00000E+00
-8.88178E-16

В данном случае один из элементов $Y(I)$ точно не равен нулю, но его величина настолько мала, что решение можно считать правильным.

Обращение матрицы

Обратной $O(I,J)$ для заданной матрицы $A(I,J)$ будет матрица удовлетворяющая условию:

$$AO = E$$

или

$$AA^{-1} = E$$

где E - единичная матрица, у которой все элементы, кроме диагональных равны нулю, а в диагонали стоят единицы.

Обратная матрица используется для решения системы линейных уравнений:

X=OB

где X - решение системы.

Ниже приведена программа нахождения обратной матрицы:

REM ----- Программа обращения матрицы -----

DEFDBL A-Z

REM DEFINT I,J,K,L,N,M

DEFDBL A-Z

CLS

N=4

DIM A(N,N),P(N,N),B(N),C(N,N),G(N),X(N)

A(1,1)=2: A(1,2)=8: A(1,3)=1: A(1,4)=11

A(2,1)=1: A(2,2)=3: A(2,3)=6: A(2,4)=2

A(3,1)=1: A(3,2)=2: A(3,3)=9: A(3,4)=3

A(4,1)=3: A(4,2)=4: A(4,3)=1: A(4,4)=7

PRINT " ИСХОДНАЯ МАТРИЦА A(I,J)"

FOR I=1 TO N

FOR J=1 TO N

PRINT USING " +#.#####^ ^ ^ ^ ";A(I,J);

NEXT J

PRINT

NEXT I

PRINT

FOR I=1 TO N

FOR J=1 TO N

P(I,J)=A(I,J)

NEXT J

NEXT I

FOR J2=1 TO N

FOR I=1 TO N

B(I)=0

NEXT I

B(J2)=1

```
FOR J3=1 TO N
FOR J4=1 TO N
A(J3,J4)=P(J3,J4)
NEXT J4
NEXT J3
```

```
GOSUB 100
```

```
FOR II=1 TO N
O(II,J2)=X(II)
REM PRINT X(II)
NEXT II
```

```
NEXT J2
```

```
SS=0
FOR I=1 TO N
FOR J=1 TO N
S=0
FOR K=1 TO N
S=S+P(I,K)*O(K,J)
NEXT K
E(I,J)=S
SS=SS+S
NEXT J
NEXT I
```

```
PRINT " ОБРАТНАЯ МАТРИЦА O(I,J)"
FOR I=1 TO N
FOR J=1 TO N
PRINT USING " +#.#####^ ^ ^ ^ ";O(I,J);
NEXT J
PRINT
NEXT I
PRINT
```

```
PRINT " СУММА ЭЛЕМЕНТОВ ЕДЕНИЧНОЙ МАТРИЦЫ S="
PRINT SS
PRINT " ЕДЕНИЧНАЯ МАТРИЦА E(I,J)"
FOR I=1 TO N
FOR J=1 TO N
PRINT USING " +#.#####^ ^ ^ ^ ";E(I,J);
NEXT J
PRINT
```

```
NEXT I

END

100 N1=N-1
FOR K=1 TO N1
IF ABS(A(K,K))>0 GOTO 200
K1=K+1
FOR M=K1 TO N
IF ABS(A(M,K))>0 GOTO 150
GOTO 165
150 FOR L=1 TO N
V=A(K,L)
A(K,L)=A(M,L)
A(M,L)=V
NEXT L
165 NEXT M
V=B(K)
B(K)=B(M)
B(M)=V
200 G(K)=B(K)/A(K,K)
K1=K+1
FOR I=K1 TO N
B(I)=B(I)-A(I,K)*G(K)
FOR J1=K TO N
J=N-J1+K
C(K,J)=A(K,J)/A(K,K)
A(I,J)=A(I,J)-A(I,K)*C(K,J)
NEXT J1
NEXT I
NEXT K
M=N
X(M)=B(M)/A(M,M)
250 M=M-1
S=0
FOR L=M TO N1
S=S+C(M,L+1)*X(L+1)
NEXT L
X(M)=G(M)-S
IF M>1 GOTO 250
RETURN
```

В результате работы программы для поиска обратной матрицы на экране получаем:

ИСХОДНАЯ МАТРИЦА A(I,J)

```
+2.0000E+00 +8.0000E+00 +1.0000E+00 +1.1000E+01
+1.0000E+00 +3.0000E+00 +6.0000E+00 +2.0000E+00
+1.0000E+00 +2.0000E+00 +9.0000E+00 +3.0000E+00
+3.0000E+00 +4.0000E+00 +1.0000E+00 +7.0000E+00
```

ОБРАТНАЯ МАТРИЦА O(I,J)

```
-3.2468E-01 +3.3766E-01 -2.4675E-01 +5.1948E-01
+6.4935E-02 +5.3247E-01 -3.5065E-01 -1.0390E-01
-1.2987E-02 -6.4935E-03 +1.2013E-01 -2.9221E-02
+1.0390E-01 -4.4805E-01 +2.8896E-01 -1.6234E-02
```

СУММА ЭЛЕМЕНТОВ ЕДИНИЧНОЙ МАТРИЦЫ

S= 3.999999999999999

ЕДИНИЧНАЯ МАТРИЦА E(I,J)

```
+1.0000E+00 -1.6653E-16 -1.1102E-16 -3.4694E-17
+0.0000E+00 1.0000E+00 +0.0000E+00 -3.4694E-17
+2.7756E-17 -1.6653E-16 +1.0000E+00 +6.9389E-18
+2.4980E-16 +1.8319E-15 -2.3315E-15 1.0000E+00
```

Видно, что все элементы единичной матрицы, кроме диагональных практически равны нулю, а по диагонали стоят единицы, как и должно быть при правильном нахождении обратной матрицы. Величина S дает сумму всех элементов единичной матрицы, которая должна быть равна 4 при N=4.

Вычисление Гамма функции

Гамма функция, часто используемая в различных математических вычислениях при Z от 0 до 1 может быть представлена в виде степенного ряда:

$$\Gamma(1 + Z) = \left[\sum_{k=1}^N A_k Z^k \right]^{-1}$$

где a_k - заданные коэффициенты, величина которых известна до N=20 и приведена в самой программе:

REM ----- Программа вычисления $\Gamma(Z+1)$ функции -----

```
DEFDBL A-Z  
DIM A(20)  
CLS
```

Z=0.46163 - аргумент функции

```
A(0)=1  
A(1)=0.57721566490153286: A(2)=-0.65587807152025388  
A(3)=-0.04200263503409523: A(4)=0.1665861138229148  
A(5)=-0.04219773455554433: A(6)=-0.00962197152787697  
A(7)=0.00721894324666309: A(8)=-0.00116516759185906  
A(9)=-0.00021524167411495: A(10)=0.00012805028238811  
A(11)=-0.00002013485478078: A(12)=-0.00000125049348214  
A(13)=0.0000013302723198: A(14)=-0.00000020563384169  
A(15)=0.00000000611609510: A(16)=0.00000000500200764  
A(17)=-0.00000000118127457: A(18)=0.0000000001044267  
A(19)=0.00000000000778226: A(20)=-0.0000000000069681
```

```
S=0  
FOR I=0 TO 20  
S=S+A(I)*Z^I  
NEXT I  
GA=1/S  
PRINT "Г=";GA
```

STOP

Результатом вычисления, при заданном Z=0.46163 будет величина:

Г= 0.8856015025262314

полностью совпадающая с известными табличными данными - 0.88560.

СИСТЕМА WINDOWS - 98

Приступая к изучению операционной системы Microsoft Windows - 98, отметим вначале ее общие возможности и отличительные черты по сравнению с Windows - 95. Новая версия операционной системы Windows обладает повышенной производительностью, надежностью и улучшенным оформлением **Рабочего стола**. Из самой операционной системы и любой прикладной программы открыт непосредственный доступ к Интернету. Новые возможности **Рабочего стола**, **Панели задач** и **Главного меню** помогают быстро открывать или создавать **Ярлыки** и запускать программы.

Перемещаться в программах, файлах и папках стало заметно проще благодаря новым свойствам **Рабочего стола**. Появилась возможность открытия файлов одним щелчком, выделение значков наведением курсора мыши, новые кнопки **Вперед** и **Назад**, удобное для настройки меню **Пуск** и наличие дополнительных кнопок управления просмотром в любом окне. Стало возможным использование нескольких мониторов с одним компьютером. Поддержка в Windows - 98 Универсальной Последовательной Шины упрощает установку нового оборудования, позволяя подключать и немедленно использовать его в работе без перезагрузки компьютера.

Запуск операционной системы Windows и приложений происходит быстрее, чем раньше. Мастер обслуживания повышает быстроту действия и эффективность работы компьютера. Средства управления электропитанием дают возможность использовать на современных компьютерах "спящий" режим с последующим быстрым включением, что устраняет необходимость отключения и повторной загрузки компьютера. Использование файловой системы FAT32 обеспечивает более эффективное размещение файлов и экономию места на жестком диске. Работа с Интернетом стала заметно проще, а Мастер подключения упрощает саму процедуру подключения к Интернету через выбранного поставщика услуг - провайдера.

Windows - 98 поддерживает работу с цифровыми видеодисками и цифровыми аудиозаписями, что дает возможность воспроизводить на компьютере высококачественные видео - и аудиозаписи. Графическое и звуковое оформление **Рабочего стола** может быть изменено с помощью различных тем - способов оформления экрана. В Windows - 98 включено несколько десятков самых различных цветовых тем. Каждая тема **Рабочего стола** включает собственный фоновый рисунок, экранные заставки, хранители экрана, объемные значки, звуки, шрифты, цветовые схемы и множество указателей курсора мыши.

При работе на современном компьютере средство On Now помогает уменьшить энергопотребление, предотвратить износ жесткого диска и уменьшить шум от его работы. Если в течение некоторого

периода времени на компьютере не производится никаких действий, программа On Now переводит его в режим отключения с возможностью быстрого запуска.

Выбор режима "Всегда Вкл" в окне управления электропитанием сокращает время запуска компьютера. В этом режиме для запуска компьютера достаточно нескольких секунд. При этом все программы восстанавливаются в том состоянии, которое они имели на момент отключения. Кроме того, этот режим позволяет компьютеру работать даже тогда, когда он кажется выключенным. Пользователь получает возможность оставить все программы выполняющимися, загружать нужные **Web** - страницы, отправлять и получать электронную почту, архивировать жесткий диск или выполнять настройку операционной системы без необходимости находиться у компьютера.

Кроме того, средства управления питанием позволяют перевести компьютер в режим ожидания (ждущий режим) для сохранения ресурсов питания. С помощью такого режима можно сократить потребление электроэнергии некоторыми устройствами компьютера или системой в целом. Для этого нужно выбрать схему управления питанием, которая представляет собой набор параметров, определяющих использование электроэнергии компьютером. Можно самому настроить отдельные параметры схемы управления питанием. Например, в зависимости от установленного на компьютере оборудования, имеются следующие возможности:

1. Перевод компьютера при его простое в ждущий режим работы. В ждущем режиме отключаются монитор и жесткие диски, и компьютер потребляет меньше электроэнергии. При возобновлении работы компьютер быстро выходит из ждущего режима и **Рабочий стол** системы Windows полностью восстанавливается.

2. Перевод компьютера в спящий режим работы. В спящем режиме отключаются монитор и жесткие диски. Все содержимое оперативной памяти сохраняется на жестком диске и компьютер выключается. При перезапуске компьютера состояние **Рабочего стола** полностью восстанавливается. Выход из спящего режима занимает больше времени, чем выход из ждущего режима.

В Windows - 98 включены сотни новых драйверов для принтеров, модемов и другого оборудования, что как никогда повысило эффективность процедуры установки и настройки нового оборудования.

При возникновении вопросов или неполадок можно быстро найти ответы и получить помощь, так как в Windows - 98 содержатся средства устранения неполадок. Кроме справки Windows, документов со сведениями о системе, в Windows содержится несколько дополнительных средств, которые могут оказать помощь:

1. Средства устранения неполадок предназначены для диагностики и устранения неполадок на компьютере. Средства устранения неполадок вызываются непосредственно из справки Windows, а также содержатся на **Web** - узле электронной поддержки Windows.

2. Программа резервного копирования поддерживает SCSI - накопители на магнитной ленте и позволяет легко, и быстро произвести архивацию данных.

3. Обслуживание системы производится с помощью мастера обслуживания. С его помощью можно проверить жесткий диск на наличие ошибок, удалить ненужные файлы и повысить скорость работы избранных приложений. Возможна настройка расписания автоматического выполнения процедур по обслуживанию компьютера.

4. Программа "Доктор Ватсон" служит для диагностики компьютера и создания отчетов о программных неполадках. При появлении неполадок в работе некоторого приложения программа "Доктор Ватсон" автоматически записывает события в журнал, который впоследствии всегда можно просмотреть.

5. Появилась возможность автоматизации выполнения задач по обслуживанию, например, проверки жесткого диска на наличие ошибок или проверки полученной электронной почты при запуске Windows или по расписанию. После настройки планировщика он запускается в фоновом режиме при каждом запуске Windows - 98.

Все эти возможности делают данную систему Windows самой настраиваемой и надежной из всех ранее выпущенных фирмой Microsoft программ подобного типа.

ЗАПУСК WINDOWS

При включении компьютера Windows запускается сразу после проверки всех систем и его загрузки в оперативную память. Этот процесс на разных компьютерах может занимать время от 1 до 2 - 3 минут, в течение которых на экране находится заставка Windows - 98. Временами заставка может исчезать, экран становится черным и появляется белая надпись "Идет загрузка Windows...".

Возможен и другой, нередко используемый вариант загрузки системы Windows. В этом случае, после включения компьютера, на экране появляется программа Norton Commander, описание которой было дано ранее. Тогда нужно нажать клавишу F10 и, если появится окно подтверждения выхода из Norton Commander щелкнуть мышкой на экране кнопку **Да** или нажать клавишу Enter.

При каждом запуске Windows на экране появляется диалоговое окно входа в Windows, которое позволяет производить вашу идентификацию, т.е. вводить свои Имя и Пароль, если они были предвари-

тельно заданы. Кроме целей обеспечения безопасности данная процедура позволяет Windows хранить личные настройки каждого пользователя, такие как оформление **Рабочего стола**, недавно использованные документы, избранные файлы и многое другое.

Если компьютер подключен к локальной сети, также может потребоваться ввод Имени и Пароля для входа и работы в сети. Данное диалоговое окно появляется сразу после предыдущего и в целом имеет такой же внешний вид.

Вход в Windows

После запуска системы Windows последовательно выполните следующие действия:

1. В появившемся на экране диалоговом окне введите Имя пользователя и Пароль, а затем щелкните мышкой кнопку **ОК** или нажмите клавишу Enter.
2. Если Пароль задается впервые, то в окне подтверждения пароля Windows повторно введите ваш Пароль и снова щелкните кнопку **ОК**. Введенный Пароль отображается на экране в виде звездочек (*).
3. Если Пароль не был предварительно задан при установке системы, то поля Имени и Пароля можно оставлять пустыми, просто щелкнув кнопку **ОК**.

После этого на экране компьютера появляется **Рабочий стол** Windows - область экрана, непосредственно используемая вами для работы. **Рабочий стол** следует считать своим основным рабочим местом на компьютере. В левой части **Рабочего стола** расположено несколько значков - **Ярлыков**. Каждый значок обозначает собой некоторый объект - **Папку** или **Программу**.

Знакомство с компьютером

Состав компьютера и его содержимое можно посмотреть самыми разными способами. Это можно сделать либо с помощью значка **Мой компьютер**, либо программой **Проводник** Windows. Оба средства легко доступны - значок **Мой компьютер** находится в левом верхнем углу **Рабочего стола**. **Проводник** запускается с помощью кнопки **Пуск**, которая находится на **Панели задач** (серая полоса внизу экрана компьютера) и открывает **Главное меню** системы.

В таблице ниже приведено сравнение различных способов перемещения по компьютеру.

<i>Способ</i>	<i>Расположение</i>	<i>Вывод локальных дисков и сети</i>	<i>Основное назначение</i>
Кнопка Пуск	Панель задач	Нет	Запуск программ, открытие папки Избранное , получение справки, выключение системы
Мой компьютер	Рабочий стол	Да (если подключен сетевой диск)	Работа с локальными дисками, системными средствами и ресурсами
Проводник Windows	Кнопка Пуск	Да	Отображение иерархии всего содержимого и ресурсов компьютера в одном окне
Сетевое окружение	Рабочий стол	Да (только сеть)	Просмотр компьютеров в сети

Панель задач и Главное меню

Панель задач и кнопка **Пуск** всегда доступны на **Рабочем столе**, сколько бы окон не было открыто на экране. Кнопки на **Панели задач** показывают, какие окна в данный момент открыты, даже если некоторые из них свернуты или находятся за другими окнами. Для перехода в нужное окно достаточно щелкнуть левой кнопкой мышки соответствующую кнопку на **Панели задач**.

С помощью **Главного меню** кнопки **Пуск** можно выполнить практически любую задачу. Оно позволяет запускать программы, открывать документы, настраивать систему, получать справочные сведения, осуществлять поиск по компьютеру и многое другое. Некоторые пункты **Главного меню** содержат стрелку вправо, что означает наличие еще одного дополнительного меню или подменю. Если поместить указатель - курсор мыши на пункт меню со стрелкой, то на экране появится новое меню.

Ваш компьютер может быть настроен несколько иначе, но обычно меню кнопки **Пуск** выглядит примерно так, как показано на рис.41. Для работы с **Главным меню**, т.е. запуска прикладных программ и изменения установок - опций системы выполните следующие действия:

1. Щелкните левой кнопкой мыши кнопку **Пуск**. На экране появится **Главное меню**.
2. Выберите элемент или раздел, который требуется открыть, установите на него курсор мыши и снова щелкните левой кнопкой. Если выбранный раздел меню открывается в дополнительное подме-

ню, укажите на нужный пункт этого подменю и щелкните левой кнопкой.

При работе с пунктами любого меню всегда используется одинарный щелчок левой кнопкой мышки. Для открытия окна **Ярлыка** программы или папки на **Рабочем столе** может применяться, как одинарный, так и двойной щелчок мышкой.

Папка Мой компьютер

Значок **Мой компьютер** удобен для просмотра содержимого одной определенной папки или диска, а так же для изменения установок, опций работы самой системы Windows. По двойному щелчку мышкой на значке **Мой компьютер**, в появившемся на экране окне показаны доступные на компьютере диски. Теперь, если дважды (или один раз) щелкнуть значок диска, в окне будут приведены имеющиеся на этом диске папки. Для просмотра содержимого любой папки следует дважды щелкнуть ее значок в открытом на экране окне.

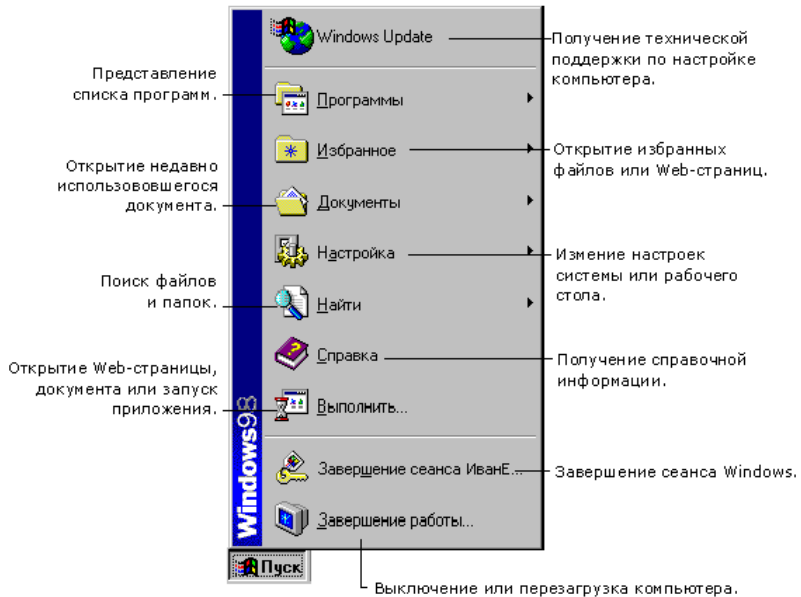


Рис.41. Меню кнопки **Пуск**.

В окне папки **Мой компьютер** обычно находятся следующие объекты:



- Просмотр содержимого жесткого диска, обычно обозначенного буквой "C:".



- Просмотр содержимого компакт - диска в устройстве для чтения компакт - дисков CD - ROM, обычно обозначаемого английской буквой "D:".



- Просмотр средств изменения параметров - опций работы системы и компьютера.



- Настройка принтера, просмотр сведений об имеющихся на компьютере принтерах и состоянии задач печати.



- Планирование и контроль работ по программному и техническому обслуживанию компьютера.



- Настройка удаленного доступа, т.е. работы компьютера в Интернет и некоторых типах локальных сетей.

Программа Проводник

Если вам удобно просматривать файлы в виде иерархической структуры, то вам будет очень полезен **Проводник** Windows. Он может быть запущен из раздела **Программы Главного меню** или при щелчке правой кнопкой мыши по кнопке **Пуск**. Левая область **Проводника** Windows содержит список дисков и каталогов, а правая отображает содержимое текущего, выделенного слева объекта. Меню **Вид** программы позволяет изменить представление значков в правой области.

Для перехода на один уровень вверх по структуре папок щелкните кнопку с папкой, на которой показана стрелка вверх на **Панели инструментов** программы или нажмите клавишу Backspace на клавиатуре. Если **Панель инструментов** не выводится на экран, выберите в меню **Вид** программы **Проводник** команду **Панели инструментов**, а затем щелкните раздел **Обычные кнопки**. В любом окне Windows включаются (или выключаются) следующие элементы **Панели инструментов**:

1. **Обычные кнопки** - включение панели кнопок, которые выполняют все основные действия при работе с файлами и папками.
2. **Подписи к кнопкам** - добавление к кнопкам поясняющих надписей. Эти подписи можно не включать, поскольку при наведении курсора мышки на кнопку на экране появляется ее подпись.

3. **Адресная строка** - определяет путь к объекту, который вы просматриваете в активном, находящемся на экране окне.

Работа с мышкой

Большинство выполняемых на компьютере действий осуществляется путем указания курсором мышки объекта на экране и нажатием какой - либо ее кнопки. Для того, чтобы указать на объект, переместите мышь так, чтобы кончик указателя мыши закрывал нужный объект на экране и выполните одно из приведенных действий:

1. **Щелчок** - один раз быстро нажмите и отпустите левую кнопку мыши.

2. **Двойной щелчок** - дважды нажмите и отпустите левую кнопку мыши.

3. **Щелчок правой кнопкой** - один раз нажмите и отпустите правую кнопку мыши. После этого на экране появится Контекстное меню, вид которого существенно зависит от места, в котором был выполнен щелчок. Для того, чтобы выбрать некоторый пункт Контекстного меню поставьте на него курсор мышки и щелкните ее левой кнопкой.

Перемещение объектов на экране весьма сходно с перемещением предметов на столе. Например, перетаскивание значка по рабочему столу похоже на действия, которые вы выполняете, когда берете со стола настоящий предмет и перекладываете его на другое место. Точно так же, для переноса объекта на экране сначала следует поместить указатель мыши на этот объект. Затем вы "берете" объект, нажимая и удерживая левую кнопку мыши. Далее, перемещайте указатель мыши на то место, куда вы хотите перенести объект, а затем отпустите кнопку мыши. Такое перемещение возможно не только на **Рабочем столе**, но и между разными открытыми на экране окнами, разных папок или программ. Подобные действия используются и для выделения текста в документе или выделения имени файла. Для выделения текста необходимо сначала поставить курсор мыши в то место, откуда вы хотите начать выделение. Затем нужно нажать левую кнопку и, удерживая ее, перемещать указатель мыши на то место, где должно кончиться выделение текста, а затем отпустить кнопку мыши. Для выделения файла, **Ярлыка** или папки на **Рабочем столе** или в другой папке достаточно просто щелкнуть по его имени. В **Web** - стиле работы экрана для выделения объекта достаточно указать на него курсором мыши, который в этом случае принимает на экране вид "руки с указательным пальцем".

Работа с окнами

После открытия любого окна на экране компьютера имеется возможность управлять его размерами и положением. Справа сверху окна находятся три кнопки, позволяющие управлять размерами окна:

1. Средней кнопкой с квадратиком можно увеличить - **Развернуть** окно на весь экран, чтобы видеть большую часть его содержимого. После разворачивания окна рисунок на этой кнопке изменится и превратится в двойной квадратик, и теперь, щелкнув по нему, можно **Восстановить** - вернуть исходный размер окна на экране компьютера.

2. Кнопкой с минусом - **Свернуть** можно временно выйти из окна и поработать в другом приложении Windows, уменьшив его размер до значка - кнопки на **Панели задач**. Окно исчезнет с экрана, но его можно быстро открыть - развернуть на экране, щелкнув по этой кнопке.

3. Для закрытия текущего активного окна, которое находится на первом плане экрана надо щелкнуть по кнопке с крестиком в его правом верхнем углу (кнопка **Закреть**) или в разделе Главного меню **Файл** программы выбрать команду **Закреть**.

Для произвольного изменения размера окна, укажите курсором мышки на его границу, и когда указатель мыши примет вид двуправленной стрелки, переместите границу мышью, передвигая ее и удерживая в нажатом состоянии левую кнопку до получения нужного размера окна. Для перемещения окна по экрану установите курсор мыши на его заголовок (синяя полоса сверху окна с названием и кнопками управления размерами в правой его части) и, удерживая нажатой левую кнопку, перемещайте мышку, перенося тем самым окно на новое место.

РАБОЧИЙ СТОЛ, ПАНЕЛЬ ЗАДАЧ И ГЛАВНОЕ МЕНЮ

Рабочий стол это рабочая область экрана, появляющаяся после запуска системы Windows. **Рабочий стол** и **Панель задач** - серая полоса внизу экрана с кнопкой **Пуск**, используются при решении практически всех вопросов и любых задач на компьютере:

1. Запуска прикладных программ, например, программ комплекса MS Office.
2. Копирования и переноса файлов и папок в одном окне или между разными окнами различных программ.
3. Подключения к Интернету и локальным компьютерным сетям.
4. Чтения и отправки электронной почты, работы с группами новостей и т. д.



Рис.42. Общий вид **Рабочего стола**.

Внешний вид **Рабочего стола** зависит от установок и настроек Windows на данном компьютере. Основные элементы **Рабочего стола** Windows показаны на рис.42 (показаны так же некоторые допол-

нительные значки - **Ярлыки**, которых нет при стандартном виде **Рабочего стола**).

Общий вид Рабочего стола

В начале работы с Windows имеется возможность настроить параметры **Рабочего стола** в соответствии с индивидуальными требованиями и предпочтениями. "По умолчанию" файлы и папки на **Рабочем столе** или в окне **Проводника** открываются двойным щелчком левой кнопкой мышки. Если вам больше нравится перемещение по компьютеру в стиле **Web**, можно настроить систему на одиночный щелчок.

Для изменения параметров обзора следует изменить стиль **Рабочего стола**. В режиме Активного **Рабочего стола** в качестве фонового рисунка можно использовать **Web** - страницу, документ или Интернетовский канал. Режим Активного **Рабочего стола** доступен при использовании **Рабочего стола**, как в стиле **Web**, так и в классическом стиле.

Windows - 98 предоставляет неограниченные возможности для интеграции компьютера с Интернетом. Даже, если вы не имеете подключения к Интернету или предпочитаете классическое оформление Windows, вы все равно можете воспользоваться преимуществами стиля **Web**. Например, теперь для просмотра и перемещения по **Web** или жесткому диску в любом окне достаточно одиночного щелчка.

Никогда ранее не было доступно столько вариантов настроек системы Windows. В том числе, можно создать собственную **Панель инструментов**, с помощью которой легко открывать избранные приложения или программы и добавлять избранные ссылки. Например, просмотр содержимого жесткого диска, и поиск в Интернете будет доступен одновременно в одном и том же окне.

Настройки стиля Рабочего стола и папок

Windows - 98 предоставляет несколько способов отображения **Рабочего стола** и просмотра файлов и папок: стиль **Web**, классический стиль Windows - 95 или их сочетание. Выбранный вариант оформления определяет способ просмотра объектов в Windows - 98, независимо от того, откуда был начат просмотр - с **Рабочего стола**, из окна **Мой компьютер**, из **Проводника** Windows, из **Главного меню** кнопки **Пуск** или из программы Internet Explorer:

1. Стиль **Web** - позволяет просматривать **Рабочий стол** и папки, как **Web** - страницы, щелкая элементы просмотра для их откры-

тия только один раз. При этом файлы, папки и приложения - программы открываются в одном окне, а не в отдельных окнах системы.

2. Классический стиль - используется для оформления **Рабочего стола** в виде классического **Рабочего стола** варианта системы Windows - 95. Элементы открываются на экране двойным щелчком мышки, причем каждый в новом окне.

3. Настраиваемый стиль - использует параметры, выбранные пользователем. К их числу относятся параметры просмотра папок, создания фоновых рисунков для окон, а также режим выделения и открытия элементов с помощью мыши. Например, сочетая параметры стиля **Web** и классического стиля, можно открывать элементы двойным щелчком мышки (классический стиль) и использовать **Web** - страницы в качестве фоновых рисунков для окон (**Web** - стиль). Или использовать классический стиль, но установить режим одиночного щелчка.

Работа в стиле Web

Стиль **Web** предназначен для тех, кто предпочитает организацию и просмотр содержимого компьютера с применением следующих параметров, напоминающих работу в Интернете:

1. Использование одиночного щелчка для просмотра **Рабочего стола** и папок. В этом случае для открытия элемента достаточно только одного щелчка, как для перехода по ссылке в **Web**. Для выделения элемента достаточно просто поместить на него указатель мыши, который превращается при этом в "руку".

2. Отображение **Web** - страниц в качестве фоновых рисунков отдельных окон и **Рабочего стола**. Для этого можно использовать **Web** - страницы, любые страницы и документы Интернетовского формата HTML. При использовании стиля **Web** любая папка с содержимым в формате HTML отображается в виде **Web** - страницы, как показано на рис.43. Страница HTML может включать в себя графику, документы Microsoft Office, мультимедиа, интерактивные формы и т.д.

Некоторые окна, такие как **Мой компьютер** и **Панель управления** имеют возможность выводить описание каждого элемента - **Ярлыка** при помещении указателя мыши на его значок (рис.43).

Для настройки **Web** - стиля **Рабочего стола** выполните следующие действия:

1. Один раз щелкните в любом свободном месте экрана правой кнопкой мыши.

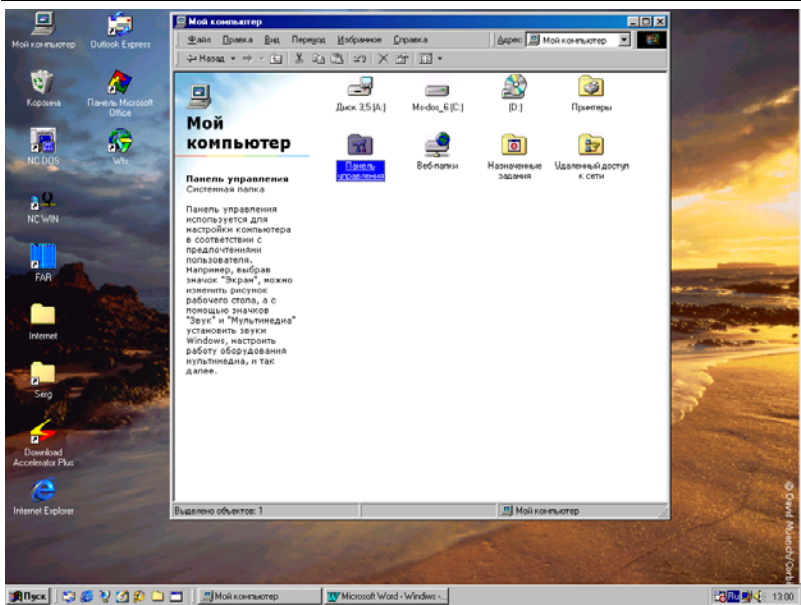


Рис.43. Окно **Мой компьютер** в стиле **Web**.

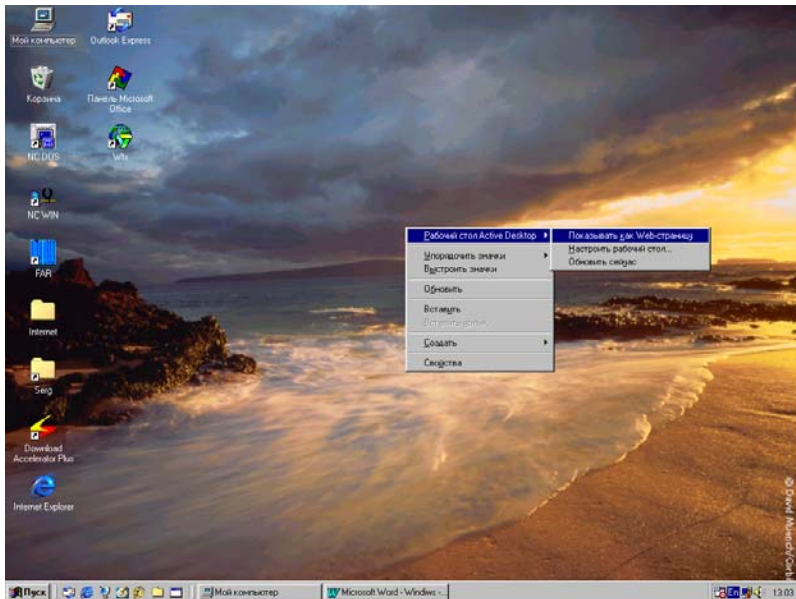


Рис.44. Меню установки стиля **Web** для **Рабочего стола**.

2. В появившемся Контекстном - Ниспадающем меню, вид которого показан на рис.44 выберите пункт **Рабочий стол...** , а в дополнительном меню щелкните раздел **Показывать как Web - страницу**, установив слева от него "птичку" (флажок).

В этом случае на экране появляется **Панель каналов** и включается режим одиночного щелчка мышкой по выбранным объектам. Существует и другой способ включения режима **Web - страницы** для **Рабочего стола**:

1. Щелкните правой кнопкой мышки в свободном месте экрана компьютера.
2. В появившемся на экране Контекстном меню выберите пункт **Свойства**.
3. В открывшемся окне **Свойства: Экран** на вкладке **Интернет** (Web) щелкните мышкой команду **Отображать Рабочий стол, как Web - страницу**.
4. Включите, если нужно режим отображения **Панели каналов** на экране компьютера - **Рабочем столе** и щелкните кнопку **ОК**, как показано на рис.45.

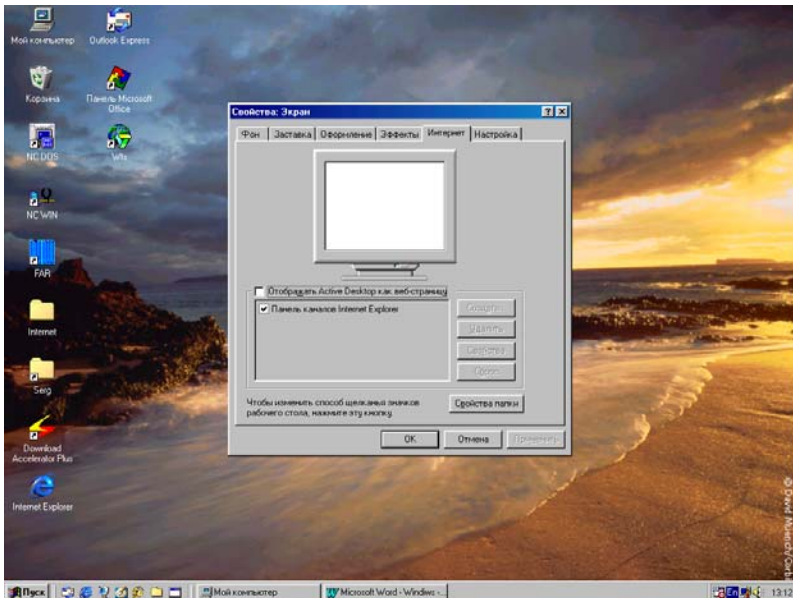


Рис.45. Установка режима **Рабочего стола** в форме **Web**.

И, наконец, в третьем способе включения **Web** - стиля нужно открыть окно **Свойства папки**. Это можно сделать, щелкнув по кнопке с таким же названием в окне на рис.45 или:

1. Откройте на **Рабочем столе** любую папку, например, **Мой компьютер**.
2. Выберите пункт **Свойства папки** в меню **Вид**. Откроется диалоговое окно **Свойства папки** (рис.46).

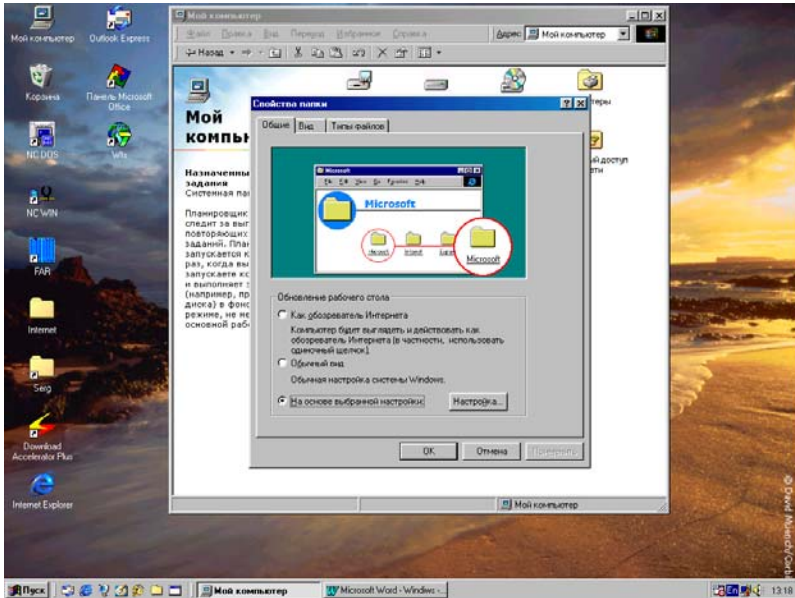


Рис.46. Окно установки **Web** - стиля для папки.

3. Выберите параметр **Как обозреватель Интернета** на вкладке **Общие** и нажмите кнопку **ОК**.

Выполненные таким образом действия, приведут к установке общего вида любой папки в форме Интернетовской **Web** - страницы, как показано на рис.43.

Работа в классическом стиле

Классический стиль напоминает оформление **Рабочего стола** Windows - 95. При его использовании открытие элементов выполня-

ется двойным щелчком, выделение объектов происходит одним щелчком и каждый элемент открывается в отдельном окне.

Для включения классического стиля системы выполните следующие действия:

1. На **Рабочем столе** откройте любую папку, например, значок **Мой компьютер**.
2. Выберите пункт **Свойства папки** в меню **Вид** открытого на экране окна.
3. Появится диалоговое окно **Свойства папки**, показанное выше на рис.46.
4. Это окно можно вывести на экран и способом, описанным в предыдущем пункте через Контекстное меню, раздел **Свойства** (рис.45).
5. Выберите параметр **Обычный вид** и щелкните левой кнопкой мышки кнопку **ОК**.

Выбор пользовательских настроек

Для того, чтобы использовать сочетание стиля **Web** и классического стиля Windows - 95, можно выполнить настройки по собственному выбору. Например, легко настроить **Рабочий стол** на просмотр с использованием двойного щелчка (классический стиль) и отображение Активного **Рабочего стола** (стиль **Web**).

Для получения таких настроек **Рабочего стола** выполните последовательность действий:

1. Откройте любую папку, например, папку **Мой компьютер** на **Рабочем столе**.
2. Щелкните мышкой пункт **Свойства папки** в меню **Вид** открытого окна.
3. Откроется диалоговое окно **Свойства папки**, которое можно вызвать и через Контекстное меню по правой кнопке мыши, и через меню кнопки **Пуск**, раздел **Настройки**, пункт **Свойства папки**.
4. Выберите параметр **На основе выбранных настроек** и нажмите кнопку **Настройка** (рис.47).
5. На экране откроется новое диалоговое окно **Пользовательские настройки**.
6. Установите нужные параметры, щелкая мышкой по выбранным пунктам и нажмите кнопку **ОК**.

В частности, режим одиночного щелчка мыши позволяет работать на **Рабочем столе** и в папках так же, как в обозревателе Интер-

Панели управления) и в появившемся Контекстном меню выберите пункт **Свойства**, вкладку **Фон**.

3. Откроется диалоговое окно **Свойства: Экран**, как показано на рис.49. Для того, чтобы установить фон, выберите рисунок в меню **Выберите документ HTML или рисунок** или нажмите кнопку **Обзор** и, в появившемся окне выберите нужный файл рисунка.

4. Для того, чтобы рисунок точно вписывался в размер экрана при любом его разрешении, в меню **Поместить** выберите режим **Растянуть** и щелкните **ОК**.

5. На вашем **Рабочем столе** появится выбранный рисунок фона (либо в виде мозаики, либо в растянутом на экране виде, либо в центре экрана, что определяется текущей установкой, выбранной вами в меню со списком **Поместить**).

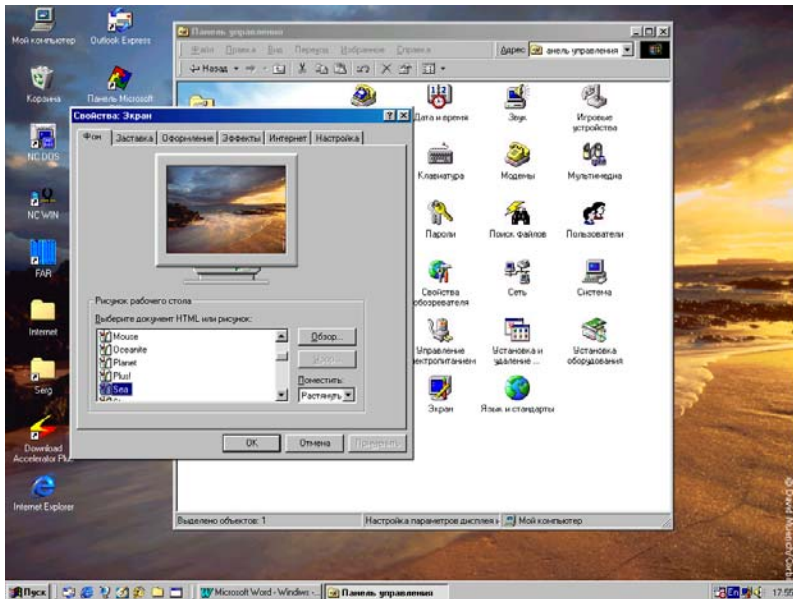


Рис.49. Окно установки свойств экрана.

Существует и другой способ установки рисунка на **Рабочий стол**: щелкните правой кнопкой мыши любой рисунок на **Web** - странице и в Контекстном - Ниспадающем меню выберите команду **Сделать рисунком Рабочего стола**. Для выбора определенной экранной заставки, изменения цветов **Рабочего стола** и окон, добавления элементов Активного **Рабочего стола**, изменения разрешения

экрана и т. д. воспользуйтесь другими вкладками этого диалогового окна.

Экранная заставка

Для установки экранной заставки - хранителя экрана выполните:

1. Выберите вкладку **Заставка** окна **Свойства: Экран**, показанного на рис.50.
2. В меню **Заставка** выберите нужный вид хранителя экрана и щелкните **ОК** (рис.50).
3. Предварительно можно просмотреть, выбранную заставку кнопкой **Просмотр**. Выход из режима просмотра осуществляется нажатием на любую клавишу или перемещением мышки. Кнопкой **Настройка** можно регулировать различные параметры заставки.
4. В этом окне, в меню **Интервал** можно установить промежуток времени через который будет включаться экранная заставка, если компьютер в это время бездействовал.

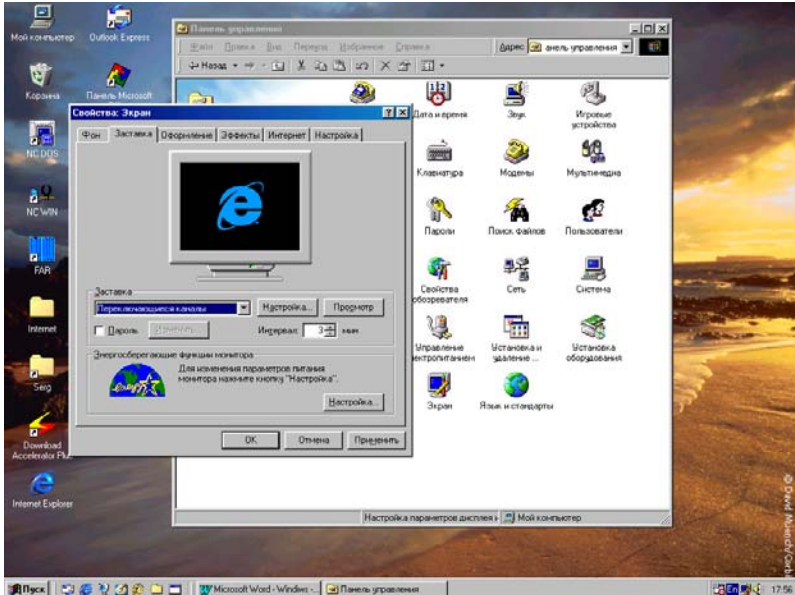


Рис.50. Окно выбора экранной заставки.

5. В разделе **Энергосберегающие функции монитора** кнопкой **Настройка** можно вывести на экран дополнительное окно

Управление электропитанием, позволяющее устанавливать режимы отключения монитора и жесткого диска, а так же определять время включения **Ждущего режима** компьютера.

Запуск заставки происходит автоматически после простоя компьютера в течение промежутка времени, заданного в поле **Интервал**. Для того, чтобы прекратить работу заставки достаточно нажать любую клавишу или переместить указатель мыши.

Разрешение экрана

Для выбора различных режимов разрешения экрана последовательно выполните действия:

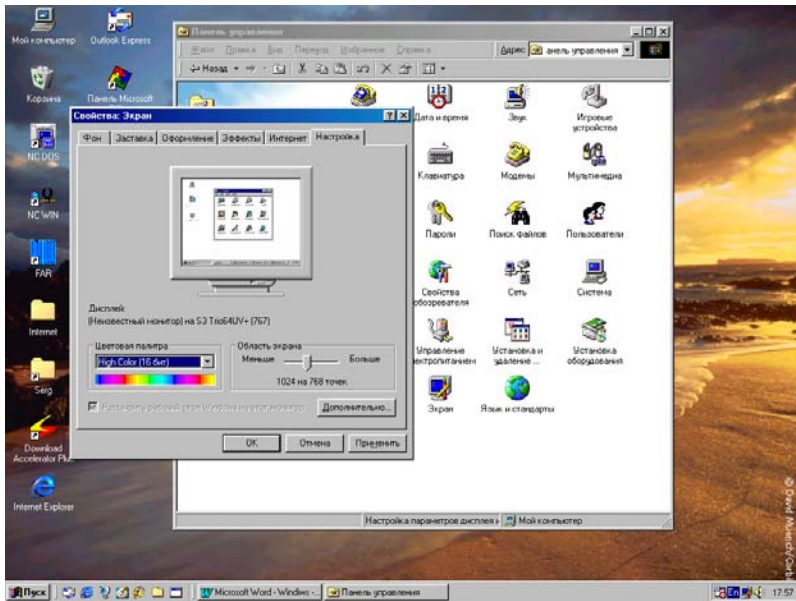


Рис.51. Выбор разрешения экрана монитора.

1. Выберите вкладку **Настройка** окна **Свойства: Экран**, показанного на рис.51.
2. В меню **Цветовая палитра** установите количество отображаемых на экране цветов. Обычно устанавливается 256 цвета при видеопамяти компьютера 1 Мб и режим High Color при 2 Мб и более.

полях **Размер** и **Цвет**, а затем установите нужный шрифт для этого элемента, как показано на рис.52.

3. После изменения настройки отдельных элементов, для их сохранения, следует нажать кнопку **Сохранить**, как и ввести имя новой схемы оформления экрана. Выбранное вами имя появится в списке **Схема**.

4. Если требуется изменить вид сразу всех элементов экрана, выберите определенную схему оформления в поле **Схема**, которая содержит различные стандартные схемы оформления **Рабочего стола** и других элементов системы.

5. После всех сделанных установок щелкните кнопку **ОК** окна **Свойства: Экран**.

Изменения вида шрифтов

Для общего изменения размера шрифтов на экране выполните следующие действия:

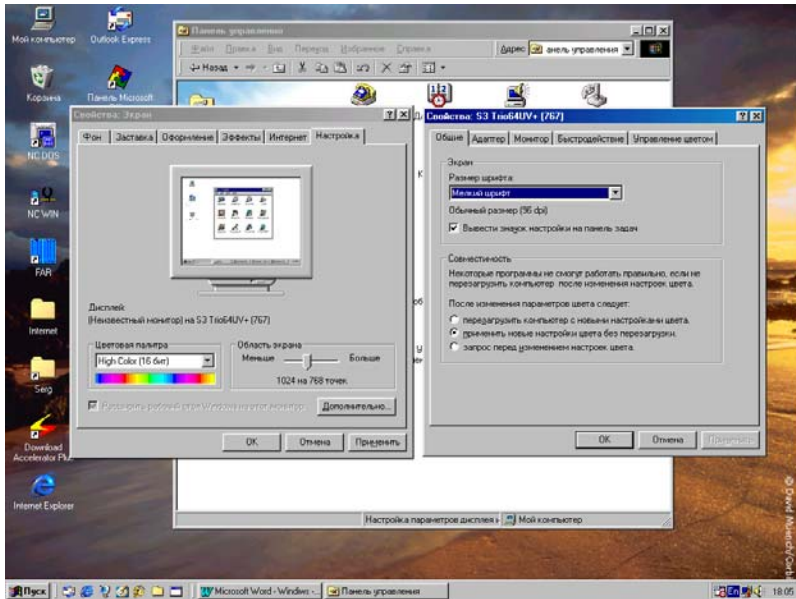


Рис.53. Установка размеров шрифта на Рабочем столе.

1. Откройте вкладку **Настройка** в диалоговом окне **Свойства: Экран** и нажмите кнопку **Дополнительно**.

2. На вкладке **Общие** в поле **Размер шрифта** выберите требуемый размер, которым должны отображаться шрифты на экране компьютера.

3. Для того, чтобы выбрать не стандартный размер шрифтов, выберите пункт **Другой** и определите его размер в новом окне, как показано выше на рис.53.

Если поле со списком **Размер шрифта** недоступно, задайте в группе **Область экрана** разрешение, превышающее 640 x 480 точек (рис.51). Если разрешение 640 x 480 точек является единственно доступным, размер экранного шрифта изменить невозможно.

Настройка Рабочего стола для нескольких пользователей

Если одним и тем же компьютером пользуются несколько человек, можно установить различные конфигурации, или профили **Рабочего стола** и папок. Каждый пользователь при входе в Windows будет видеть только свои собственные настройки, включая настройки **Рабочего стола**, прикладных программ, меню кнопки **Пуск** и папку **Избранное**. Поскольку каждый профиль настроек уникален, внесение изменений одним пользователем не влияет на профили других пользователей.

Для того, чтобы воспользоваться профилями, необходимо сначала задать уникальное Имя и Пароль для каждого пользователя, работающего на данном компьютере:

1. Нажмите кнопку **Пуск**, перейдите к меню **Настройка** и выберите пункт **Панель управления**.

2. Щелкните левой кнопкой мыши значок **Пользователи**. Будет запущен мастер разрешения многопользовательских настроек.

3. Следуйте инструкциям мастера, в окнах которого необходимо задать Имя конкретного пользователя компьютера и его Пароль, а затем щелкните **ОК**.

4. Выберите значок **Пароли** в окне **Панель управления**, который позволяет изменять установленные Пароли и определять режимы работы для каждого пользователя.

5. Перейдите на вкладку **Профили пользователей** открытого на экране окна.

6. Установите параметр **Каждый пользователь устанавливает личные настройки**, а затем щелкните мышкой кнопку **ОК**.

В этом случае любой пользователь, который войдет в Windows получит возможность изменять свои настройки без изменения на-

строек других пользователей. При вашем следующем входе в систему ваши настройки будут полностью сохранены.

Повторите описанную процедуру для личных настроек каждого пользователя. Теперь при запуске системы на **Рабочий стол** будет выводиться список пользователей, где нужно выбрать конкретное Имя и ввести заданный Пароль.

Для завершения сеанса работы данного пользователя нажмите кнопку **Пуск**, выберите команду **Завершение сеанса "Имя пользователя"** и в появившемся окне нажмите кнопку **Да**. При этом будут закрыты все программы данного пользователя, компьютер будет отключен от локальной сети или Интернета и подготовлен к открытию сеанса работы другого пользователя.

Настройка Главного меню и Панели задач

Для повышения эффективности работы с системой можно выполнить пользовательскую настройку меню **Пуск**. Добавить в него часто открываемые папки или файлы, чтобы в любое время быстро открыть их из **Главного меню**, или создать собственные группы файлов и программ. Например, для уменьшения размера меню удалите из него программы, которые вы не используете. Для этого достаточно щелкнуть по выбранному пункту меню правой кнопкой мышки и в Контекстном меню выбрать команду **Удалить**. При удалении элемента из **Главного меню** кнопки **Пуск** программа с компьютера не удаляется.

Настройка Главного меню системы

Для настройки **Главного меню** выполните последовательность действий:

1. Нажмите кнопку **Пуск**, перейдите к разделу **Главного меню Настройка** и выберите пункт **Панель задач и меню "Пуск"**.
2. На экране компьютера откроется диалоговое окно **Свойства: Панель задач** (рис.54).
3. Тот же результат можно получить, щелкнув свободное место на **Панель задач** правой кнопкой мышки и выбрав в Контекстном меню раздел **Свойства**.
4. Перейдите на вкладку **Настройка меню**, щелкнув мышкой по его названию.
5. Нажмите кнопку **Добавить** или **Удалить** и следуйте инструкциям на экране. В частности, при создании нового пункта меню

появляется окно **Создание ярлыка**, где нужно набрать имя устанавливаемой программы или использовать для этого кнопку **Обзор**.

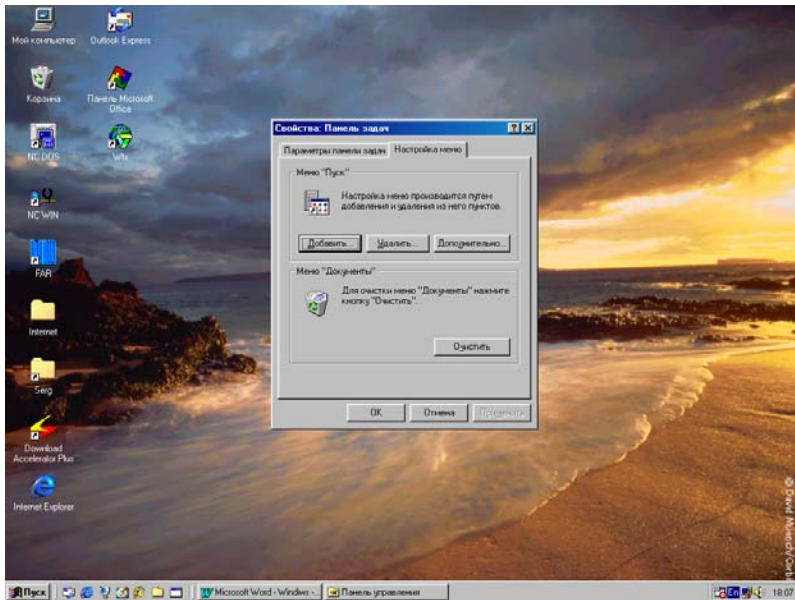


Рис.54. Окна настройки **Главного меню**.

6. В этом случае найдите нужную программу и дважды (или один раз в зависимости от сделанных установок **Рабочего стола**) щелкните мышкой ее значок.

7. Нажмите кнопку **Далее**, а затем выберите нужную папку в структуре меню или создайте новую папку.

8. Введите в поле название новой команды или примите установленное "по умолчанию" и нажмите кнопку **Готово**.

9. Если выводится приглашение выбрать значок для данной программы, выберите его и снова нажмите кнопку **Готово**.

Настройка Панели задач Windows

Для настройки **Панели задач** последовательно выполните, приведенные ниже действия:

1. Нажмите кнопку **Пуск**, перейдите к меню **Настройка** и выберите пункт **Панель задач и меню Пуск**.

2. Откроется диалоговое окно **Свойства: Панель задач**. Перейдите на вкладку **Параметры Панели задач** (рис.55).

3. Включите нужные параметры **Панели задач**, щелкнув мышкой по выбранным разделам и установив там "птичку", а затем щелкните кнопку **ОК**, или нажмите клавишу Enter.

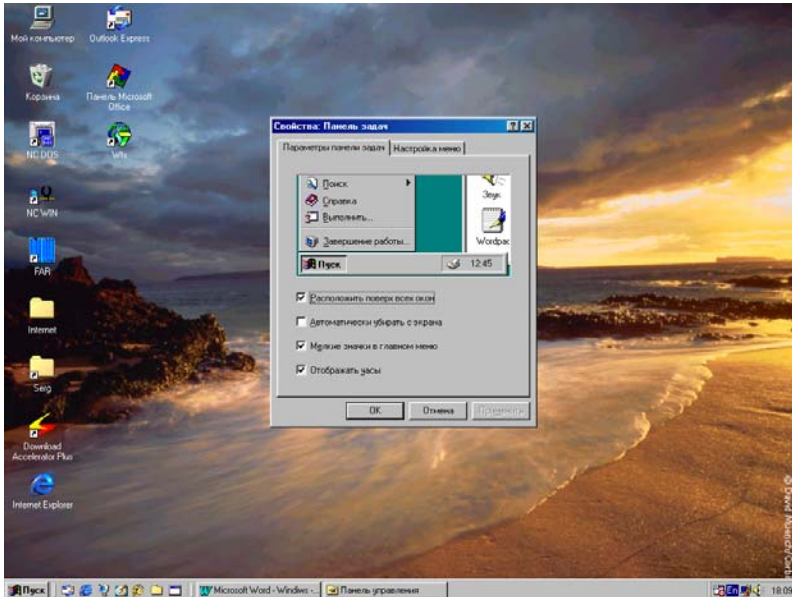


Рис.55. Окно настройки вида **Панели задач**.

Панель задач, "по умолчанию", расположена в нижней части **Рабочего стола**, но ее нетрудно перетащить на любой край экрана. Для этого нужно поставить на нее курсор мыши, и удерживая нажатой левую кнопку перемещать мышку к любому краю экрана.

Кроме того, **Панель задач** можно убирать с экрана на то время, когда она не используется (**Автоматически убирать с экрана**), или всегда отображать поверх всех окон (**Расположить поверх всех окон**), как показано на рис.55.

Отображение панелей инструментов на Панели задач

В новой версии системы **Панели инструментов** можно помещать на **Панель задач** Windows, чтобы еще более упростить доступ к программам, файлам, папкам и избранным **Web** - страницам. Кроме

того, можно разместить **Панель инструментов** в любой части **Рабочего стола**, менять ее размеры и дополнительные настройки.

Существуют четыре стандартные **Панели инструментов**, которые можно добавлять на **Панель задач** уже в готовом виде:

1. Панель **Быстрый запуск** - содержит **Ярлыки** некоторых часто используемых компонентов, в том числе, обозревателя Интернета и почтового приложения Outlook Express.

2. Панель **Адрес** - позволяет сразу на **Панели задач** вводить Интернетовские адреса **Web** - страниц (URL), не открывая программы обозревателя Internet Explorer.

3. Панель **Ссылки** - содержит **Ярлыки** наиболее важных **Web** - узлов, которые вы выбрали для сохранения, что позволяет открывать их без помощи обозревателя.

4. Панель **Рабочий стол** - содержит все **Ярлыки** вашего **Рабочего стола**, группируя их на одной **Панели инструментов** в одном, удобном для доступа месте.

Для вывода на **Панель задач** одной из перечисленных **Панелей инструментов** щелкните правой кнопкой мышки в свободном месте **Панели задач** и в Контекстном меню, в разделе **Панели инструментов** выберите нужную панель, щелкнув по ней и, установив слева от ее названия "птичку"

Помимо готовых **Панелей инструментов**, вы можете использовать свои собственные, создавая их на основе содержимого любой папки:

1. Щелкните правой кнопкой пустое место на **Панели задач**. Откроется Контекстное меню.

2. Переместите указатель мыши на пункт **Панели инструментов** и в дополнительном меню выберите пункт **Создать панель Инструментов**.

Вам будет предоставлена возможность создать пользовательскую **Панель инструментов**, на основе некоторой папки или новую пустую панель, в которую впоследствии можно помещать любые **Ярлыки** из самых различных папок или некоторые значки с **Рабочего стола**.

Запуск приложений с Панели задач

Самый простой способ запуска часто используемого приложения - поместить его **Ярлык** на **Панель задач** (в меню **Быстрого запуска** или любое другое, включенное на **Панели задач** меню):

1. Щелкните в пустом месте **Панели задач** правой кнопкой мыши, в Контекстном меню выберите команду **Панели инструментов** и убедитесь в том, что флажок, например, панели **Быстрый запуск** установлен.

2. Из окна **Мой компьютер** или из **Проводника** перетащите файл приложения (файл с расширением имени .bat, .exe или .com) в строку быстрого запуска на **Панели задач**.

На **Панели задач** рядом с остальными значками строки **Быстрого запуска** появится значок этого приложения.

Изменение структуры меню Программы

Для изменения состава **Главного меню** можно выполнить и следующие действия:

1. Щелкните правой кнопкой мышки кнопку **Пуск** и в Контекстном меню выберите команду **Открыть**.

2. Выберите папку **Программы** или любую другую и откройте ее одинарным или двойным щелчком в зависимости от сделанных установок **Рабочего стола**.

3. Теперь можно добавить или удалить программы или папки, переместить программы в другие папки или разместить одни папки внутри других с помощью обычного перетаскивания мышкой значков - **Ярлыков** этих программ.

4. Все сделанные вами изменения будут отражены в разделе **Главного меню Программы**.

Описанные действия применимы не только к разделу **Программы**, но и к любому другому пункту **Главного меню** по кнопке **Пуск**.

Открытие Web - страницы или файла с помощью Адресной строки

Для открытия Интернетовских объектов с помощью **Адресной строки** выполните действия:

1. Откройте на экране любую папку, например, папку значка **Мой компьютер**.

2. В меню **Вид** окна **Мой компьютер** выберите команду **Панели инструментов**, а затем пункт **Адресная строка**, установив слева от него "птичку". Теперь на **Панели инструментов** этого окна появится меню **Адресной строки**.

3. Введите в **Адресной строке** адрес Интернета или путь к файлу и нажмите клавишу Enter.

Существует и другой способ вывода **Адресной строки** непосредственно на **Панель задач**:

1. Щелкните правой кнопкой мыши пустое место на **Панели задач**. На экране компьютера откроется Контекстное меню.

2. Выберите команду **Панели инструментов**, а затем пункт **Адрес**. На **Панели задач** появится **Адресная строка**, которая будет постоянно доступна в любом сеансе работы с системой, не зависимо от того, какие окна открыты на **Рабочем столе** (рис.56).

3. Введите в **Адресной строке** адрес Интернета (URL -адрес) или путь к файлу на жестком локальном или сетевом диске и нажмите клавишу Enter на клавиатуре.



Рис.56. Вид Адресной строки на **Панели задач**.

Новые элементы Главного меню

По сравнению с предыдущей версией системы в **Главном меню** кнопки **Пуск** появилось несколько новых пунктов - разделов:

1. **Избранное** - теперь избранные **Web** - узлы можно открывать из любого места на компьютере. Список избранных узлов доступен не только из меню **Пуск**, но и из меню в окнах **Мой компьютер**, **Сетевое окружение**, а также в **Проводнике**, **Панели управления** и даже в окне **Корзины** на **Рабочем столе** системы.

2. **Настройка** - в это меню было добавлено два новых пункта, показанные на рис.57:

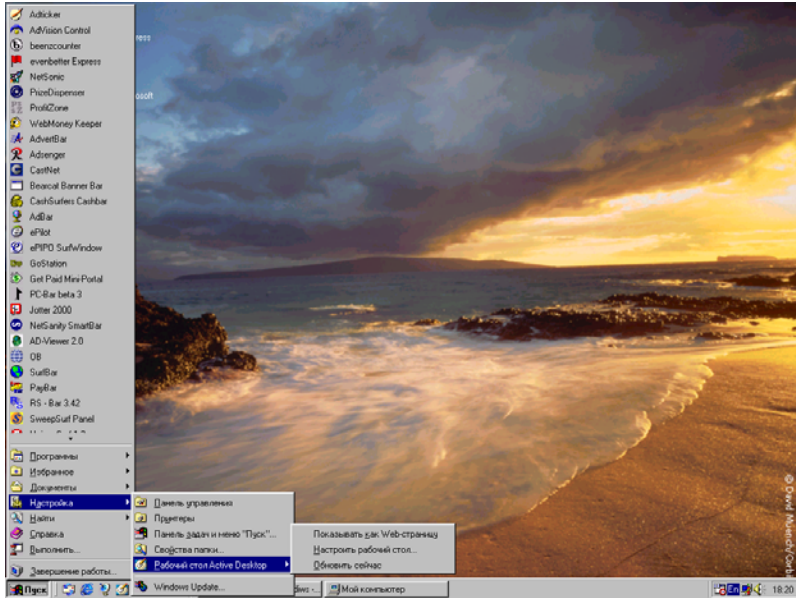


Рис.57. Новые элементы **Главного меню**.

1) **Рабочий стол Active Desktop** - этот элемент меню позволяет настраивать, а также включать и выключать интерфейс **Активного Рабочего стола**.

2) **Свойства папки** - позволяет быстро открыть диалоговое окно **Свойства папки**, где можно изменить такие настройки, как использование двойных и одинарных щелчков мышью, классический или **Web** - стиль **Рабочего стола** и т.д.

3. **Найти** - в этом меню также появился один новый раздел **В Интернете** - этот пункт используется для поиска в Интернете нужной информации. В частности, можно выполнять поиск с помощью поисковых серверов Интернета.

4. **Завершение сеанса "Имя пользователя"** - этот новый пункт позволяет быстро завершить текущий сеанс, работающего в данный момент пользователя, чтобы мог войти в систему и начать работать другой пользователь.

Кроме того, в **Главном меню** кнопки **Пуск** стало легко упорядочивать списки **Избранное** и **Программы**. Для того, чтобы изменить порядок элементов в **Главном меню**, нажмите кнопку **Пуск**, укажите пункт **Программы** или **Избранное** и перетащите нужный элемент (папку или **Ярлык**) в любое новое место. Таким же способом можно перемещать элементы любых других подменю.

Выключение компьютера

Завершив работу с Windows - 98, воспользуйтесь командой **Завершение работы** в меню **Пуск** для закрытия всех окон и программ, а также для подготовки компьютера к выключению. Если какие-то измененные файлы не были предварительно сохранены, система попросит сохранить их или подтвердить выключение компьютера без сохранения этих файлов.

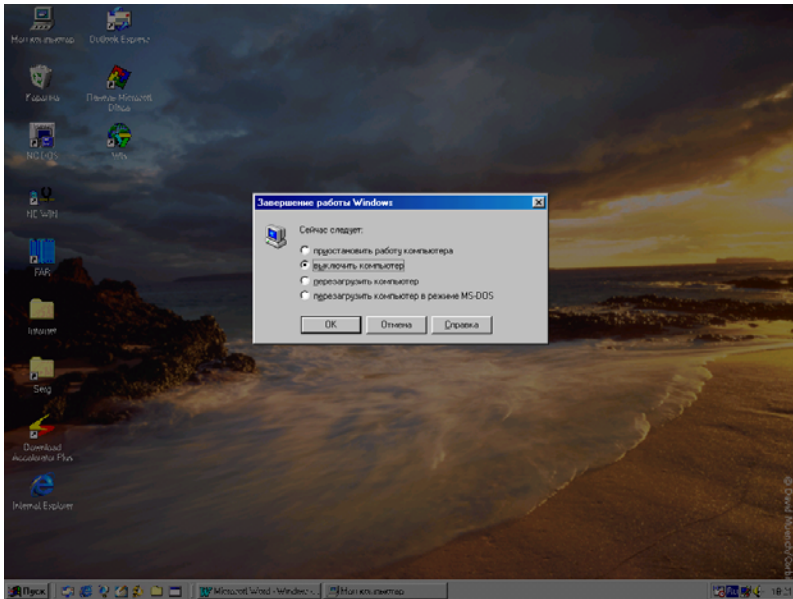


Рис.58. Окно выключения компьютера.

После нажатия кнопки **Пуск** и выбора пункта **Завершение работы** на экране появится диалоговое окно **Завершение работы Windows** (рис.58). Для выключения компьютера выберите пункт **Выключить компьютер** и нажмите кнопку **ОК**. Если компьютер не выключится автоматически, то на экране появится сообщение о том, что теперь можно отключить питание компьютера.

Не выключайте компьютер до тех пор, пока на экране не появится сообщение о том, что это можно сделать. Выключая компьютер без правильного завершения работы операционной системы, вы рискуете потерять некоторые ваши данные.

В этом же меню имеются пункты для перезагрузки компьютера и перевода его в ждущий режим (рис.58):

1. Пункт **Приостановить работу компьютера** переводит компьютер в ждущий режим с выключением монитора и жесткого диска. Вывести компьютер из такого режима можно нажатием на любую клавишу или перемещением мышки.
2. **Перезагрузить компьютер** приводит к полной перезагрузке системы компьютера.
3. Пункт **Перезагрузить компьютер в режиме MS DOS** приводит к перезагрузке с последующим запуском режима имитации операционной системы DOS. Этот режим обычно используется для запуска и работы старых компьютерных программ, рассчитанных только под операционную систему DOS.

НАСТРОЙКА МЫШИ И КЛАВИАТУРЫ

Мышью называется ручное устройство выбора объекта, которое позволяет управлять указателем - курсором мыши на экране монитора. Когда вы двигаете мышь по столу или коврику мыши, на экране перемещается указатель мыши. Для выполнения каких - либо действий над объектом следует поместить указатель на объект и один или два раза нажать кнопку мыши ("щелкнуть" или "дважды щелкнуть" объект). Например, для открытия и работы с файлами обычно используется двойной щелчок. Для переноса файлов и папок из одного места на другое их "перетаскивают" при нажатой левой кнопке мыши, а для выделения файлов используется одиночный щелчок или просто наведение курсора на объект.

Указатель мыши обычно имеет вид стрелки, но он может принимать и другую форму. В частности, при наведении на объект он может превращаться в "руку", а название самого объекта выделяется подсветкой. В таком случае требуется только один щелчок левой кнопкой для открытия выбранного объекта. Если при наведении на объект курсор остается стрелкой, то для выделения нужен одинарный щелчок, а для открытия - двойной.

Щелчок правой кнопкой мыши приводит к выводу на экран Контекстного меню, вид которого существенно зависит от места, где выполнен такой щелчок.

Изменение параметров работы мыши

Для изменения конфигурации работы мыши выполните следующие действия:

1. Откройте диалоговое окно **Мышь** в папке **Панель управления**, значка **Мой компьютер** (рис.59).
2. На вкладке **Кнопки мыши** выберите **Для правши** или **Для левши** и щелкните **ОК**.
3. Кнопку **ОК** можно нажать и после установок режимов работы мыши на всех вкладках открытого окна **Свойства: Мышь**.

Для открытия диалогового окна **Свойства: Мышь** можно также нажать кнопку **Пуск**, выбрать команды **Настройка** и **Панель управления**, а затем щелкнуть значок **Мышь**. Для изменения скорости двойного щелчка левой кнопкой мышки выполните:

1. Перетащите бегунок регулятора в группе **Скорость двойного нажатия** в сторону **Выше** или **Ниже** (рис.59).

2. В **Области проверки** можно потренироваться выполнять двойной щелчок.

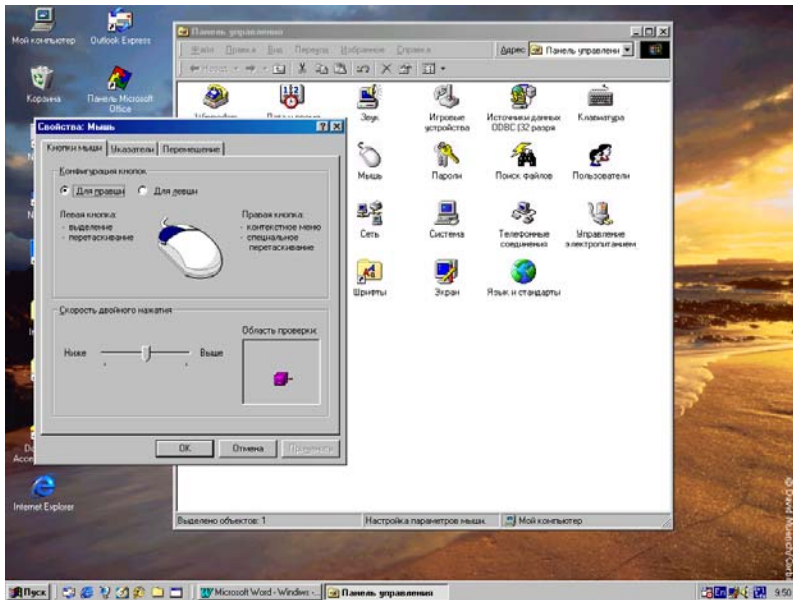


Рис.59. Установка режимов работы мыши.

3. После, сделанных установок щелкните кнопку **ОК** в открытом на экране окне.

Для изменения вида курсора мыши:

1. Перейдите на вкладку **Указатели** открытого окна **Свойства: Мышь** (рис.60).
2. Здесь можно выбрать одну из стандартных **Схем** для указателя мыши.
3. Или, щелкнув кнопку **Обзор**, выберите один из дополнительных вариантов курсоров.

Для изменения скорости перемещения мыши выполните следующие действия:

1. Перейдите на вкладку **Перемещение** окна **Свойства: Мышь** (рис.61).

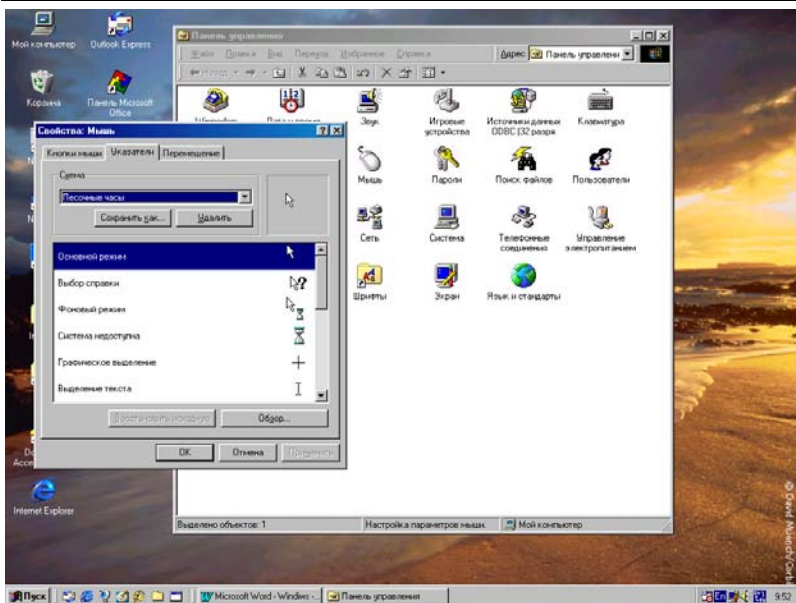


Рис.60. Выбор указателя мыши.

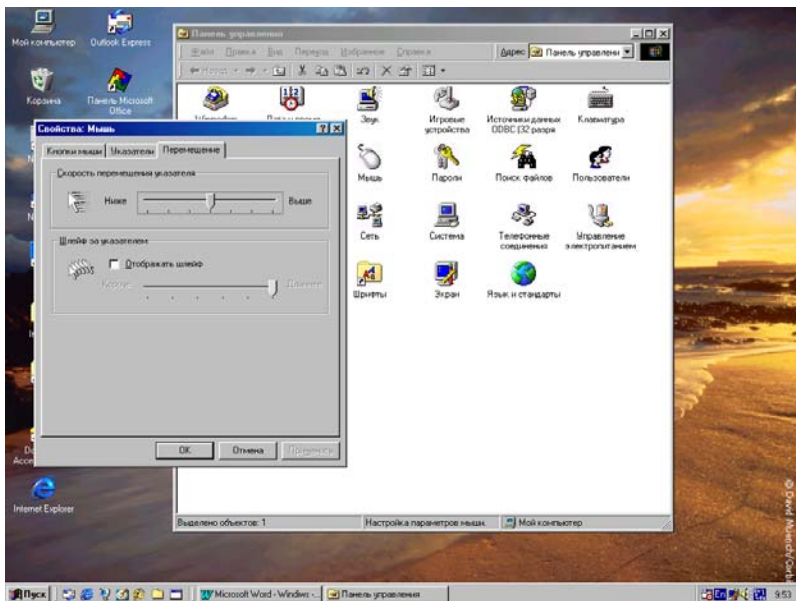


Рис.61. Установка скорости перемещения мыши.

Для того, чтобы изменить промежуток времени между нажатием клавиши и переходом клавиатуры в режим непрерывного ввода выбранного символа, перетащите бегунок регулятора **Задержка перед началом повтора символа** в сторону **Длиннее** или **Короче**. Для изменения скорости мигания экранного курсора выполните действия:

1. Откройте диалоговое окно **Свойства: Клавиатура** в папке **Панель управления**.
2. На вкладке **Скорость** перетащите бегунок регулятора **Скорость мерцания курсора** в сторону **Ниже** или **Выше** (рис.62).
4. После сделанных изменений нажмите кнопку **ОК**, открытого на экране окна.

Скорость мерцания экранного курсора демонстрируется в окне выше регулятора.

СОЗДАНИЕ ЯРЛЫКОВ

Ярлыки позволяют быстро запустить программу или открыть файл, или папку без необходимости искать расположение соответствующего объекта в окне **Проводника** или папке **Мой компьютер**. Особенно удобно и полезно создавать **Ярлыки** для часто используемых программ, файлов или папок и размещать их на **Рабочем столе** или в меню **Быстрого запуска** на **Панели задач**.

Создание ярлыка в папке

Для создания нового **Ярлыка** внутри некоторой папки выполните действия:

1. В окне **Мой компьютер** или программе **Проводник** Windows откройте папку, в которой требуется создать новый **Ярлык** для некоторой программы. Эта программа может находиться в любом месте на жестком диске компьютера.

2. Для создания **Ярлыка**, не выделяя объектов в этом окне откройте меню **Файл** и выберите пункт **Создать**, а затем щелкните команду **Ярлык**, как показано на рис.63.

3. В появившемся окне задайте имя **Ярлыка** или используйте кнопку **Обзор** для поиска нужной папки или программы, для которой создается **Ярлык**. В окне обзора дважды или один раз (в зависимости от сделанных установок **Рабочего стола** и папок) щелкните нужный файл или папку и нажмите кнопку **Далее**.

4. Если будет предложено выбрать значок для этого **Ярлыка** - выберите его и нажмите (щелкните мышкой) кнопку **Готово**.

Для создания **Ярлыка** присутствующего в папке объекта (файла или вложенной папки) нужно:

1. Щелкнуть объект правой кнопкой мышки и в появившемся Контекстном меню выбрать пункт **Создать Ярлык** (рис.64).

2. То же самое можно сделать и через меню **Файл** открытого на экране окна, после выделения выбранного объекта.

3. В этом случае **Ярлык** будет создан сразу и с тем же именем, что выделенный объект.

Помещение ярлыка на Рабочий стол

В окне **Мой компьютер** или в **Проводнике** выберите объект, например, файл, программу, папку, принтер или компьютер, для которого требуется создать **Ярлык**:

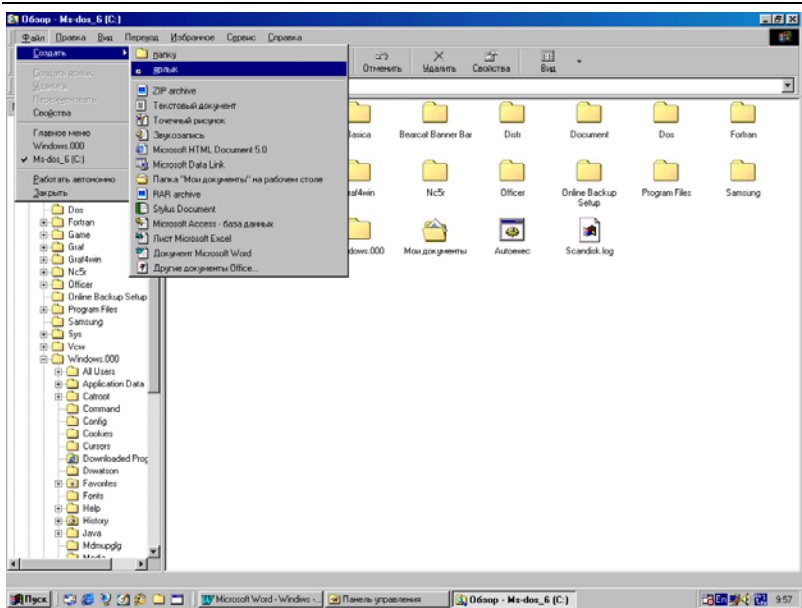


Рис.63. Создание нового Ярлыка.

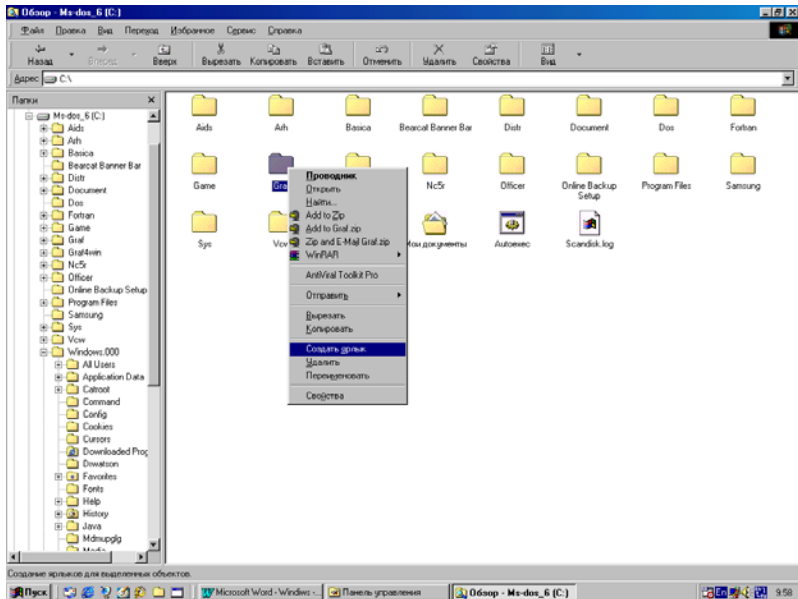


Рис.64. Создание Ярлыка выделенного объекта.

1. В меню **Файл** открытого окна выберите команду **Создать**, пункт **Ярлык**.

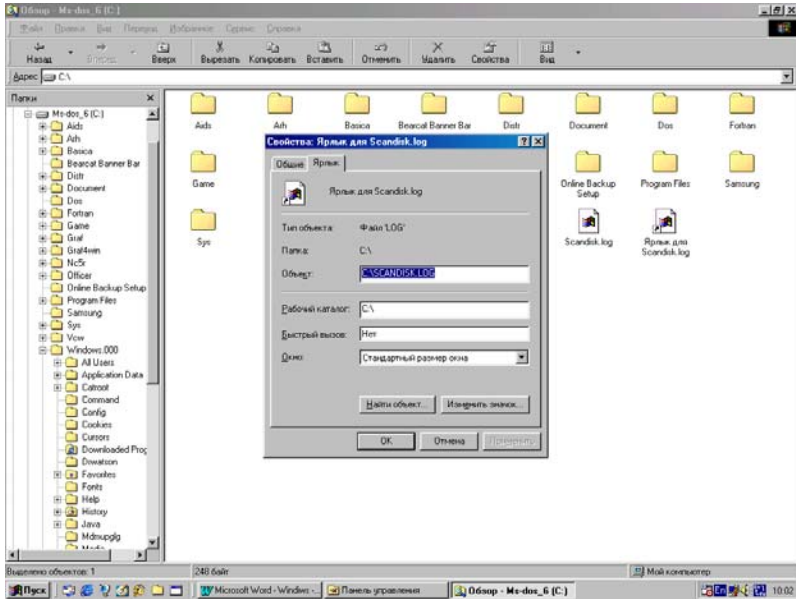


Рис.65. Окно установки свойств **Ярлика** программы.

2. Или щелкните по выбранному объекту правой кнопкой мыши и в появившемся на экране компьютера Контекстном меню выберите раздел **Создать Ярлык**.

3. После создания, перетащите **Ярлык** на **Рабочий стол**, т.е. установите на него курсор и удерживая нажатой левую кнопку мыши переместите **Ярлык** в нужное место.

4. Можно так же переместить **Ярлык** на **Панель задач** в одно из созданных меню или **Панелей инструментов**.

Можно поместить **Ярлык** на **Рабочий стол** и другим способом - щелкните его правой кнопкой мыши и в Контекстном меню выберите пункт **Вырезать**. Затем щелкните правой кнопкой в свободном месте экрана и в новом Контекстном меню выберите пункт **Вставить**.

Для того, чтобы изменить параметры **Ярлика**, например, тип открываемого окна, щелкните **Ярлык** правой кнопкой мышки и в появившемся Контекстном меню выберите команду **Свойства** (рис.65).

Для удаления **Ярлыка**, перетащите его на значок **Корзина** или щелкните по нему правой кнопкой мыши и в Контекстном меню выберите пункт **Удалить**.

Добавление подменю в Главное меню

Для создания нового подменю в любом пункте **Главного меню** кнопки **Пуск** выполните последовательность следующих действий:

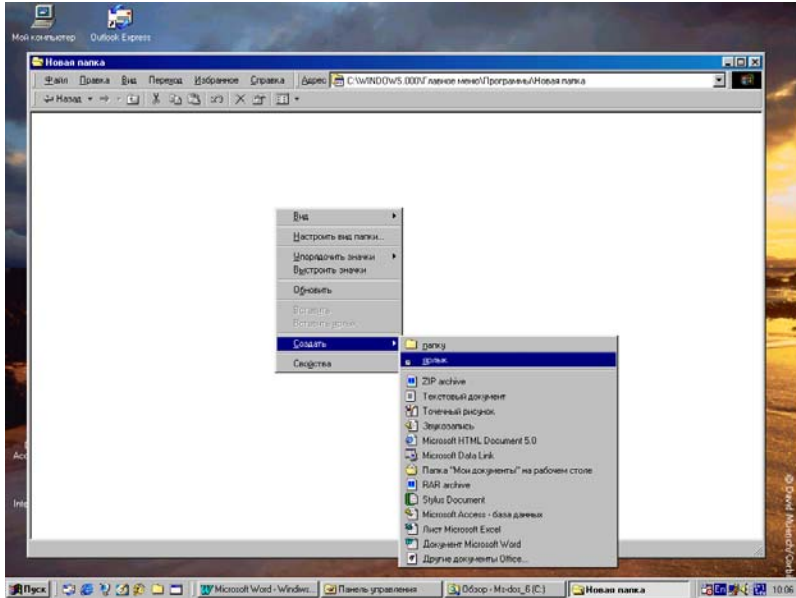


Рис.66. Меню создания **Ярлыка** программы
в папке.

1. Щелкните правой кнопкой мыши кнопку **Пуск** и выберите команду **Открыть**.
2. В появившемся на экране окне откройте, например, папку **Программы**.
3. В меню **Файл** выберите команду **Создать**, а затем щелкните пункт **Папка**. В открытом окне будет создана папка с именем "Новая папка".
4. Щелкните по ней правой кнопкой и в Контекстном меню выберите пункт **Переименовать**. Название папки будет выделено

синей подсветкой и в конце названия появится мигающий экранный курсор.

Теперь сразу можно вводить на клавиатуре имя нового подменю - папки, а далее:

1. Нажмите клавишу Enter для задания папке набранного имени и откройте созданную папку.
2. В меню **Файл** этой папки выберите команду **Создать**, а затем щелкните пункт **Ярлык** (рис.66).
3. С помощью мастера создания **Ярлыков** добавьте в папку - подменю все необходимые пункты - **Ярлыки** нужных программ.
4. После выполнения всех действий закройте открытое на экране окно, в котором был создан новый пункт меню.

Теперь в меню **Программы** кнопки **Пуск** будет присутствовать новый, только что созданный раздел - подменю с выбранным вами именем, содержащий **Ярлыки** некоторых программ. Таким образом, можно создавать подменю в любых пунктах **Главного меню** или же новые пункты самого подменю.

ШРИФТЫ

Система Windows позволяет использовать несколько сотен различных шрифтов для отображения текста и надписей на экране. Обычно используются масштабируемые шрифты True Type, позволяющие получить полное совпадение изображения текста на экране и на листе бумаги после печати.

Установка нового шрифта

Для установки нового шрифта на компьютер, чтобы им можно было пользоваться в прикладных программах выполните действия:

1. Нажмите кнопку **Пуск**, выберите команды **Настройка** и **Панель управления**, а затем щелкните значок **Шрифты**.
2. В меню **Файл** открывшегося окна выберите команду **Установить шрифт**.
3. В следующем окне **Добавление шрифтов** выберите диск, а затем папку, которая содержит новый устанавливаемый шрифт.
4. Щелкните мышкой значок устанавливаемого шрифта, чтобы выделить его.

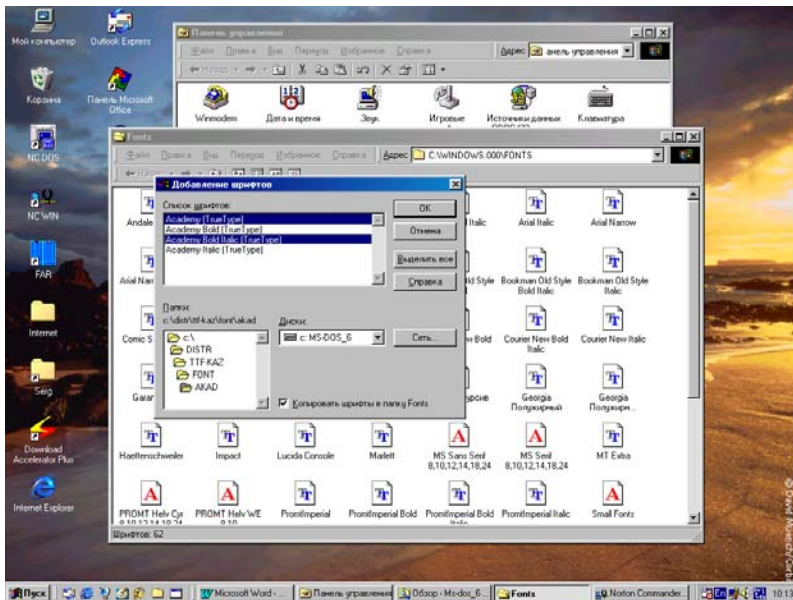


Рис.67. Установка шрифтов.

5. Для выделения сразу несколько шрифтов, выбирайте их, удерживая нажатой клавишу **Ctrl**, как показано на рис.67. Можно так же нажать кнопку **Выделить все**, позволяющую подключать все новые шрифты из выбранной папки.

6. После выбора нужных вам шрифтов щелкните мышкой по кнопке **ОК**.

Для шрифтов True Type, Adobe Type 1 или точечных шрифтов можно также добавить шрифт, перетаскив мышкой соответствующие файлы в папку **Шрифты - Fonts**, которая находится в папке Windows. Для того, чтобы установить шрифты с сетевого диска, не занимая при этом места на локальном жестком диске, снимите перед их установкой флажок **Копировать шрифты в папку Fonts** в диалоговом окне **Добавление шрифтов**. Этот параметр является доступным только при установке шрифтов типа True Type или точечных шрифтов с помощью команды меню **Установить шрифт**.

Удаление шрифта с компьютера

Для удаления любого не нужного шрифта выполните следующие действия:

1. Нажмите кнопку **Пуск**, в **Главном меню** последовательно выберите команды **Настройка** и **Панель управления**, а затем щелкните значок **Шрифты**, чтобы открыть на экране папку **Шрифты**.

2. Выделите значок удаляемого шрифта наведением курсора мыши или щелчком по нему (в зависимости от установок режимов работы папок и **Рабочего стола**).

3. В меню **Файл** выберите команду **Удалить** и в новом окне щелкните кнопку **Да**.

4. Или щелкните по выбранному шрифту правой кнопкой мыши, в Контекстном меню выберите пункт **Удалить** и подтвердите удаление кнопкой **Да**.

5. Для того, чтобы выделить и впоследствии удалить сразу несколько шрифтов, выбирайте их, щелкая по ним мышкой и удерживая нажатой клавишу **Ctrl**.

Просмотр установленных шрифтов

Для просмотра установленных на компьютер шрифтов выполните действия:

1. Нажмите кнопку **Пуск**, выберите команды **Настройка** и **Панель управления**, а затем щелкните мышкой значок **Шрифты**.

2. Для того, чтобы увидеть образец шрифта, дважды (или один раз) щелкните его значок, как показано на рис.68.

При установке принтера могут устанавливаться некоторые встроенные шрифты принтера. Имена встроенных шрифтов принтера не появляются в папке **Шрифты**, однако присутствуют в списках шрифтов большинства приложений Windows, таких как Word Pad или Word.

Для печати выбранного шрифта выполните следующие действия:

1. Нажмите кнопку **Пуск**, выберите команды **Настройка** и **Панель управления**, а затем откройте мышкой папку значка **Шрифты**.

2. В открывшемся окне выделите нужный шрифт и в меню **Файл** щелкните пункт **Печать**.

Поиск схожих шрифтов

Для поиска схожих шрифтов выполните действия:

1. Нажмите кнопку **Пуск**, выберите команды **Настройка** и **Панель управления**, а затем откройте мышкой папку значка **Шрифты**.

2. В меню **Вид**, открывшегося окна выберите команду **Группировать схожие шрифты**.

3. Выберите шрифт в поле со списком **Группировать шрифты по схожести со шрифтом** (рис.69).

Список шрифтов разбивается на три группы в соответствии с признаком схожести с выбранным шрифтом. Сведения, используемые при группировке по подобию, хранятся непосредственно в файле шрифта. Если эти сведения в файле шрифта отсутствуют, шрифт помещается в конец списка, а в поле со списком **Группировать шрифты по схожести со шрифтом** его имя не попадает.

МНОГОЯЗЫКОВАЯ ПОДДЕРЖКА

Система Windows дает возможность создавать документы на многих языках. Однако использование в документах языков центральной и восточной Европы, прибалтийских государств, а также греческого и турецкого языков требует установки многоязыковой поддержки, т.е. подключения к системе дополнительных языков.

Установка многоязыковой поддержки

Для установки многоязыковой поддержки выполните следующие действия:

1. Откройте вкладку **Установка Windows** в диалоговом окне **Установка и удаление программ** папки **Панель управления**.

2. Выберите пункт **Многоязыковая поддержка** и нажмите кнопку **Состав**.

3. Проверьте, чтобы были установлены флажки для всех нужных языков и языковых групп. Обычно выбирают пункты **Кириллица** и **Языки центральной Европы**. При необходимости могут быть подключены и другие языки, присутствующие в списке. Однако, в этом случае придется делать на клавиатуре дополнительные надписи для букв, которые отличаются от английских или русских.

4. Далее нажмите кнопку **ОК** в окне **Многоязыковая поддержка**, а затем кнопку **ОК** окна **Установка и удаление программ**, которое показано на рис.70.

Внесенные изменения вступят в силу после перезагрузки компьютера.

Смена раскладки клавиатуры

Для смены раскладки клавиатуры (выбора типа используемой на компьютере клавиатуры) для уже установленного в системе языка:

1. Откройте мышкой вкладку **Язык** в диалоговом окне **Свойства: Клавиатура** значка **Клавиатура** из папки **Панель управления**.

2. Выберите в списке **Язык**, для которого требуется сменить раскладку и нажмите кнопку **Свойства**, как показано на рис.71.

3. Выберите в списке нужную раскладку и нажмите кнопку **ОК** в окне **Свойства языка**, а затем **ОК** в окне **Свойства: Клавиатура**.

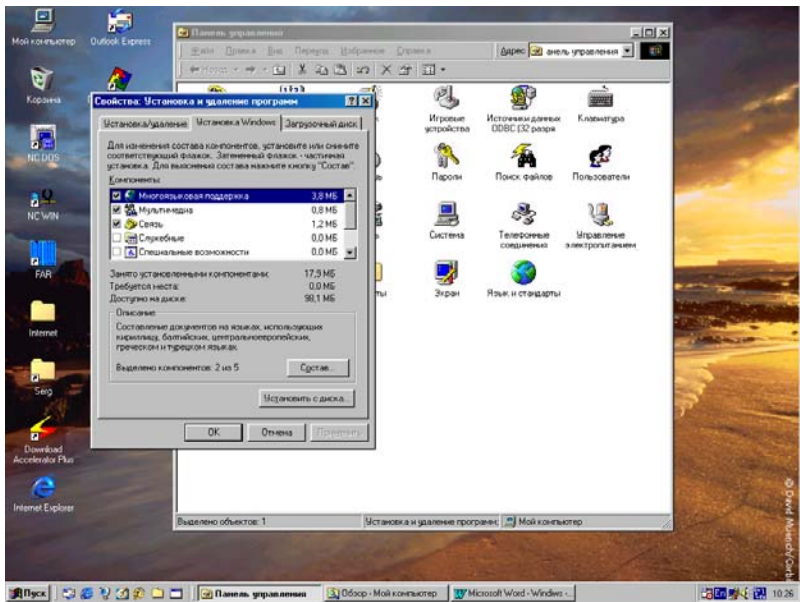


Рис.70. Окно установки многоязыковой поддержки.

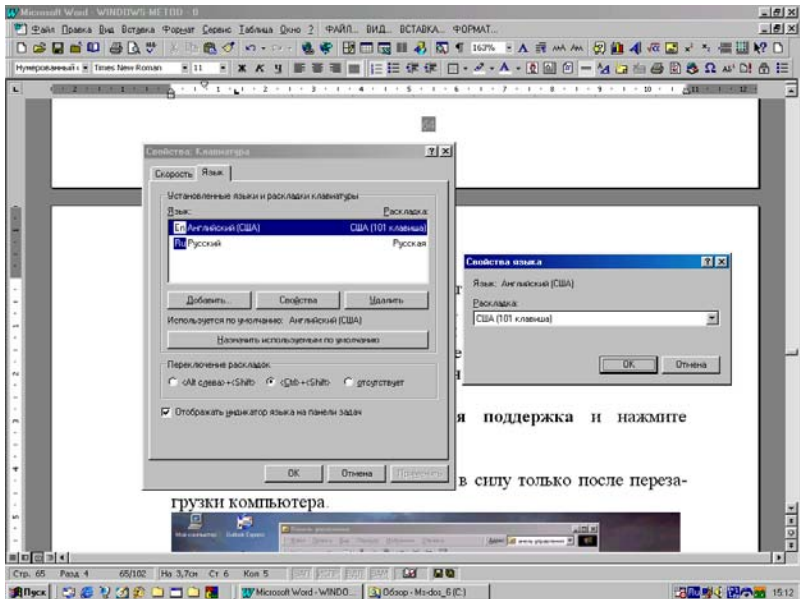


Рис.71. Окно установки раскладки клавиатуры.

В разделе **Переключение раскладок** можно выбрать определенную комбинацию клавиш на клавиатуре, позволяющую переключаться между разными установленными в системе языками из списка **Язык** вкладки **Язык** окна **Свойства: Клавиатура**.

Раскладка клавиатуры обычно устанавливается при установке самой системы на компьютер. Менять ее может понадобиться только при смене клавиатуры, если она совершенно другого типа и имеет другое число клавиш управления. Однако, система обычно сама определяет тип клавиатуры при каждом включении Windows.

Установка дополнительного языка

Для установки дополнительного (еще не установленного в системе) языка на компьютер выполните последовательность следующих действий:

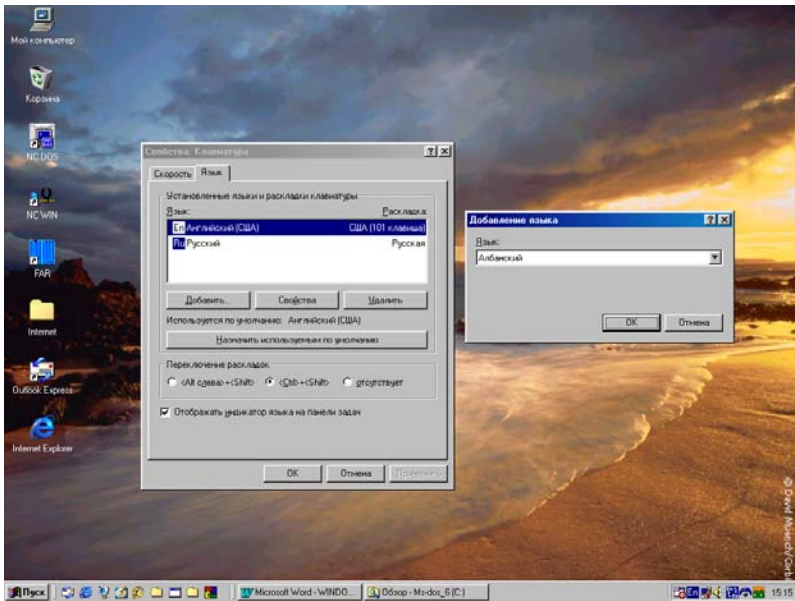


Рис.72. Окно для добавления нового языка.

1. Откройте вкладку **Язык** в диалоговом окне **Свойства: Клавиатура** из папки **Панель управления** и щелкните мышкой кнопку **Добавить**.

2. В диалоговом окне **Добавление языка** выберите нужный язык, щелкнув по нему и нажмите кнопку **ОК** окна **Добавление языка**. Выбранный язык появляется в меню **Язык** вкладки **Язык**.

3. Теперь щелкните кнопку **ОК** в окне **Свойства: Клавиатура**, как показано на рис.72 и язык будет полностью установлен.

Кнопка **Назначить используемым "по умолчанию"** позволяет сделать выбранный язык основным, который сразу включается при каждом запуске системы.

Если флажок **Отображать индикатор языка на панели задач** на вкладке **Язык** установлен, а в системе установлено не менее двух языков, то на **Панели задач** появляется индикатор, отображающий текущий язык. Для того, чтобы быстро сменить язык, достаточно щелкнуть индикатор, а затем выбрать нужный язык из появившегося списка.

Для открытия вкладки **Язык** в диалоговом окне **Свойства: Клавиатура** можно также нажать кнопку **Пуск**, выбрать команды **Настройка** и **Панель управления**, а затем щелкнуть левой кнопкой мышки **Ярлык** - значок **Клавиатура**.

Удаление языка

Для удаления установленного в системе языка выполните следующие действия:

1. Откройте вкладку **Язык**, щелкнув по ней мышкой в диалоговом окне **Свойства: Клавиатура**.

2. Выберите в списке удаляемый язык, нажмите кнопку **Удалить** и в новом окне подтвердите удаление кнопкой **Да**.

Удаленный язык всегда можно установить заново методами, описанными в предыдущем параграфе.

РАБОТА С ПАПКАМИ И ФАЙЛАМИ

В процессе работы с программами системы или установки новых программ на компьютер на жестком диске создаются файлы. Файлы могут находиться где угодно - на локальном или сетевом диске, дискете или других носителях данных (например, на магнитной ленте). Эффективным средством организации любых файлов является их хранение на диске в определенных папках - директориях (каталогах).

Открытие файла или папки

Для открытия некоторой папки или файла выполните следующие действия:

1. Запустите программу **Проводник** Windows, щелкнув, например, по кнопке **Пуск** правой кнопкой мыши и выбрав **Проводник** из открывшегося Контекстного меню.
2. Выберите диск, дискету или компьютер в папке **Сетевое окружение** (в самой программе **Проводник**), на котором содержится нужный файл или папка.
3. Щелкните файл или папку (если установлен режим одиночного щелчка), которую следует открыть.
4. При щелчке по выбранному файлу автоматически запускается программа в которой был создан этот файл и которая позволяет продолжить работу с ним.
5. При щелчке по папке она открывается и в окне **Проводника** будет показано ее содержимое.

Напомним, что для того, чтобы выделить объект нужно либо поставить на него курсор мыши или один раз щелкнуть по нему. В первом случае, при наведении курсора на объект он превращается в "руку с указательным пальцем" и объект выделяется синей подсветкой. Если этого не происходит и курсор мыши при наведении на объект остается "стрелкой", то для его выделения нужен однократный щелчок.

Для выделения нескольких объектов щелкайте по ним, удерживая нажатой клавишу **Ctrl**. Для выделения непрерывной последовательности элементов щелкните на первом из них, нажмите и удерживайте нажатой клавишу **Shift**, и щелкните на последнем элементе в последовательности (режим двукратного щелчка). Для выделения

всего содержимого окна выберите в меню **Правка** команду **Выделить все**.

При выделении объекта наведением курсора, для его открытия нужно один раз щелкнуть по нему. Если выделение осуществляется щелчком мыши, то для открытия объекта требуется двойной щелчок.

Изменение программы, используемой для открытия файла

Операционная система Windows позволяет поставить в соответствие разным типам файлов программы, автоматически запускаемые при открытии файлов такого типа. Так, при открытии текстового файла запускается программа **Блокнот** Windows, а при открытии файлов типа .doc будет запущен текстовый редактор Word, если он установлен на данном компьютере. В противном случае будет включаться программа **Word Pad**, входящая в саму систему Windows.

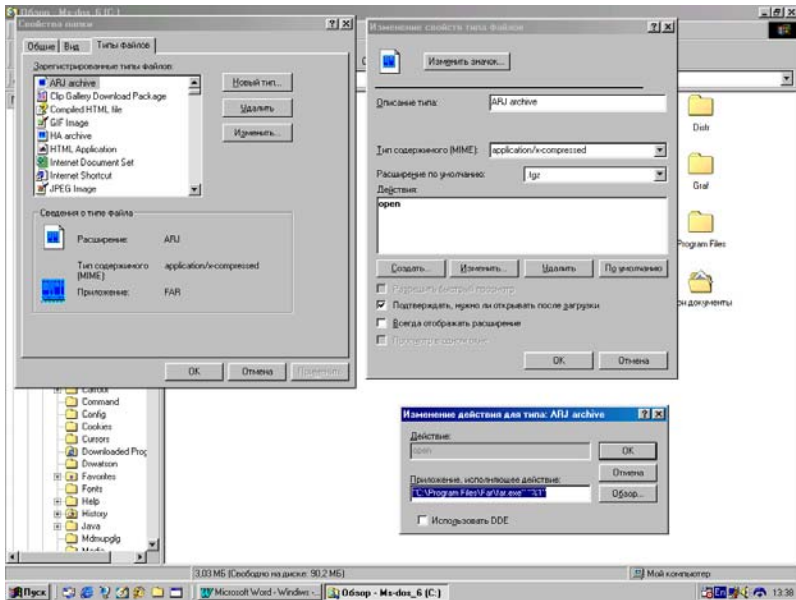


Рис.73. Окна для установки программ открытия файлов.

Программу, с помощью которой автоматически открывается файл, можно изменить:

1. В окне **Мой компьютер** или в программе **Проводник** Windows выберите команду **Свойства папки** в меню **Вид** любого из этих окон. **Свойства папки** можно вызвать и через кнопку **Пуск**, раздел **Настройка**.

2. В новом открывшемся на экране окне перейдите на вкладку **Типы файлов**.

3. В поле **Зарегистрированные типы файлов** выберите тип файла, который должен открываться другой программой и нажмите кнопку **Изменить**.

4. В поле **Действия** нового окна **Изменение свойств типа файлов** выберите пункт **Ореп** и снова щелкните левой кнопкой мыши кнопку **Изменить**.

5. В открывшемся на экране третьем окне **Изменение действия для типа...** в поле **Приложение, исполняющее действие** введите путь к программе, которая должна использоваться для открытия файлов выбранного типа.

6. Для выбора нужной программы можно также использовать кнопку **Обзор** (рис.73).

7. Затем трижды щелкните кнопку **ОК** в трех открытых на экране окнах.

Открытие недавно использованных файлов

Доступ к часто используемым документам системы существенно облегчен, поскольку в разделе **Документы** меню **Пуск** помещаются ссылки на все недавно открывавшиеся файлы. Кроме того, на **Рабочем столе** имеется папка **Мои документы**, которую удобно применять для хранения часто используемых файлов и папок.

Для удобного доступа к часто используемым файлам можно также создать **Ярлыки** на **Рабочем столе**. **Ярлык** не изменяет местонахождение файла - он лишь является ссылкой на файл или папку, позволяющей быстро открыть их. Если удалить с **Рабочего стола** некоторый **Ярлык**, исходный файл или папка останутся без изменений.

Для открытия недавно используемого объекта последовательно выполните действия:

1. Нажмите кнопку **Пуск** и в появившемся меню выберите раздел **Документы**.

2. В дополнительном меню щелкните мышкой нужный документ. При этом будет запущено приложение (определенная программа), которое работает с данным файлом. В этом списке хранится до 15 ранее сохраненных документов из самых различных программ.

Некоторые программы не помещают имена своих файлов в меню **Документы**. Если нужный документ отсутствует в списке, нажмите кнопку **Пуск**, выберите команду **Найти**, а затем раздел **Файлы и папки**. Найдите файл с помощью диалогового окна **Найти**. Или запустите приложение, которое работало с нужным файлом, например, редактор Word и в меню **Файл** этой программы выберите файл, с которым нужно продолжить работу. В меню **Файл** обычно хранится до девяти последних файлов документов.

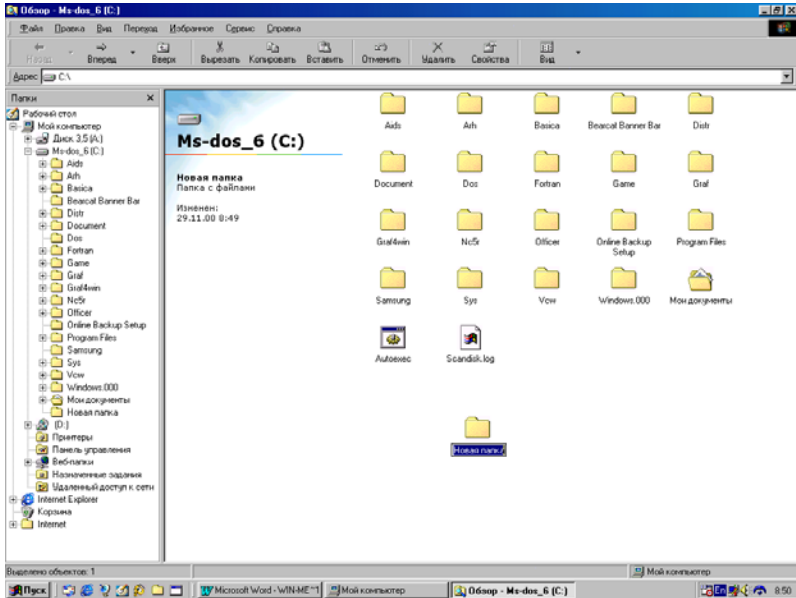


Рис.74. Окно **Проводника** для создания новой папки.

Создание новой папки

Для создания новой папки с заданным вами именем выполните следующие действия:

1. В левой части окна программы **Проводник** выберите диск, на котором требуется создать новую папку, например, диск C: . Для этого нужно щелкнуть по имени диска, выделив его синей подсветкой. При этом в правой части окна будет показано его содержимое - состав диска.

2. В меню **Файл** выберите команду **Создать**, а затем щелкните пункт **Папка**.
3. В правой части окна появляется новая папка с именем, которое присвоено ей "по умолчанию" - "Новая папка".
4. Щелкните по ней правой кнопкой мыши и в Контекстном меню выберите пункт **Переименовать**. После этого имя папки будет выделено синей подсветкой с мигающим в конце имени экраным курсором.
5. Наберите на клавиатуре имя новой папки и нажмите клавишу Enter. При этом имя "по умолчанию" будет заменено на вводимое вами, как показано на рис.74.

Если новую папку требуется создать внутри другой папки, щелкните мышкой эту папку в левой части окна **Проводника** и выполните шаги 2 - 5.

Возможен и другой способ создания новой папки - щелкните правой кнопкой мыши в свободном месте окна какой - либо открытой на экране папки (например, в правой части окна **Проводника**) и в появившемся на экране Контекстном меню выберите раздел **Создать**, подраздел **Папка**, а затем повторите действия пункта 5.

Изменение имени файла или папки

Для переименования файла или папки последовательно выполните следующие действия:

1. В окне **Мой компьютер** или окне **Проводника** выделите файл или папку, которую требуется переименовать.
2. В меню **Файл** выберите команду **Переименовать** или щелкните по этому объекту правой кнопкой мыши и в Контекстном меню выберите команду **Переименовать**. Имя выбранного объекта выделится синей подсветкой, справа от которой будет мигать экранный курсор.
3. Введите на клавиатуре новое имя и нажмите клавишу Enter на клавиатуре.

Обычно эти действия выполняются в правой области окна **Проводника**, но они допустимы и в левой части его окна. Предельная длина имени файла или папки составляет 255 символов, включая пробелы, а имена не должны содержать следующих символов:

\ / : * ? " < > |

Удаление файла или папки

При удалении файла или папки они временно помещаются в папку **Корзина**, которая находится на **Рабочем столе**. Если вы передумали удалять файл или папку их всегда можно **Восстановить** из **Корзины**. Однако при полной очистке корзины все попавшие в нее элементы всегда удаляются с компьютера и их уже ни какими средствами **Восстановить** не удастся.

Для удаления файла или папки последовательно выполните следующие действия:

1. В окне **Мой компьютер** или в программе **Проводник** Windows выделите файл или папку, которую требуется удалить с помещением их в **Корзину**.
2. В меню **Файл** этого окна выберите команду **Удалить**, а затем в новом, появившемся на экране окне подтвердите удаление кнопкой **Да**.
3. Или щелкните по объекту правой кнопкой мышки и в Контекстном меню выберите пункт **Удалить**. А затем также подтвердите удаление объекта.

Удалить объект после его выделения можно и кнопкой **Удалить** на **Панели инструментов** открытого на экране окна, если включена его панель инструментов. Можно также напрямую перетащить мышкой значок файла или папки на значок **Корзина** на **Рабочем столе**. Если при перетаскивании удерживать клавишу Shift, то объект будет удален без помещения в папку **Корзина** и его уже нельзя будет **Восстановить**.

Корзина хранит удаленные объекты в течение заданного в ее параметрах времени при условии, что объем удаленных объектов не превышает заданной величины.

Восстановление удаленных объектов

Для восстановления из **Корзины** удаленных файлов и папок выполните следующие действия:

1. Откройте папку **Корзина** на **Рабочем столе** - экране вашего компьютера.
2. В появившемся окне выделите объект, который нужно **Восстановить**.
3. В меню **Файл** открытого окна **Корзины** выберите команду **Восстановить**.

4. Или щелкните этот объект правой кнопкой мыши и в Контекстном меню выберите команду с таким же именем (рис.75).

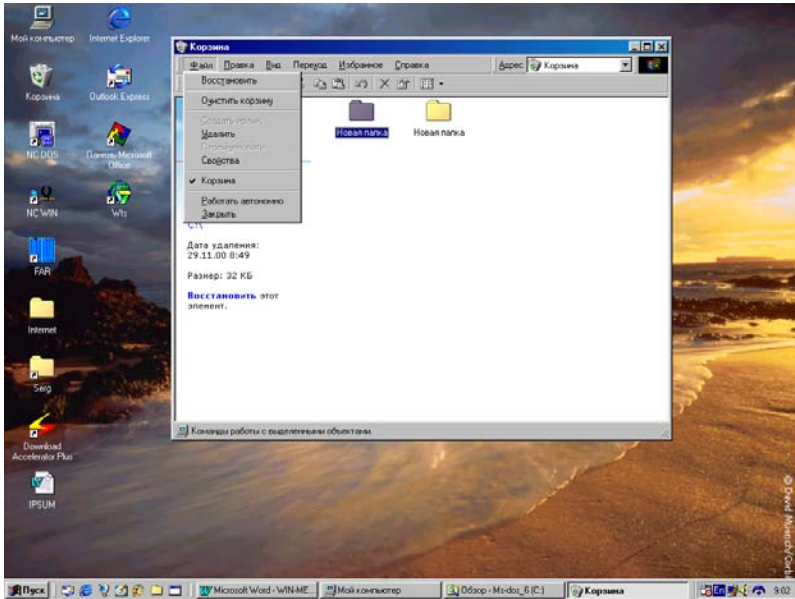


Рис.75. Удаление или восстановление объектов из **Корзины**.

Восстановленный объект будет помещен в то же место, из которого был ранее удален.

Окончательное удаления файлов

Для полного и окончательного удаления объекта с компьютера выполните действия:

1. Щелкните на **Рабочем столе** значок **Корзина**. На экране появится окно **Корзина**.
2. В меню **Файл** этого окна выберите команду **Очистить корзину** (рис.75).

Если нужно удалить только некоторые объекты, выделите их, щелкая по ним мышкой и удерживая нажатой клавишу Ctrl, а затем в меню **Файл** окна **Корзина** выберите команду **Удалить**.

Изменение способа отображения объектов в папке

Для изменения вида отображения объектов в окне открытой папки выполните действия:

1. В окне **Мой компьютер** или в окне **Проводника** выберите в меню **Вид** одну из команд - **Крупные значки**, **Мелкие значки**, **Список** или **Таблица**.
2. Подменю **Упорядочить значки** меню **Вид** позволяет отсортировать объекты по определенным признакам - по имени, размеру, дате создания или типу, в зависимости от выбранного представления.

Для того, чтобы выяснить назначение любой команды меню, достаточно установить на нее указатель мыши. Пояснение к команде выводятся в строке состояния, расположенной в нижней части открытого на экране окна. Если строка состояния в окне отсутствует, выберите в меню **Вид** команду **Строка состояния** и щелкните по ней мышкой.

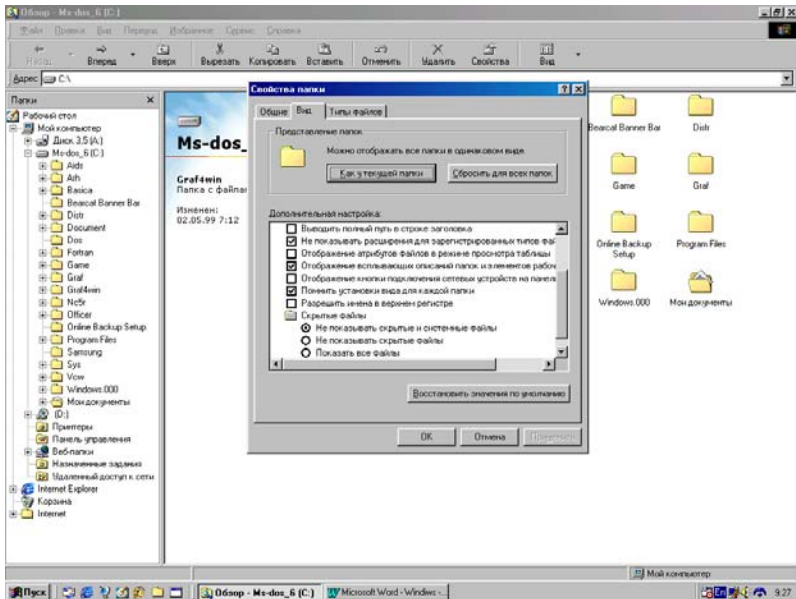


Рис.76. Установка режимов показа имен файлов.

Отображение имен файлов и расширений имен файлов

Для показа на экране имен всех файлов, имеющихся на компьютере выполните:

1. В окне **Мой компьютер** или в **Проводнике** Windows выберите папку, которую требуется просмотреть.
2. В меню **Вид** этого окна щелкните мышкой команду **Свойства папки**.
3. Перейдите на вкладку **Вид** и в разделе **Скрытые файлы** меню со списком **Дополнительная настройка** выберите параметр **Показать все файлы**, как приведено на рис.76.

Для показа на экране всех расширений имен файлов выполните действия:

1. Выполните действия 1 и 2, описанные в предыдущем пункте этого параграфа.
2. На вкладке **Вид** в разделе **Файлы и папки** меню со списком **Дополнительные настройки** снимите флажок **Не показывать расширения для зарегистрированных типов файлов**.

Копирование файла или папки

После создания файла или папки может понадобиться перенести или скопировать ее в другое место. Избегайте переноса стандартных программ - папок и системных файлов. Это может привести к нарушениям в работе системы в целом.

Для копирования любых объектов (файлов, **Ярлыков** или папок) выполните действия:

1. В окне **Мой компьютер** или в программе **Проводник** Windows выберите (выделите) файл или папку (или их группу), которую требуется скопировать.
2. В меню **Правка** или на **Панели инструментов** выберите команду (кнопку) **Копировать**.
3. Откройте папку или диск, куда требуется вставить копию, выбранного объекта.
4. В меню **Правка** или на **Панели инструментов** выберите команду (кнопку) **Вставить**.

Для переноса объекта в другое место все действия выполняются точно так же, как описано выше, только вместо команды (кнопки) **Копировать** нужно использовать команду (кнопку) **Вырезать**.

Отправка файла или папки на диск

Для копирования объекта на дискету или в некоторое другое место на компьютере выполните следующее:

1. Если файл или папка копируется на дискету, вставьте дискету в дисковод.
2. В окне **Мой компьютер** или в **Проводнике** Windows выделите файл или папку, которую требуется скопировать.
3. В меню **Файл** выберите команду **Отправить**, а затем выберите дисковод, на который копируется файл или папка.

По команде отправки объекта на диск отправляется копия файла или папки, а оригинал объекта остается в исходном положении. Точно так же можно создать **Ярлык** папки на **Рабочем столе** или отправить объект по электронной почте.

РЕЖИМ ПОИСКА

Поиск нужного файла (только по его имени или "вручную") обычно занимает достаточно много времени, особенно если вы забыли в какой папке он был сохранен. Однако, его можно ускорить, указав более конкретно тип файла, например, документы редактора Word.

В системе Windows программа **Поиск файлов** служит для ускорения поиска файлов. Для того, чтобы поиск выполнялся быстрее, Индекс (создаваемый заранее список файлов определенного типа, имеющихся на компьютере) программы **Поиск файлов** должен находиться в том же каталоге, где производится поиск, или в вышестоящем каталоге, причем он должен охватывать все искомые файлы.

Для того, чтобы проверить наличие Индекса на компьютере, щелкните дважды значок **Поиск файлов** на **Панели управления** в окне **Мой компьютер**. В меню **Индекс** выберите команду **Обновить**, а затем выберите индекс по его каталогу. Как правило, если Индекс существует, он находится в корневом каталоге диска. Если неизвестно, существует ли Индекс и где он находится, попробуйте создать Индекс в корневом каталоге того диска, в котором предполагается произвести поиск файла. Если в корневом каталоге или в одном из вложенных каталогов имеется уже созданный Индекс, вместо создания нового Индекса будет автоматически предложено обновить имеющийся. Если Индекс отсутствует, его можно создать, выбрав команду **Создать индекс** в меню **Индекс** (рис.77).

Если индекс существует, убедитесь, что он охватывает все типы файлов, которые обычно приходится искать. Если это не так, индекс можно удалить и создать снова. При создании индекса удобно использовать параметр **Документы MS Office** и **Web**. Этим индексом будут охвачены все документы программ MS Office, а также текстовые файлы, файлы формата RTF и гипертекст HTML документы.

Параметр **Все файлы** можно выбирать только в том случае, если приходится часто искать файлы, не упомянутые в списке. Причем Индексы, использующие параметр **Все файлы**, обычно занимают на жестком диске много места.

Кроме того, убедитесь, что установлен флажок **Автоматическое обновление индекса**. Если производится поиск элементов файлов убедитесь, что в диалоговом окне **Создание индекса** установлен флажок **Ускорить поиск по фразам**. При использовании подстановочных знаков (* или ?) или установки флажка **С учетом регистра** в окне **Поиска** время поиска заметно увеличивается.

Обычно размер Индекса не превышает 5 процентов от размера документов, включенных в Индекс. Если установлен флажок **Уско-**

ритель поиск по фразам, размер индекса может составлять около 30 процентов от размера документов, включенных в Индекс.

Если Индекс предварительно не создан, то сам процесс поиска файлов в режиме **Поиск** будет протекать заметно медленнее.

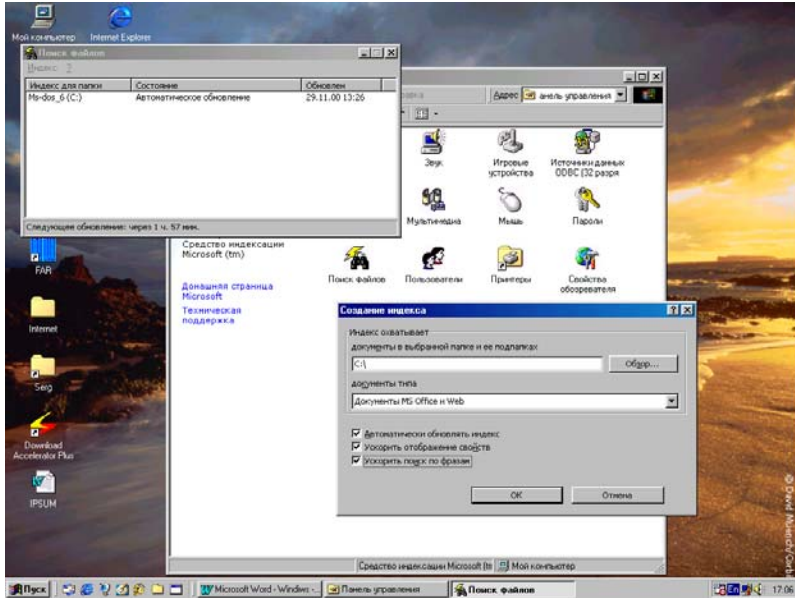


Рис.77. Окна создания Индекса.

Поиск файла или папки

Для поиска файлов на сетевом или жестком диске можно использовать раздел **Найти** меню кнопки **Пуск**. Поиск файла может производиться по имени, типу, дате последнего изменения, по наличию в файле определенного текста или по указанным свойствам файла. Для того, чтобы найти файл по заданным свойствам, например, по имени автора, теме или заголовку, или по свойствам, не относящимся к тексту, например, по дате создания, следует задать нужные параметры поиска в окне **Найти**.

Свойствами файла называются определенные сведения, упрощающие его поиск, такие как описание, имя автора, тема, ключевые слова и т.п. Некоторые свойства, например размер файла, дата создания и вид (тип) файла, добавляются автоматически при его сохранении. Дополнительные свойства файла, например, описание и ключевые слова, однозначно его определяющие, можно ввести вручную.

Команда **Найти** позволяет быстро просмотреть некоторый диск или весь компьютер в поисках нужного файла:

1. Нажмите кнопку **Пуск**, выберите команду **Найти**, а затем раздел **Файлы и папки**.

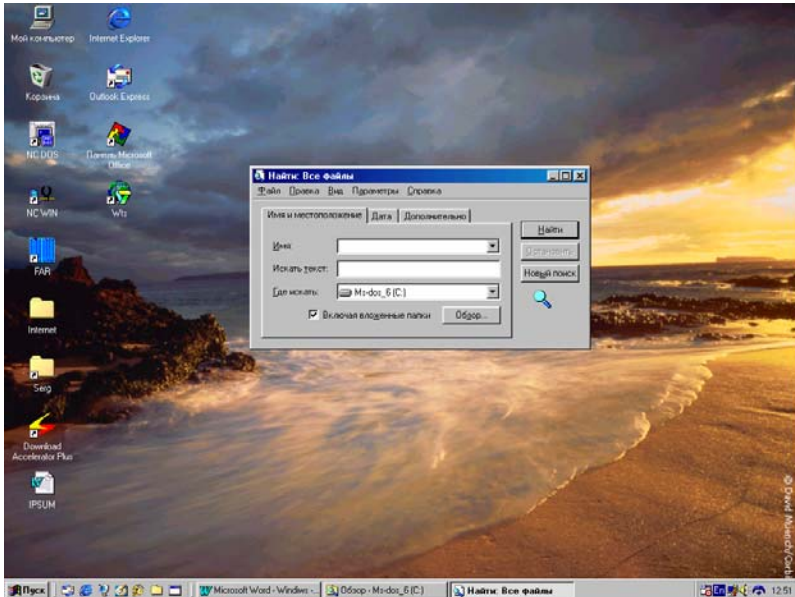


Рис.78. Окно **Поиска** файла по его имени.

2. В поле **Имя** введите полное имя файла или его часть. Здесь можно использовать замещающие символы звездочка (*) и знак вопроса (?).

3. В поле **Искать текст** введите слово или фразу, если имя файла неизвестно, но известно, что он содержит определенный текст.

4. Кнопкой **Обзор** выберите папку или диск, в которой начинается поиск.

5. Щелкните далее кнопку **Найти**, как показано выше на рис.78.

Для того, чтобы задавать условия поиска с учетом регистра символов, в меню **Параметры** окна **Найти** установите метку у команды **С учетом регистра**. Метка устанавливается при выборе команды и

снимается при ее повторном выборе (рис.78). Для того, чтобы прервать процесс поиска, нажмите кнопку **Остановить** окна **Найти**.

Дополнительные возможности поиска

Файлы можно искать, например, по дате создания, их размеру или ключевым словам:

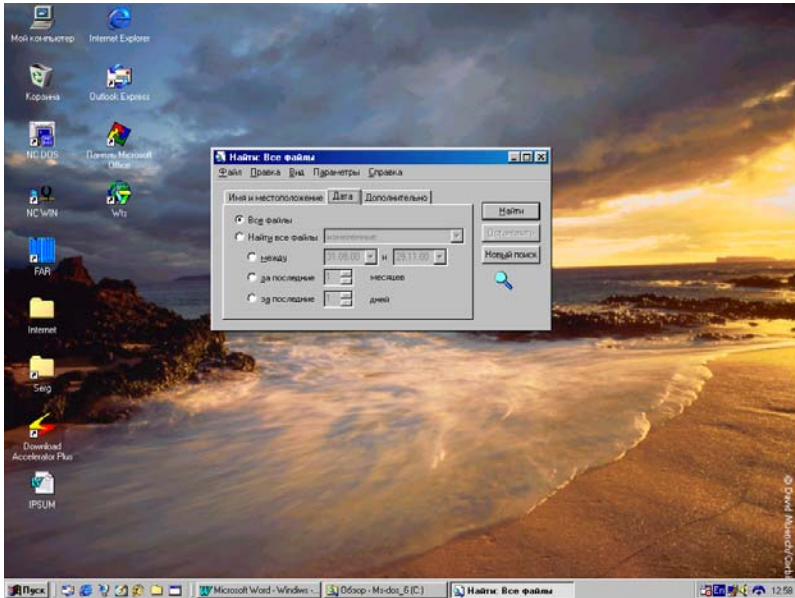


Рис.79. Окно установки режимов **Поиска** по дате.

1. Нажмите кнопку **Пуск**, выберите команду **Найти**, а затем раздел **Файлы и папки**.
2. Выберите вкладку **Дата** для поиска файлов, созданных или измененных в конкретном диапазоне дат создания. Окно этой вкладки показано на рис.79.
3. Для поиска файлов определенного типа или размера выберите вкладку **Дополнительно**.
4. Если нужно искать файл по известному тексту, в поле **Искать текст** вкладки **Имя и местоположение** окна **Файлы и папки** введите требуемые слова.
5. Для того, чтобы организовать поиск словосочетания, например, "текстовый редактор", заключите его в кавычки.

6. После сделанных выборов режимов поиска щелкните мышкой кнопку **Найти**.

Для того, чтобы ограничить список файлов или папок, одновременно задавайте условия на вкладках **Дата** и **Дополнительно**. Если на вкладке **Дата** выбран параметр **Все файлы**, то на жестком диске проводится поиск файлов, удовлетворяющих условиям, заданным на вкладках **Имя и местоположение** и **Дополнительно**.

СПОСОБЫ ЗАЩИТЫ ФАЙЛОВ

Защита применяется для предотвращения использования компьютера или его файлов посторонним пользователем, который не должен, например, знать содержимое определенного файла или вообще не имеет права включить компьютер. Существует несколько различных способов защиты файлов и самого компьютера от несанкционированного доступа.

Архивация файлов

Архивация обычно производится на внешний магнитный носитель, например, магнитную ленту и позволяет сохранить копии нужных файлов. На самом компьютере эти файлы можно стереть, сохранив их, тем самым, от несанкционированного доступа. В дальнейшем можно использовать такие файлы непосредственно с магнитного носителя или снова восстановить их на компьютер. Для запуска программы архивации нужно выполнить следующие действия:

1. Нажать кнопку **Пуск** и выбрать в **Главном меню** раздел **Программы**.
2. Перейдите в подраздел **Стандартные**, а затем выберите пункт **Служебные**.
3. Щелкните пункт **Архивация данных** и выполняйте выходящие на экран инструкции.

Если команда **Архивация данных** отсутствует в меню **Служебные**, это означает, что программа архивации на компьютере не установлена и ее нужно установить, как один из компонент системы.

Использование паролей

Пароли используются для защиты компьютера и файлов от нежелательного, несанкционированного доступа посторонним человеком. Возможно задание Пароля для входа в систему при ее загрузке, выходе из спящего режима, для экранной заставки или для файлов с документами редактора Word и других программ MS Office. В некоторых других прикладных программах также имеется возможность задания Пароля.

Парольная защита компьютера в спящем режиме

Для установки этого режима последовательно выполните следующие действия:

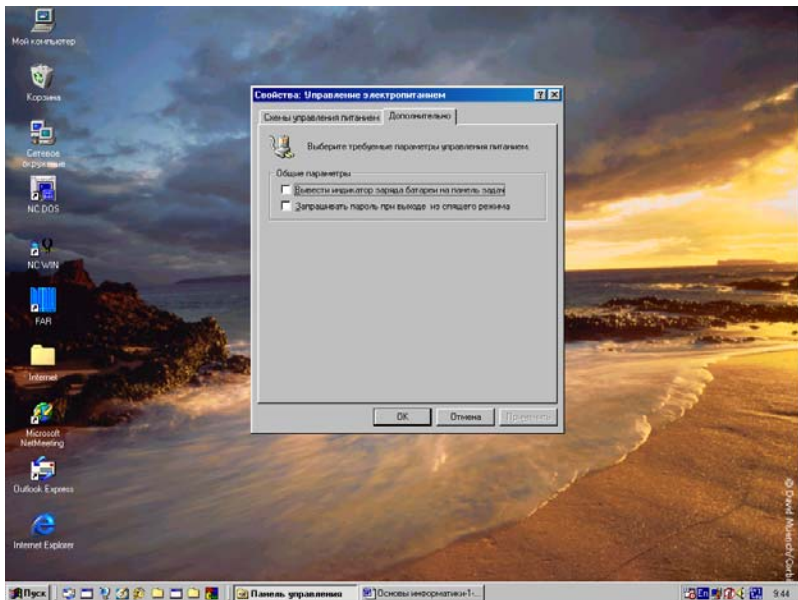


Рис.80. Окно установки пароля при выходе из спящего режима.

1. Откройте диалоговое окно **Свойства: Управление электропитанием**, щелкнув мышкой по значку **Управление электропитанием** в окне **Панель управления** значка **Мой компьютер**.
2. Выберите вкладку **Дополнительно** и установите флажок **Запрашивать пароль при выходе из спящего режима** (рис.80).
3. Затем щелкните левой кнопкой мышки кнопку **ОК** открытого на экране окна.

Назначения пароля экранной заставке

Для задания Пароля экранной заставке выполните, приведенные ниже действия:

1. Откройте вкладку **Заставка** в диалоговом окне **Свойства: Экран**, которое можно вызвать через Контекстное меню в разделе **Свойства**.
2. В поле со списком **Заставка** выберите определенное имя экранной заставки.

3. Установите флажок в меню **Пароль** и щелкните мышкой кнопку **Изменить**.

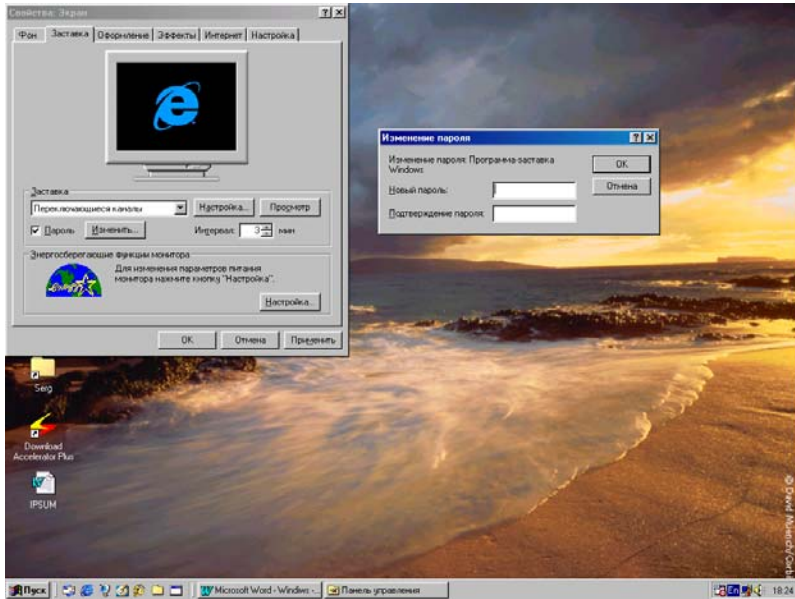


Рис.81. Окно установки пароля экранной заставки.

4. В новом окне введите Пароль, а затем повторно определите его для подтверждения и щелкните кнопку **ОК** (рис.81).

5. После всех сделанных выше установок щелкните мышкой кнопку **ОК** в окне **Свойства: Экран**.

*Назначение пароля при входе
в систему*

Пароль может задаваться при начальной установке системы, при первом ее запуске. Кроме того, можно устанавливая или меняя Пароль с помощью окна **Пароли** в папке **Панель управления**.

Для изменения Пароля определенного пользователя выполните следующие действия:

1. Щелкните мышкой кнопку **Пуск** и в **Главном меню** последовательно выберите команды **Настройка** и **Панель управления**.
2. Откройте окно **Пароли**, в котором нужно выбрать определенного пользователя.

3. Введите старый Пароль пользователя, а затем новый, который будет использоваться в дальнейшем при каждом входе в Windows.

Если при установке системы пользователи и их Пароли определены не были, то задать или изменить Пароль в процессе работы нельзя. Сначала с помощью значка **Пользователи** окна **Панель управления** нужно задать хотя бы одного пользователя и только затем можно задать и менять его Пароль.

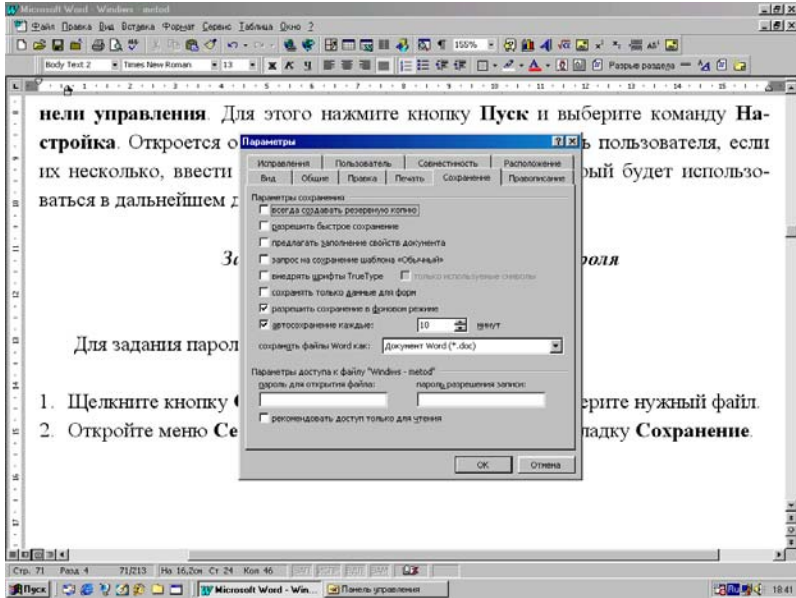


Рис.82. Установка пароля документа Word.

Защита файлов назначением пароля документу Word

Для задания пароля документу редактора выполните следующие действия:

1. Щелкните кнопку **Открыть** **Панели инструментов** редактора и выберите нужный файл.
2. Откройте меню **Сервис**, выберите пункт **Параметры** и перейдите на вкладку **Сохранение**.

3. Если в разделе **Пароль для открытия файла** установить выбранный вами Пароль, то в следующий раз открыть такой файл кнопкой **Открыть** или каким - то другим образом не удастся без ввода заданного Пароля.

4. Если установить Пароль в разделе **Пароль разрешения записи**, то файл можно открыть, но будет закрыт доступ для внесения какие - либо изменений без задания Пароля (рис.82).

5. Если установить флажок в пункте **Рекомендовать доступ только для чтения**, то можно будет открыть файл, вносить изменения, но сохранить их можно только под другим именем файла. В файле с данным именем такие изменения сохранить не удастся.

МУЛЬТИМЕДИА

К мультимедиа относятся дополнительные программы и различные возможности самой системы Windows, используемые для воспроизведения и работы со звуковыми и видео файлами на компьютере.

Сопоставление звуковых сигналов событиям в программах

Для того, чтобы каждое событие при работе с прикладными программами и окнами системы сопровождалось определенным звуковым сигналом нужно выполнить следующие установки:

1. Откройте диалоговое окно **Свойства: Звук**, щелкнув значок **Звук** в папке **Панель управления** окна **Мой компьютер**.
2. В списке **События** выберите определенное действие, которому назначается звуковой сигнал.
3. В поле со списком **Файл** выберите звуковой файл, который будет исполняться при каждом возникновении этого события (рис.83).
4. Если нужный файл звука в списке отсутствует, нажмите кнопку **Обзор** и найдите файл, который требуется подключить к системе.
5. После подключения звукового файла можно прослушать его с помощью кнопки **Проба**.
6. После изменения настройки параметров звука следует их сохранить, используя кнопку **Сохранить как** и ввести имя новой звуковой схемы. Указанное вами имя появится в поле со списком **Схемы**.
7. После всех сделанных установок щелкните мышкой кнопку **ОК**.

Для открытия диалогового окна **Свойства: Звук** можно также нажать кнопку **Пуск**, выбрать в **Главном меню** команды **Настройка** и **Панель управления**, а затем дважды щелкнуть значок **Звук**.

Регулировка уровня воспроизведения

Регулировать громкость звука можно не только с помощью регуляторов уровня воспроизведения на колонках или наушниках, но и программными средствами, которые предоставляет система Windows.

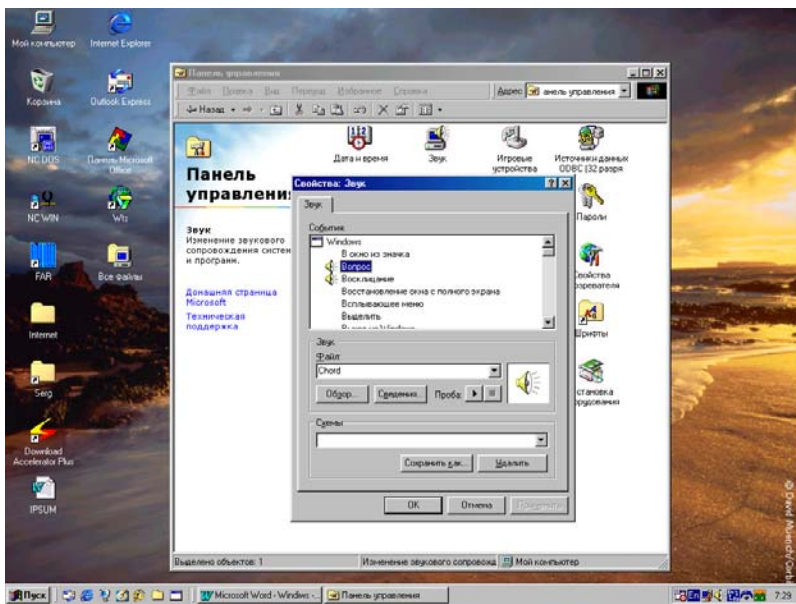


Рис.83. Установка звуков Windows.

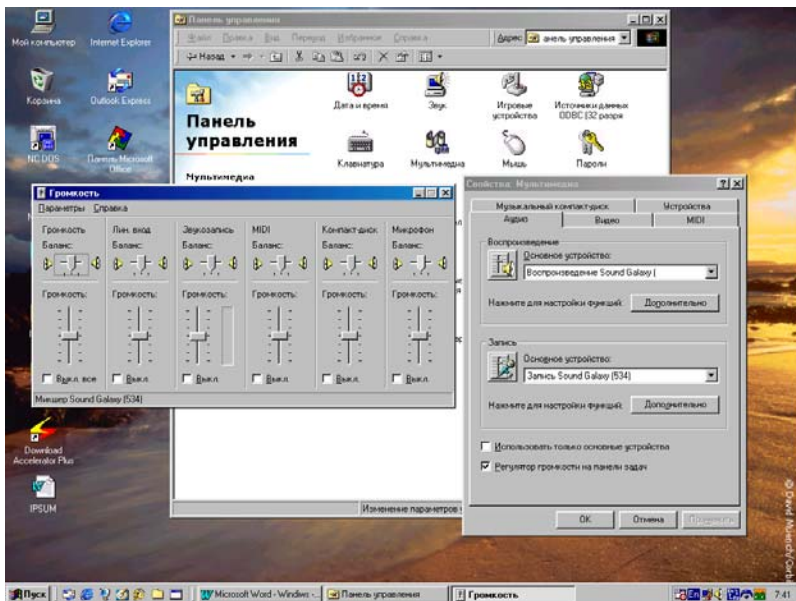


Рис.84. Установка уровня воспроизведения звука.

стью, справа на **Панели задач** появляется значок динамика. Для того, чтобы изменить громкость звука, достаточно дважды щелкнуть этот значок и переместить бегунок нужного регулятора.

В окне **Громкость** можно регулировать и баланс воспроизведения, если на компьютере проигрывается стереозвук. В меню **Параметры** окна **Громкость** можно установить вид панели регуляторов, в частности, можно дополнительно включить регулятор записи через микрофон или линейный вход звуковой платы.

Изменение размеров окна воспроизведения видеоклипов

Средства **Мультимедиа** системы позволяют просматривать на экране компьютера видеоклипы и любое другое видеоизображение, и менять размер окна просмотра. Для изменения окна просмотра последовательно выполните действия:

1. Откройте вкладку **Видео** в диалоговом окне **Свойства: Мультимедиа**, значка **Мультимедиа** **Панели управления** в окне **Мой компьютер**.

2. Щелкните пункт **В окне**, а затем выберите размер окна видео в поле со списком.

3. Для развертывания окна просмотра на весь экран выберите параметр **Во весь экран** (рис.85) раздела **Отображение видео**.

4. После сделанных установок щелкните мышкой кнопку **ОК** открытого окна.

Заметим, что наиболее высокое качество изображения достигается при выборе в меню **В окне** параметра **Исходный размер**.

Для того, чтобы открыть вкладку **Видео** в диалоговом окне **Свойства: Мультимедиа**, можно также нажать мышкой кнопку **Пуск**, выбрать команды **Настройка** и **Панель управления** и дважды щелкнуть значок **Мультимедиа**.

ПЕЧАТЬ ДОКУМЕНТОВ

Перед началом работы с документами системы убедитесь, что принтер правильно подключен к компьютеру, включен и вам известны его название и модель. Для подключения к общему принтеру также необходимо знать сетевой путь (например, \\group\HP-1100).

Если уже имеются установленные принтеры, их значки содержатся в папке **Принтеры** окна **Мой компьютер**. Открыть папку **Принтеры** можно также, нажав кнопку **Пуск** и выбрав команды **Настройка** и **Принтеры**.

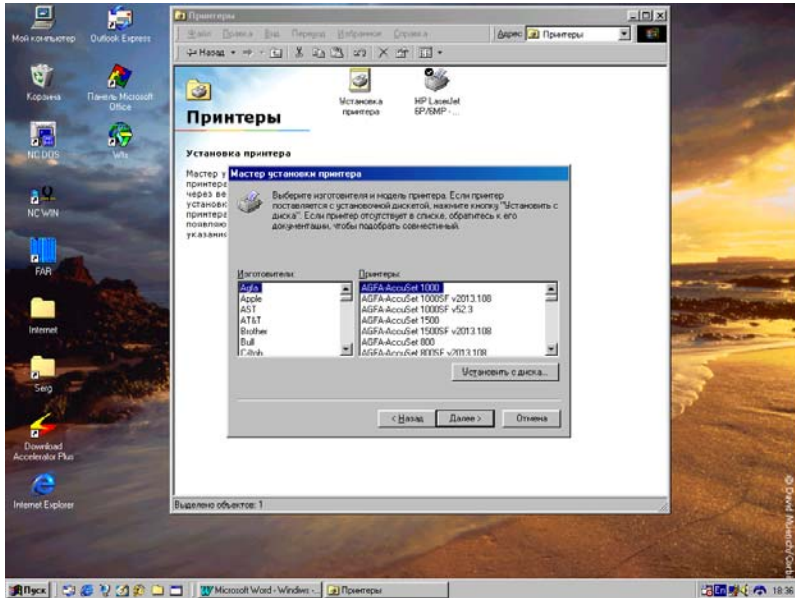


Рис.86. Окно установки принтера в системе Windows.

Установка принтера

Для установки в систему нового принтера выполните следующие действия:

1. Нажмите мышкой кнопку **Пуск**, выберите команды **Настройка** и **Принтеры**, а затем дважды щелкните значок **Установка принтера**.

2. Выполняйте выводящиеся на экран инструкции (рис.86). Выбрав принтер из списка или установив с дискеты нажмите кнопку **Далее**.

3. В конце установки нового принтера щелкните мышкой кнопку **Готово**.

По завершении установки значок принтера появится в папке **Принтеры**. Если требуется напечатать пробную страницу, включите принтер и приведите его в состояние готовности:

1. Щелкните по значку принтера правой кнопкой мыши и в Контекстном меню выберите пункт **Свойства**.

2. Окно свойств для каждого принтера различно, но в любом случае оно будет содержать кнопку **Напечатать пробную страницу**.

Для того, чтобы воспользоваться сетевым принтером, найдите его в папке **Сетевое окружение** на **Рабочем столе**, щелкните значок принтера, чтобы выделить его, а затем выберите в меню **Файл** команду **Установить**.

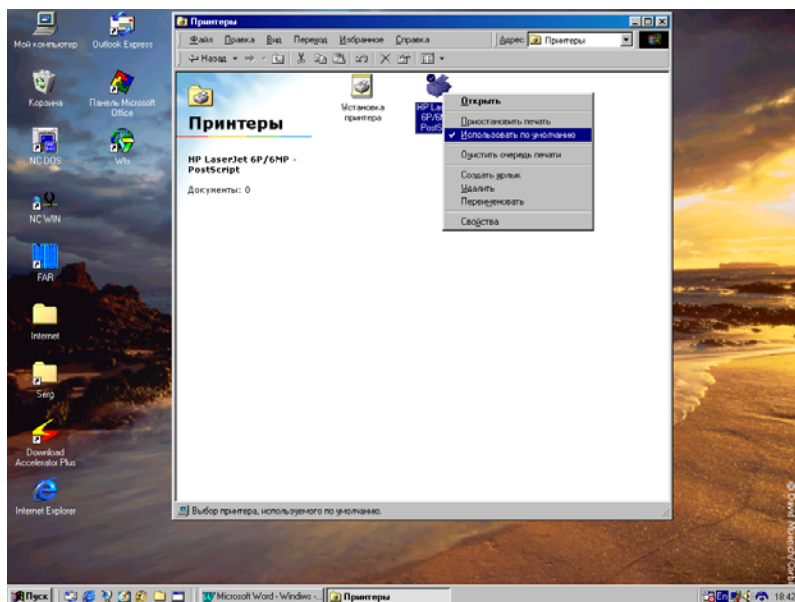


Рис.87. Меню установки принтера "по умолчанию".

Установка принтера "по умолчанию"

Для того, чтобы установить принтер "по умолчанию" необходимо иметь уже установленный на компьютере принтер и выполнить далее следующие действия:

1. На **Панели задач** системы Windows щелкните мышкой кнопку **Пуск**, выберите в **Главном меню** пункт **Настройка**, а затем команду **Принтеры**.

2. Правой кнопкой мыши щелкните значок принтера, который должен использоваться "по умолчанию", а затем в появившемся на экране Контекстном меню щелкните левой кнопкой мыши команду **Использовать "по умолчанию"**, как показано на рис.87.

Если эта команда уже выбрана (отмечена в Контекстном меню "птичкой"), то данный принтер будет в дальнейшем использоваться "по умолчанию", т.е. при каждом обращении к режиму печати.

Печать документов

Если некоторый документ открыт в какой - либо прикладной программе, выберите в меню **Файл** этой программы команду **Печать**. Если документ не открыт, перетащите его значок из папки **Мой компьютер** или окна программы **Проводник** Windows на соответствующий принтер в открытой на экране папке **Принтеры**.

На время печати документа в нижнем правом углу **Панели задач** появится значок принтера. Его исчезновение означает, что печать документа завершена. Для того, чтобы упростить доступ к установленному "по умолчанию" принтеру, можно создать для него **Ярлык на Рабочем столе**.

Для изменение параметров настройки принтера выполните следующие действия:

1. Нажмите кнопку **Пуск**, выберите раздел **Настройка**, а затем пункт **Принтеры**.

2. Щелкните правой кнопкой мыши значок нужного принтера и в появившемся Контекстном меню выберите команду **Свойства**.

Набор доступных параметров существенно зависит от типа используемого принтера. Параметры всегда располагаются на нескольких вкладках (рис.88). Изменение настройки принтера оказывает влияние на все печатающиеся на нем документы.

Для того, чтобы изменить настройку для определенного документа, выберите в меню **Файл** приложения, из которого производит-

ся печать, команду **Макет страницы** или **Настройка принтера** (или эквивалентную команду) и выполните необходимые изменения.

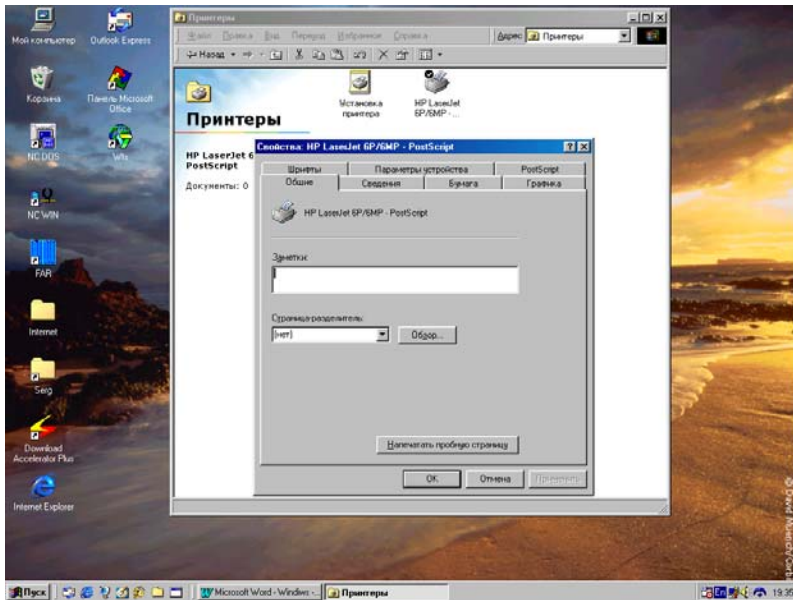


Рис.88. Окно установки параметров принтера.

Назначение общего принтера

При работе в сети вы можете сделать свой принтер (принтер своего компьютера) общим с разрешением доступа к нему других пользователей этой локальной сети.

Для назначения общего сетевого принтера последовательно выполните действия:

1. Нажмите кнопку **Пуск**, выберите команду **Настройка**, а затем раздел **Принтеры**.
2. Щелкните значок принтера, который требуется сделать общим правой кнопкой мыши и в появившемся Контекстном меню выберите команду **Свойства**.
3. Перейдите на вкладку **Доступ**, а затем выберите параметр **Общий ресурс**.

Если вкладка **Доступ** отсутствует, необходимо включить службы общего доступа к файлам и принтерам, щелкнув значок **Сеть** папки **Панель управления**. Более подробно работа принтеров в локальной компьютерной сети будет рассмотрена во второй части книги.

ПРЕОБРАЗОВАНИЕ ДИСКА В FAT 32

Программа преобразования диска позволяет преобразовать диск в файловую систему FAT32, являющуюся расширением формата таблицы размещения файлов (FAT) FAT16. В этом формате данные на жестком диске компьютера сохраняются более эффективно, что позволяет получить до нескольких сотен Мб (Мегабайт) дополнительного места. Кроме того, прикладные программы загружаются быстрее, а компьютер использует меньше системных ресурсов.

После преобразования диска в формат FAT32 использование формата FAT16 становится невозможным. Для этого придется снова выполнить разбиение диска на разделы и переформатировать его. Если преобразован диск, на который установлена операционная система Windows, то после разбиения диска на разделы потребуется переустановка Windows.

Многие старые программы уплотнения диска несовместимы с системой FAT32. Если диск уже сжат, преобразование к FAT32 может оказаться невозможным. Если преобразовать съемный жесткий диск и попытаться использовать его с другой операционной системой, несовместимой с FAT32, то доступ к диску под управлением этой операционной системой будет невозможен.

На компьютере с функцией спящего режима преобразование может привести к отключению этой функции. Поскольку предыдущие версии Windows не совместимы с FAT32, не возможно после преобразования удалить установку Windows - 98 и вернуться, например, к Windows - 95.

Хотя преобразование от FAT16 к FAT32 не влияет на большинство программ, некоторые утилиты работы с дисками, предназначенные для FAT16, не работают с дисками FAT32. При попытке запуска такой утилиты на экран будет выведено соответствующее сообщение.

После преобразования жесткого диска к FAT32 с помощью программы преобразования диска становится не возможным использование режима двойной загрузки для запуска предыдущих версий Windows (Windows - 95, Windows - NT 3.x, Windows - NT 4.0 и Windows - 3.x). Однако можно будет получить доступ к жесткому диску FAT32 из предыдущих версий Windows через локальную сеть.

После преобразования к FAT32 данные на жестком диске нельзя будет сжимать или преобразовывать обратно в FAT16 без использования служебных программ сторонних производителей.

Принцип хранения информации

При сохранении файла или установке новой программы компьютер размещает данные на жестком диске в маленьких областях, на-

зываемых кластерами. Чем меньше размер кластеров, тем эффективнее используется место на диске. Размер кластера зависит от размера раздела, который в свою очередь зависит от используемой файловой системы.

В большинстве компьютеров "по умолчанию" используется только один раздел на жестком диске. Хотя всегда имеется возможность задать любое число таких разделов. Обычно, при большом объеме диска задают разделы размером около 2 - 3 Гб (Один Гигабайт (Гб) = 1000 Мегабайт (Мб), 1 Мб = 1000 Килобайт (Кб), 1 Кб = 1000 байт).

В показанной ниже таблице приведены размеры разделов и кластеров, поддерживаемые файловой системой FAT32. FAT16 не поддерживает разделы более 2 Гигабайт (Гб), а FAT32 не поддерживает разделы менее 512 Мегабайт (Мб).

<i>Размер раздела</i>	<i>Размер кластера FAT16</i>	<i>Размер кластера FAT32</i>
32 Мб	2 Кб	-
28 Мб	2 Кб	-
256 Мб	4 Кб	-
512 Мб	8 Кб	4 Кб
1 Гб	16 Кб	4 Кб
2 Гб	32 Кб	4 Кб
3 Гб - 7 Гб	-	4 Кб
8 Гб - 16 Гб	-	8 Кб
16 Гб - 32 Гб	-	16 Кб
Более 32 Гб	-	32 Кб

При размерах жесткого диска менее 2 Гб преобразование в файловую систему FAT32 может не дать заметного увеличения производительности по сравнению с FAT16. Если размер диска составляет от 2 Гб до 2 Терабайт (1 Терабайт (Тб) = 1000 Гб), то для повышения эффективности использования его можно преобразовать в FAT32. По сравнению с FAT16 файловая система FAT32 имеет следующие преимущества:

1. Прикладные программы запускаются быстрее - разница в скорости может составить до 50%.
2. На жестком диске используется меньший размер кластера, что позволяет эффективнее использовать все дисковое пространство - разница в эффективности может составить до 15%.
3. Возможно форматирование жестких дисков размером до 2 Тб без деления на части - разделы.

4. Возможно перемещение корневого каталога и использование резервных копий FAT, что повышает надежность работы компьютера.

Преобразование файловой системы

Для преобразования файловой системы диска нужно выполнить следующие действия:

1. Нажмите кнопку **Пуск**, выберите пункты **Программы, Стандартные, Служебные** и запустите **Преобразование диска в FAT32**.

2. Ознакомьтесь с приведенными сведениями и нажмите кнопку **Далее**.

3. Из списка **Диски** выберите диск для преобразования и снова нажмите кнопку **Далее**.

4. Появится сообщение о том, что предыдущие версии MS - DOS, Windows - 95 и Windows - NT не смогут получить доступ к дискам FAT32.

5. Нажмите кнопку **ОК**, чтобы продолжить преобразование, или кнопку **Отмена**, чтобы выйти из мастера преобразования дисков.

6. Мастер преобразования дисков выполнит поиск антивирусных и служебных дисковых программ, несовместимых с FAT32 (рис.89).

7. При обнаружении несовместимых программ нажмите кнопку **Сведения**. Если несовместимые программы не обнаружены, нажмите кнопку **Далее**.

8. Для того, чтобы создать резервную копию файлов перед преобразованием в FAT32, щелкните мышкой кнопку **Создать копию**.

9. Если программа Backup Windows не установлена, будет предложено установить ее.

Для продолжения преобразования диска дважды нажмите кнопку **Далее**. Компьютер перезагрузится и начнется преобразование. На больших дисках или на дисках, имеющих мало свободного места, дефрагментация и преобразование может длиться час и более.

После выполнения преобразования диска появится сообщение об удачном или неудачном его завершении. Если преобразование прошло успешно, нажмите кнопку **Готово**. Тогда преобразование диска будет полностью завершено. Если преобразование не было успешно выполнено, на экране появится сообщение с указанием причины.

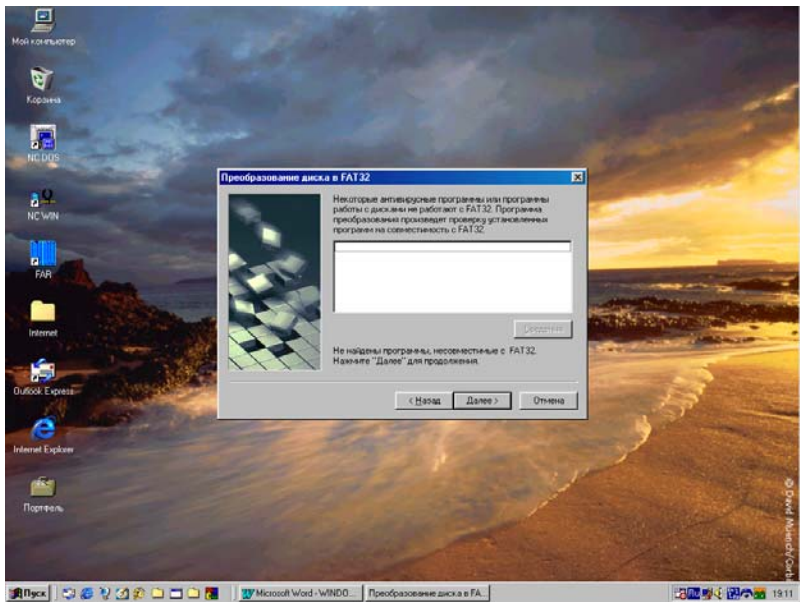


Рис.89. Преобразование диска в FAT32.

Установленная на компьютере антивирусная программа может обнаружить изменения в таблице разделов и загрузочной записи и предложить "восстановить" их. Этого делать не следует. В противном случае загрузочная запись или таблица разделов компьютера будут изменены, а диск и все имеющиеся на нем данные могут оказаться недоступными.

СТАНДАРТНЫЕ ПРОГРАММЫ

Данный раздел включает в себя наборы некоторых стандартных программ Windows, которые могут быть установлены на компьютер вместе с установкой самой системы.

Игры Windows

Для запуска одной из игр системы Windows выполните следующие действия:

1. Нажмите кнопку **Пуск** и последовательно выберите разделы **Программы, Стандартные и Игры**.

2. В открывшемся меню **Игры** выберите нужную игру, щелкнув по ее названию. Пример одной из карточных игр Windows показан на рис.90.

Для получения сведений о правилах конкретной игры обращайтесь к меню **Справка** этой программы.

Если некоторая игра или весь раздел **Игры** отсутствуют в меню **Стандартные**, это значит, что они не были установлены при установке Windows на данный компьютер и их следует доустановить.

Калькулятор

Программа **Калькулятор**, используемый для выполнения различных арифметических и логических вычислений, имеет два режима работы:

1. Обычный, предназначенный для простейших арифметических вычислений.

2. Инженерный, который обеспечивает доступ ко многим математическим (в том числе логическим и статистическим) функциям.

Для запуска программы **Калькулятора** выполните следующие действия:

1. Нажмите кнопку **Пуск** и по очереди выберите разделы **Программы и Стандартные**, а затем щелкните команду **Калькулятор**. Окно программы показано на рис.91.

2. Для получения дополнительных сведений об использовании **Калькулятора** в различных режимах работы обращайтесь к меню **Справка** этой программы.

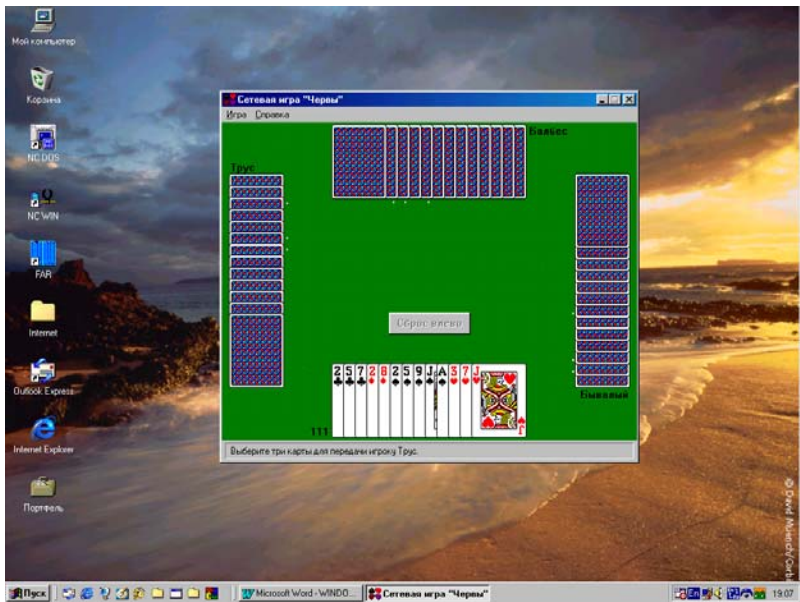


Рис.90. Игра "Червы".

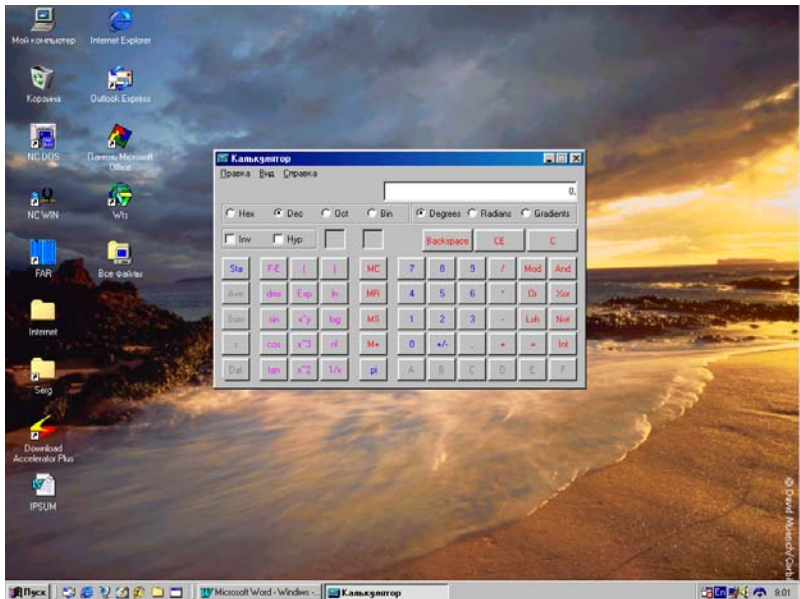


Рис.91. Окно Калькулятора Windows.

Все действия здесь выполняются, как на обычном калькуляторе. На кнопки окна калькулятора можно нажимать, щелкая по ним левой кнопкой мышки или включить на клавиатуре клавишу Num Lock (чтобы горела соответствующая лампочка) и использовать цифровую клавиатуру.

В этом случае клавиша Back Space служит для стирания последнего введенного символа, а Esc для очистки всего окна ввода цифр. Действия выполняются в следующем порядке:

1. Наберите на клавиатуре первое число, которое будет показано на экране калькулятора.
2. Выберите знак арифметического действия (+, -, /, *) или оператор (Sin, Cos, Tan и т.д.).
3. Введите второе число, которое также показывается в окне программы. При выборе математического оператора, а не знака арифметического действия этот пункт выполнять не следует.
4. Щелкните мышкой знак равенства (=) в окне калькулятора или нажмите Enter на клавиатуре.
5. На экране калькулятора вы получите результат, выполненных вычислений.

Калькулятор, показанный на рис.91, позволяет выполнять достаточно сложные инженерные вычисления. Можно находить значения различных тригонометрических функций и логарифмов, использовать возможности памяти калькулятора и применять в вычислениях логические операции.

Графический редактор Paint

Графический редактор **Paint** позволяет создавать, изменять и просматривать рисунки. Рисунок, созданный в **Paint**, можно вставить в другой документ или использовать, как фоновый рисунок **Рабочего стола** системы. **Paint** позволяет также просматривать и редактировать фотографии, снятые сканером. Для запуска графического редактора **Paint** нужно выполнить следующие действия:

1. Нажать кнопку **Пуск**, выбрать разделы **Программы** и **Стандартные**, а затем команду **Paint**. Окно программы после ее запуска показано на рис.92.
2. Слева вверху окна находится **Панель инструментов**. Щелкнув левой кнопкой мышки по некоторой кнопке вы выбираете определенный инструмент для рисования различных фигур и линий или стирания части рисунка ластиком.

3. Слева внизу - **Палитра цветов**. Выбор цвета выполняется щелчком мышкой. Самый левый цветной квадратик показывает текущий - выбранный в данный момент цвет.

Пользователь имеет возможность отменить при рисовании три последних изменения, выбирая в меню **Правка** команду **Отменить** для каждого изменения.

Для получения дополнительных и более подробных сведений об использовании графического редактора **Paint** обращайтесь к меню **Справка** этой программы.

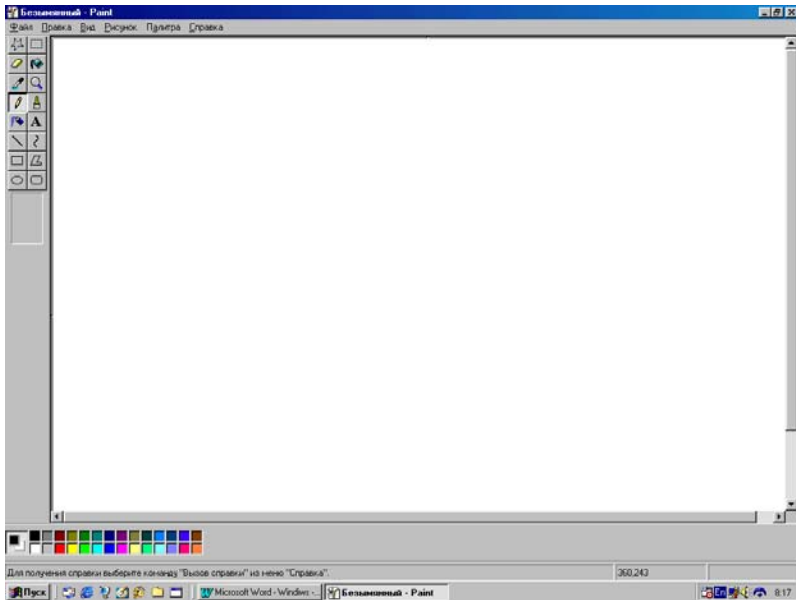


Рис.92. Окно графического редактора **Paint**.

Текстовый редактор **Word Pad**

Текстовый редактор **Word Pad** предназначен для работы с небольшими файлами и поддерживает форматирование документов с использованием различных шрифтов и стилей абзаца, установленных в системе. Редактор позволяет работать с текстовыми файлами, сохраняя их в кодировке **Текст Windows**. Однако, его возможности намного меньше, чем у редактора **Word - 97** и использовать его рекомендуется только при отсутствии на компьютере основного редактора **Word**.

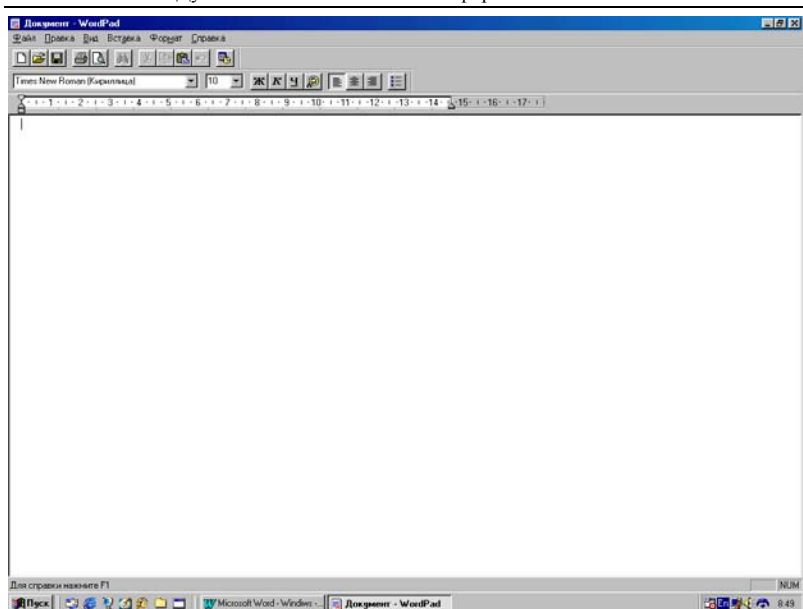


Рис.93. Окно редактора **Word Pad**.

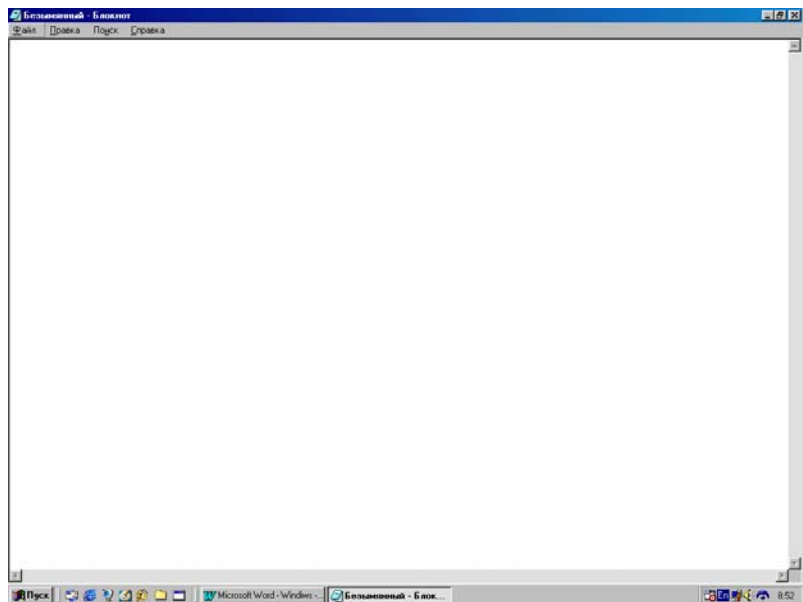


Рис.94. Окно редактора **Блокнот**.

Для запуска редактора **Word Pad** щелкните мышкой кнопку **Пуск**, выберите в **Главном меню** пункт **Программы**, а затем раздел **Стандартные**. Окно программы показано на рис.93.

Для получения дополнительных сведений об использовании редактора **Word Pad** обращайтесь к **Справке** этой программы.

Текстовый редактор Блокнот

Блокнот предназначен для создания и редактирования текстовых файлов, не требующих форматирования и не превышающих по размеру 64 Кб. Программа **Блокнот** работает только с файлами в формате **Текст** системы Windows.

Если файл требует форматирования текста или он превышает по размеру 64 Кб, следует использовать текстовый редактор **Word Pad** или основной редактор **Word**.

Для запуска **Блокнота** нужно нажать кнопку **Пуск**, выбрать разделы **Программы** и **Стандартные**, а затем команду **Блокнот** (рис.94). Для получения дополнительных сведений об использовании **Блокнота** обращайтесь к меню **Справка** этой программы.

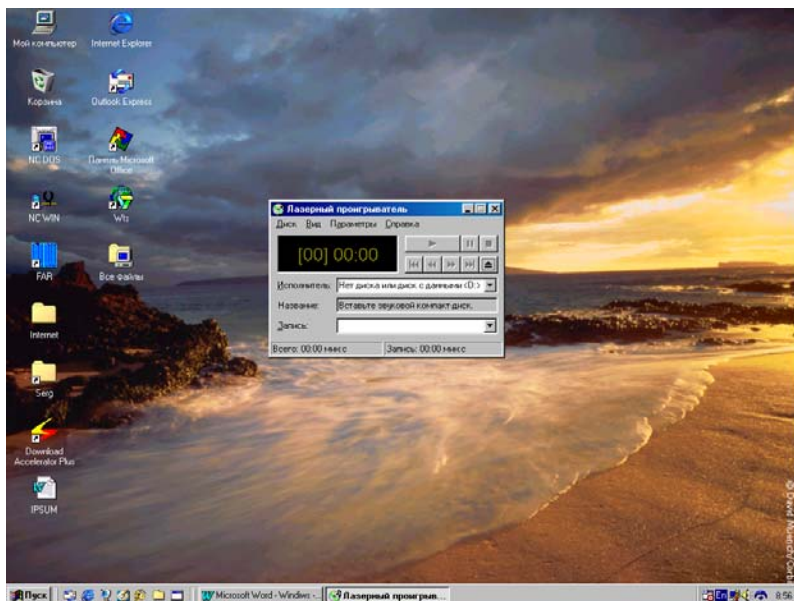


Рис.95. Окно программы **Лазерного проигрывателя**.

Лазерный проигрыватель

Лазерный проигрыватель предназначен для воспроизведения музыкальных и любых аудио компакт - дисков с помощью специального устройства для чтения компакт - дисков (CD - ROM). Прослушать компакт - диск можно через наушники, непосредственно подключенные к дисководу компакт - дисков.

Если на компьютере установлена звуковая плата (карта), становится возможным воспроизведение компакт - диска через внешние динамики - колонки. При установке на CD - ROM музыкального диска с режимом Автозапуска, программа проигрывателя запускается автоматически. Все кнопки в окне программы аналогичны кнопкам управления на обычном магнитофоне.

Для запуска **Лазерного проигрывателя** нужно нажать кнопку **Пуск**, в **Главном меню** выбрать разделы **Программы**, **Стандартные** и **Развлечения**, а затем щелкнуть команду **Лазерный проигрыватель** (рис.95). Для получения дополнительных сведений об использовании **Лазерного проигрывателя** обращайтесь к меню **Справка** этой программы.

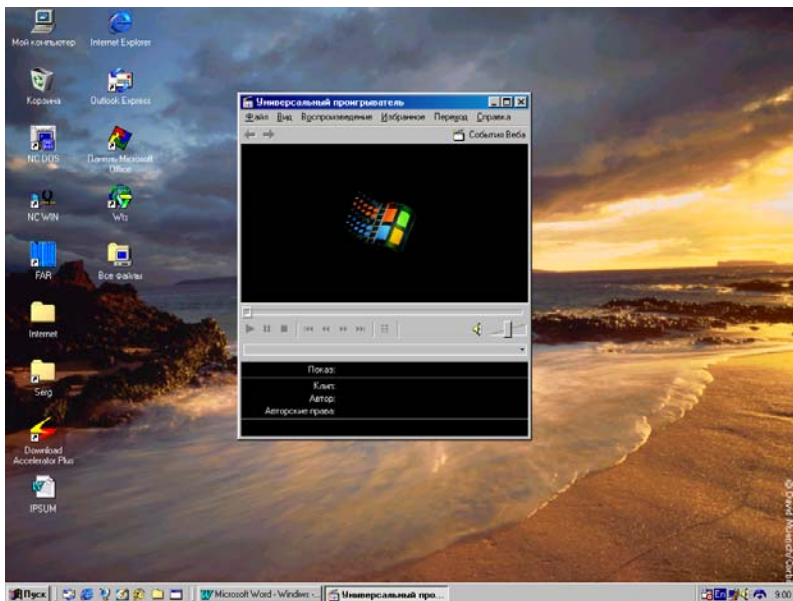


Рис.96. Окно **Универсального проигрывателя**.

Универсальный проигрыватель

Универсальный проигрыватель предназначен для воспроизведения аудио-, видеозаписей и программ мультимедиа. Для использования программы **Универсального проигрывателя** необходимо наличие на компьютере звуковой платы (карты) и дополнительных внешних колонок.

Для запуска **Универсального проигрывателя** можно нажать кнопку **Пуск**, в **Главном меню** выбрать разделы **Программы**, **Стандартные** и **Развлечения**, а затем щелкнуть команду **Универсальный проигрыватель** (рис.96). Все кнопки в окне программы аналогичны кнопкам на обычном видеомаягнитофоне.

Для получения дополнительной информации об использовании **Универсального проигрывателя** обращайтесь к меню **Справка** этой программы.

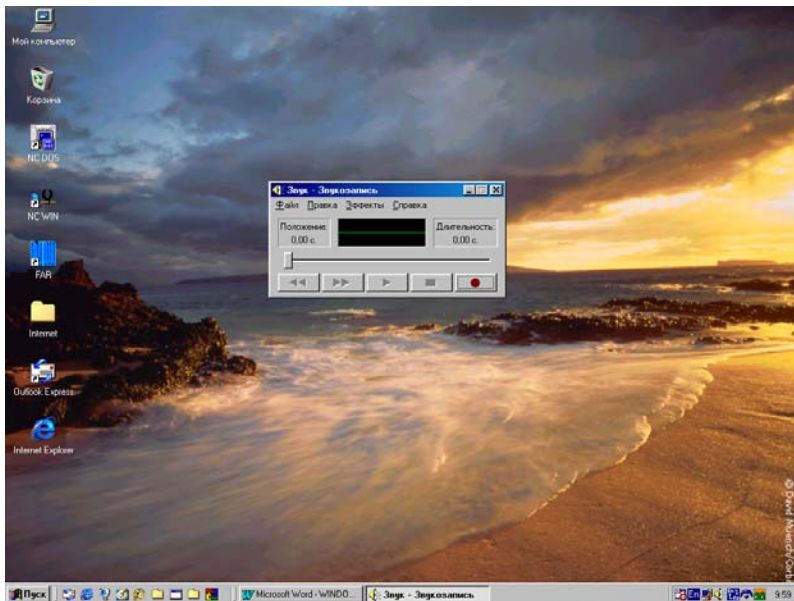


Рис.97. Программа **Звукозаписи**.

Программа звукозаписи

Программа **Звукозапись** используется для записи, воспроизведения и редактирования звуковых файлов. Для работы этой программы требуется наличие звуковой платы и подключенных к компьютеру

ру динамиков. Для записи живого звука также требуется микрофон. Кнопки управления записью и проигрывания записанного файла аналогичны кнопкам обычного магнитофона. Окно программы **Звукозаписи** показано на рис.97.

Для запуска программы **Звукозапись** нужно нажать кнопку **Пуск**, последовательно открыть пункты **Программы**, **Стандартные** и **Развлечения**, а затем выбрать команду **Звукозапись**.

Для получения дополнительных сведений об использовании режима **Звукозаписи** обращайтесь к разделу **Справка** Главного меню этой программы.

Программа Kodak Imaging

Программа **Kodak Imaging** позволяет просматривать, аннотировать, редактировать и выполнять все основные задачи обработки графических документов, включая файлы факсов и отсканированных изображений любого формата и цвета.

Программа дает возможность вырезать или копировать части рисунков многих форматов, менять характеристики изображения, преобразовывать один формат рисунка в другой и обладает многими другими полезными возможностями.

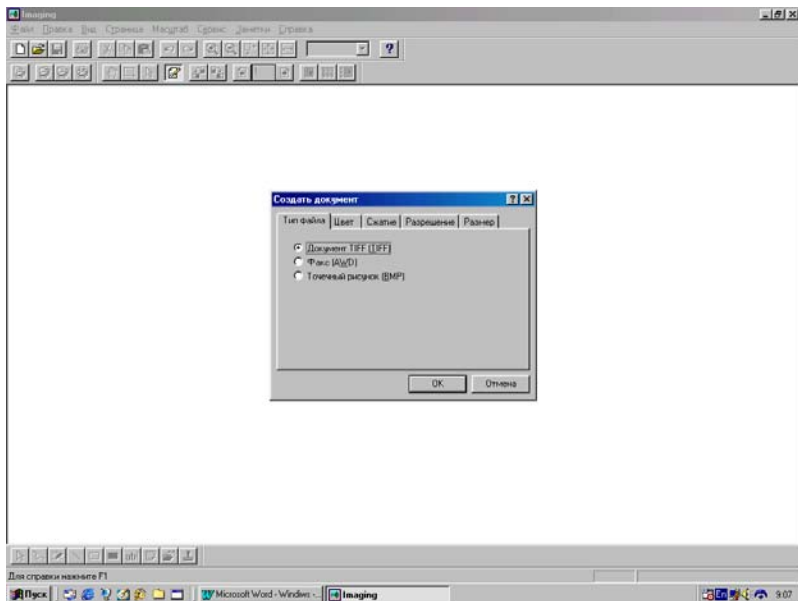


Рис.98. Окно графического редактора **Imaging**.

На рис.98 показано основное окно этой программы с окном **Создания документа**, которое позволяет установить тип нового документа для создания или редактирования графического файла.

Для запуска программы **Kodak Imaging** можно нажать кнопку **Пуск**, в **Главном меню** последовательно выбрать разделы **Программы** и **Стандартные**, а затем щелкнуть мышкой команду **Imaging**.

Для получения дополнительных сведений об использовании **Imaging** обращайтесь к меню **Справка** этой программы.

Использование Портфеля

Портфель удобен при работе с несколькими копиями одних и тех же документов на разных компьютерах. Он позволяет, например, согласовать между собой копии документов на переносном и настольном компьютерах.

Все эти действия, впрочем, можно выполнить и простым копированием документов с переносного компьютера на стационарный с помощью некоторого файлового менеджера, например, программы **Проводник** системы Windows.

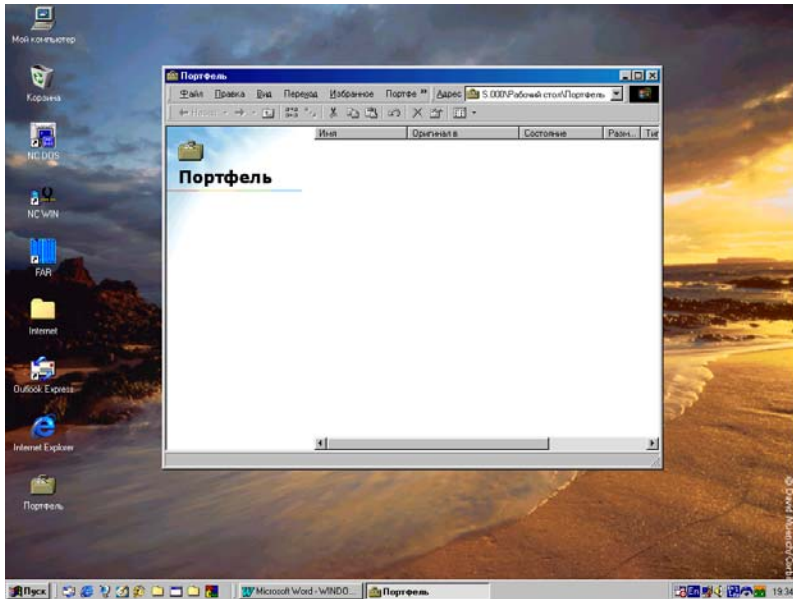


Рис.99. Окно программы **Портфеля**.

Синхронизация файлов на двух компьютерах

Для синхронизации файлов последовательно выполните следующие действия (рис.99):

1. Соедините оба компьютера специальным шнуром или они должны быть подключены в одну локальную сеть.
2. Скопируйте файлы из общих папок настольного компьютера в **Портфель** переносного компьютера.
3. Теперь переносной компьютер можно отсоединить от основного и работать с файлами автономно.
4. Завершив работу с файлами, снова соедините два компьютера и дважды щелкните значок **Портфель**.
5. В меню **Вид** открытого окна **Портфель** выберите команду **Обновить все**.
6. Для обновления только выбранных файлов, выделите нужные файлы, щелкая по ним мышкой и удерживая нажатой клавишу Ctrl и выберите команду **Обновить выделенные объекты**.

Синхронизация файлов гибким диском

Для синхронизации файлов с помощью дискеты последовательно выполните действия:

1. Вставьте гибкий диск в дисковод основного, стационарного компьютера.
2. Скопируйте нужные файлы из рабочих папок в **Портфель** основного компьютера.
3. Перетащите мышкой **Портфель** на гибкий диск, удерживая нажатой ее левую кнопку.
4. Вставьте гибкий диск в переносной компьютер и начните работу с файлами в папке **Портфель**.
5. Для синхронизации версий файлов снова вставьте гибкий диск, содержащий **Портфель**, в основной компьютер и дважды щелкните значок **Портфель**.
6. В меню **Вид** открытого окна **Портфель** выберите команду **Обновить все**.
7. Для обновления только выбранных файлов, выделите нужные файлы и щелкните мышкой команду **Обновить выделенные объекты**.

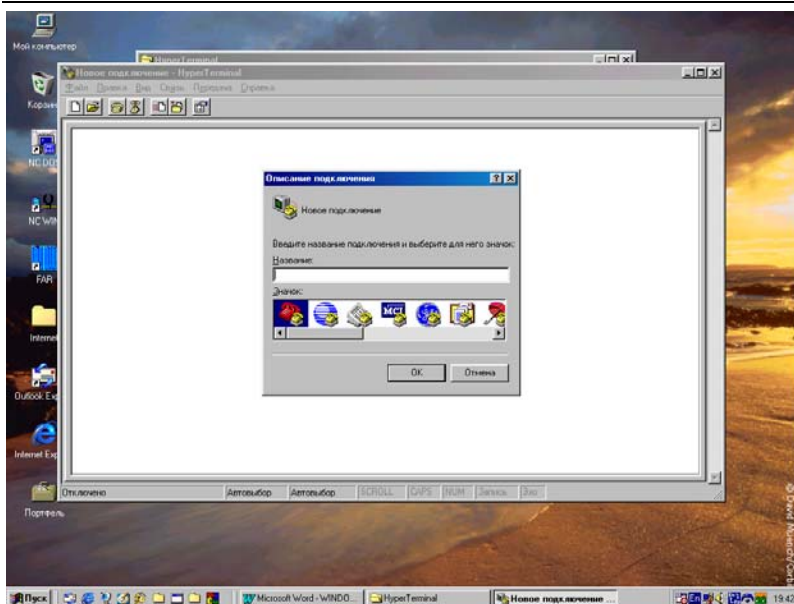


Рис.100. Окно программы **Hyper Terminal**

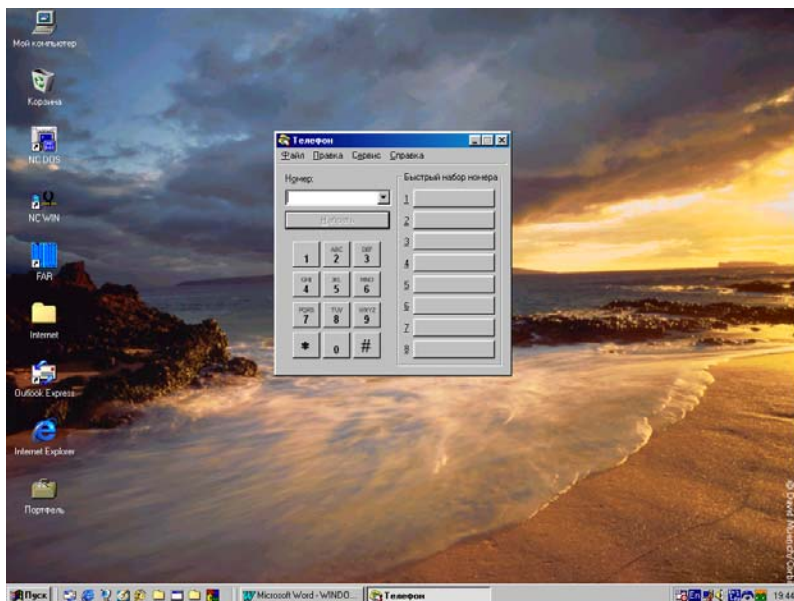


Рис.101. Окно программы номеронабирателя.

Программа **Hyper Terminal**

Программа связи **Hyper Terminal** позволяет установить с помощью модема по телефонной линии связь с удаленным компьютером. Эта программа предназначена для отправки и приема различных файлов, подключения к всевозможным электронным доскам объявлений, различных банкам программ (типа BBS) и многим другим интерактивным службам, имеющимся в сетях.

Для запуска программы **Hyper Terminal** можно нажать кнопку **Пуск** в **Главном меню**, выбрать разделы **Программы**, **Стандартные** и **Связь**, а затем щелкнуть команду **Hyper Terminal** (рис.100).

Для получения более подробных сведений о работе **Hyper Terminal** смотрите раздел **Справка** этой программы.

Однако, для того, чтобы организовать свободный, не ограниченный (при отсутствии паролей на вход в сеть) доступ к файлам, папкам и принтерам другого компьютера, работающего под управлением Windows, рекомендуется использовать не программу **Hyper Terminal**, а систему удаленного доступа к локальной сети (если такая сеть существует), которая будет описана в следующей части книги.

Набор телефонного номера

Программа **Телефон** предназначен для набора телефонных номеров с помощью присоединенного к компьютеру модема или другого аналогичного устройства.

Для запуска программы **Телефон** можно нажать кнопку **Пуск**, выбрать разделы **Программы**, **Стандартные** и **Связь**, а затем щелкнуть команду **Телефон** (рис.101).

Для набора номера достаточно щелкнуть мышкой по нужным цифрам и нажать кнопку **Набрать** или использовать кнопки памяти, в которую предварительно занесен нужный номер. Память программы позволяет запоминать до восьми избранных телефонных номеров для их последующего быстрого набора.

СЛУЖЕБНЫЕ ПРОГРАММЫ

Служебные программы устанавливаются на компьютер вместе с установкой самой системы и служат, в основном, для обслуживания других программ, файлов и папок, и самого компьютера в процессе их работы.

Таблица символов

Таблицу символов используют для вставки специальных символов в документ, например, редактора **Word Pad**. Таблица символов может использоваться совместно лишь с приложениями (другими программами), работающими на основе Windows. Текстовый редактор Word имеет свою собственную **Таблицу символов**, используемую в его документах.

Для запуска **Таблицы символов** можно нажать кнопку **Пуск**, выбрать в **Главном меню** разделы **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть команду **Таблица символов** (рис.102).

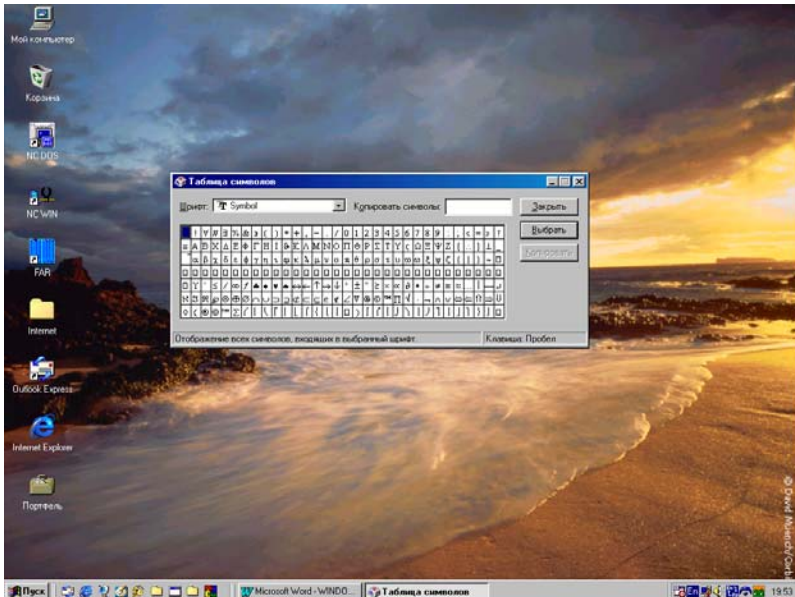


Рис.102. Таблица символов.

Для вставки символа в позицию экранного курсора некоторого приложения Windows нужно щелкнуть мышкой по выбранному сим-

волу, а затем по кнопке **Вставить**. Для выключения окна вставки символов на экране следует нажать кнопку **Заккрыть**.

Окно буфера обмена

Окно **Буфера обмена** используют для просмотра на экране содержимого буфера обмена - определенной области оперативной памяти системы Windows. В любом приложении системы можно вставить содержимое буфера обмена в позицию экранного курсора нажатием кнопки **Вставить**. Для помещения информации в буфер следует использовать кнопки **Копировать** или **Вырезать**.

Для запуска окна **Буфера обмена** можно нажать кнопку **Пуск** на **Панели задач**, выбрать разделы **Программы**, **Стандартные** и **Службные**, а затем щелкнуть команду **Буфер обмена** (рис.103).

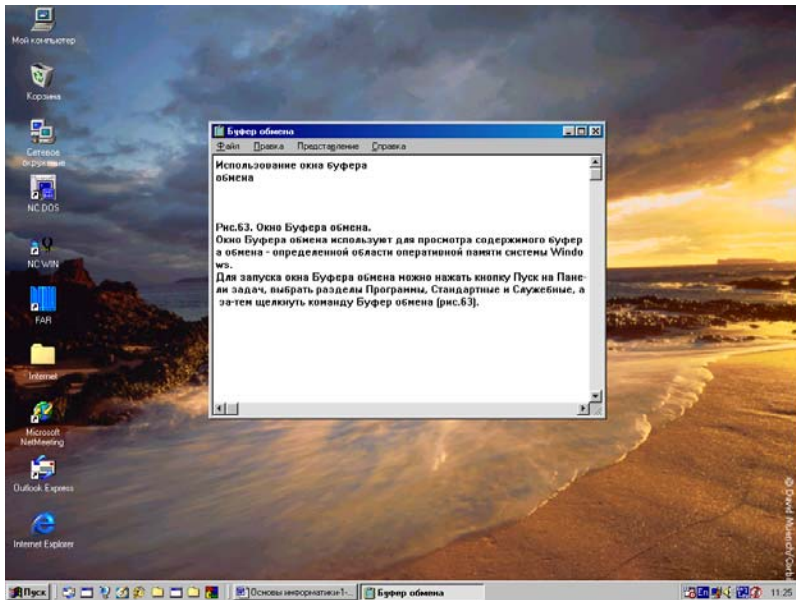


Рис.103. Окно **Буфера обмена**.

Программа очистки диска

Очистка диска позволяет освободить место на жестком диске, стирая все не нужные и временные файлы на компьютере. Программа **Очистки диска** проверяет диск и выводит на экран перечень времен-

ных файлов, файлов, загружаемых из Интернета, а также не нужных программных файлов, которые можно удалить.

Для запуска программы **Очистки диска** выполните следующие действия:

1. Нажмите кнопку **Пуск**, в **Главном меню** выберите разделы **Программы**, **Стандартные** и **Служебные**, а затем щелкните мышью команду **Очистка диска** (рис.104).

2. После запуска программы появится окно, в котором нужно выбрать диск и щелкнуть кнопку **ОК**.

3. В новом окне, для очистки диска снова нажмите кнопку **ОК** и подтвердите удаление, найденных лишних файлов кнопкой **Да**.

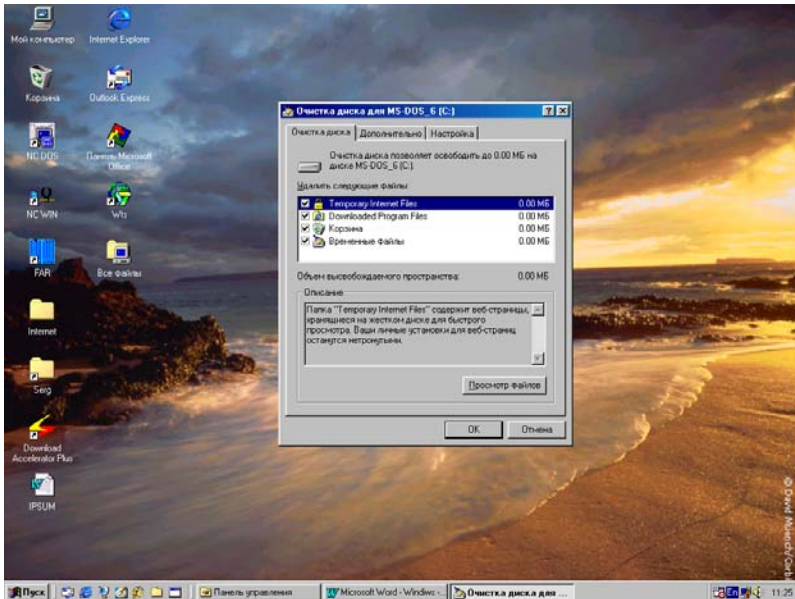


Рис.104. Программа **Очистки диска**.

При этом в окне **Удалить следующие файла** должны быть установлены "птички" в тех разделах диска, которые нужно очистить.

Дефрагментация жесткого диска

Программа **Дефрагментации** позволяет ускорить выполнение программ за счет перераспределения файлов и неиспользуемого объ-

ема на жестком диске. Для запуска программы **Дефрагментации** выполните следующие действия:

1. Нажмите кнопку **Пуск**, в **Главном меню** выберите команды **Программы, Стандартные и Служебные**, а затем команду **Дефрагментация диска**.

2. После запуска появятся окна **Дефрагментация диска** и **Выберите диск**.

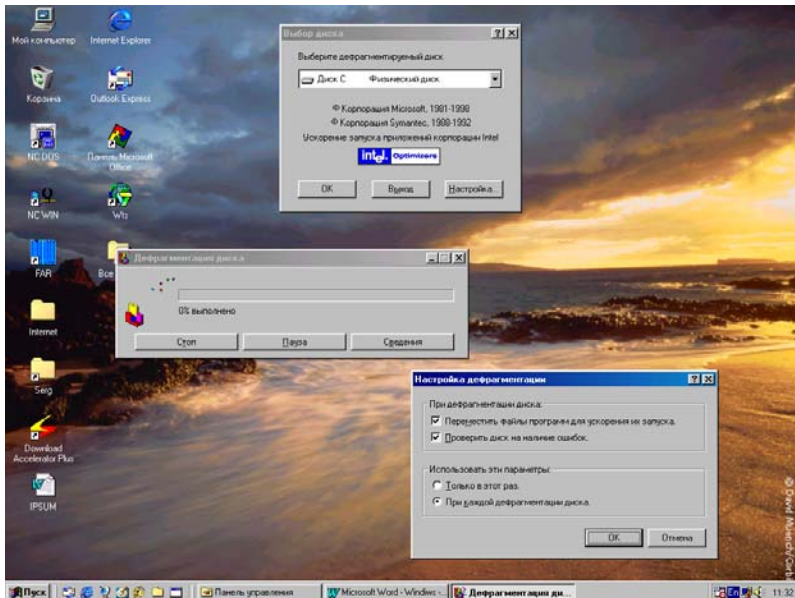


Рис.105. Программа **Дефрагментации жесткого диска**.

3. В последнем окне нужно выбрать диск и щелкнуть кнопку **Настройка**, установить нужные параметры и нажать **ОК**, а затем нажать **ОК** в окне **Дефрагментация диска**.

4. После всех выполненных действий начинает работать программа дефрагментации (рис.105).

Мастер обслуживания

Мастер обслуживания позволяет быстрее выполнять программы, проверять жесткий диск на наличие ошибок и освобождать место на диске. Создав расписание для регулярного выполнения этих слу-

жебных программ, вы сможете добиться максимальной производительности компьютера. Во время выполнения запланированных заданий компьютер должен быть включен.

Для запуска **Мастера обслуживания** можно нажать кнопку **Пуск**, в **Главном меню** выбрать разделы **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть команду **Мастер обслуживания** (рис.106).

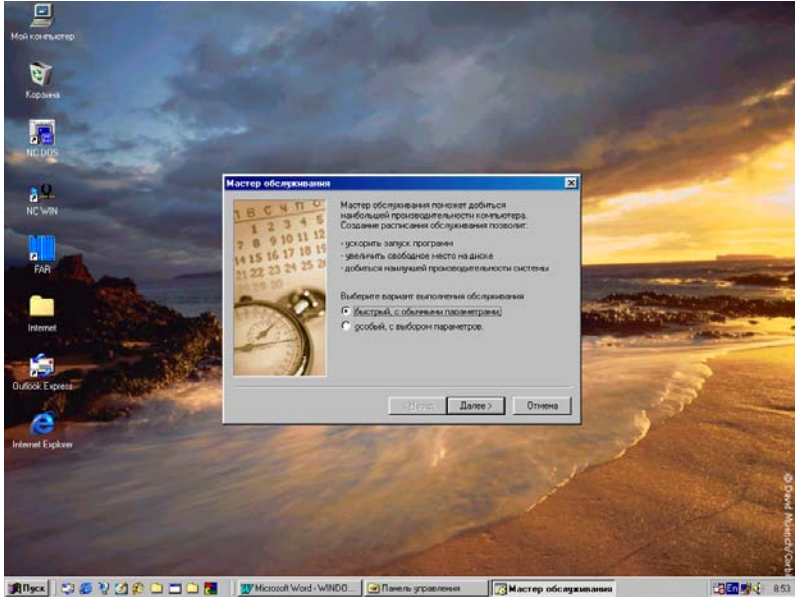


Рис.106. Окно **Мастера обслуживания**.

Индикатор ресурсов

Индикатор ресурсов отображает сведения об использовании системных ресурсов работающими на компьютере программами.

Можно запустить **Индикатор ресурсов**, нажав кнопку **Пуск**, выбрав пункты **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть **Индикатор ресурсов** (рис.107).

Для просмотра подробных сведений об использовании системных ресурсов, щелкните правой кнопкой значок **Индикатор ресурсов** у правого края **Панели задач** и в **Контекстном меню** выберите команду **Сведения**.

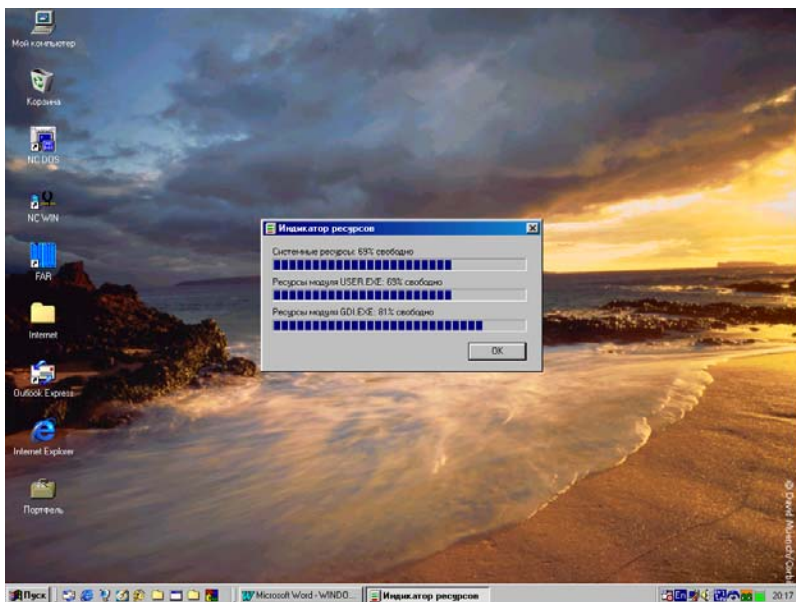


Рис.107. Индикатор системных ресурсов.

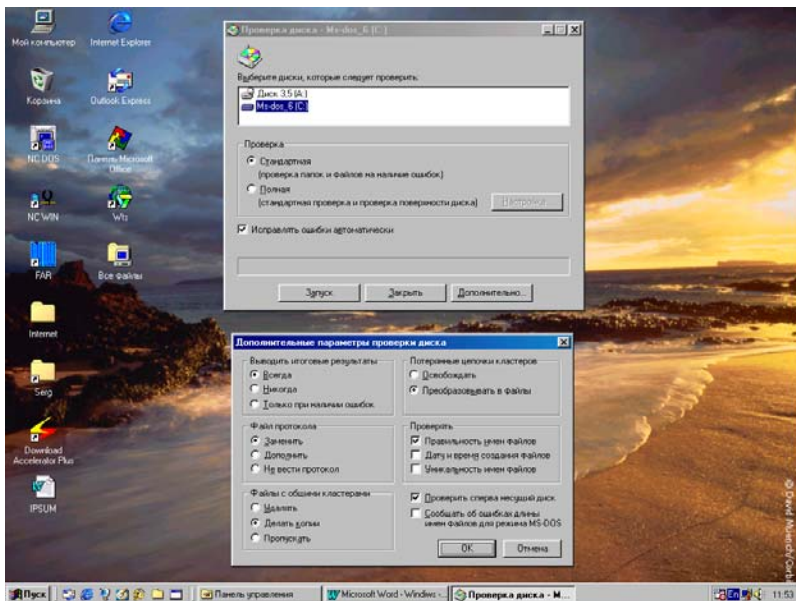


Рис.108. Окно программы Проверки диска.

Программа проверки диска

Программа **Проверки диска** позволяет проверить жесткий диск на наличие логических и физических ошибок. После проверки поврежденные области могут быть автоматически исправлены.

Для проверки диска можно нажать кнопку **Пуск**, выбрать разделы **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть команду **Проверка диска** (рис.108).

Кнопкой **Дополнительно** можно установить дополнительные режимы проверки, окно которых так же показано на рисунке. Затем в окне **Выберите диск** надо указать диск, который следует проверить. В окне **Проверка** обычно выбирают режим **Стандартная** и устанавливают "птичку" в меню **Исправлять ошибки автоматически**. После всех установок нужно щелкнуть кнопку **Запуск**.

В результате проверки выбранного диска на экране появится информация с результатами проверки. После просмотра результатов нужно щелкнуть **ОК** этого окна и кнопку **Закрыть** окна **Проверки диска**.

Программа планировщика заданий

Программа **Планировщик заданий** является средством назначения заданий (таких, как дефрагментация и проверка диска, очистка диска и т.п.) для выполнения их в наиболее удобное время. Планировщик заданий запускается при каждом запуске Windows и выполняется в фоновом режиме. Программа планировщика предоставляет намного более широкие возможности по сравнению с **Мастером обслуживания**, поскольку позволяет автоматически запускать практически любую программу. **Планировщик заданий** позволяет:

1. Назначить задание (запустить некоторую программу) для выполнения ежедневно, еженедельно, ежемесячно или в указанной время, например, при запуске или во время простоя компьютера.
2. Изменить расписание выполнения задания, установив новые дни и время.
3. В любой момент остановить выполнение любого назначенного задания.
4. Назначить дополнительные параметры выполнения задания в указанное время.

Для запуска **Планировщика заданий** нажмите кнопку **Пуск** и выберите разделы **Программы**, **Стандартные**, **Служебные**, а затем **Назначение заданий** (рис.109).

Когда планировщик запущен и его окно раскрыто на экране, для добавления заданий можно воспользоваться перетаскиванием сценариев, программ или документов из проводника Windows или с рабочего стола в окно **Планировщика заданий** или выбрать нужное задание из предложенного списка программ.

В системе Windows **Планировщик заданий** позволяет назначить выполнение любого приложения.

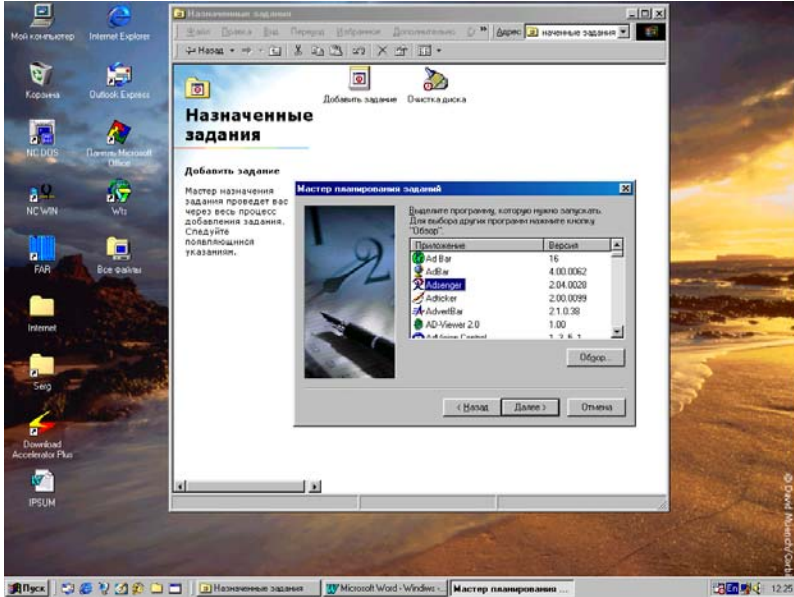


Рис.109. Окно **Планировщика заданий**.

Назначение нового задания

Для назначения нового задания выполните следующие действия:

1. Откройте окно **Планировщика заданий**, щелкнув, например, значок **Назначенные задания** в окне **Мой компьютер**.
2. В окне планировщика дважды щелкните мышкой значок **Добавить задание**.
3. Выполняйте инструкции, выводящиеся на экран в окнах мастера планирования заданий.
4. Если требуется установить дополнительные параметры задания, установите флажок **Установить дополнительные параметры** после нажатия кнопки **"Готово"** в последнем окне мастера.

5. Выберите далее вкладку **Расписание** и нажмите кнопку **Дополнительно** (рис.110). Установите нужные режимы работы планировщика.

Планировщик заданий использует системную дату и время при выполнении назначенных заданий. Для проверки или изменения даты и времени дважды щелкните значение времени на **Панели задач**.

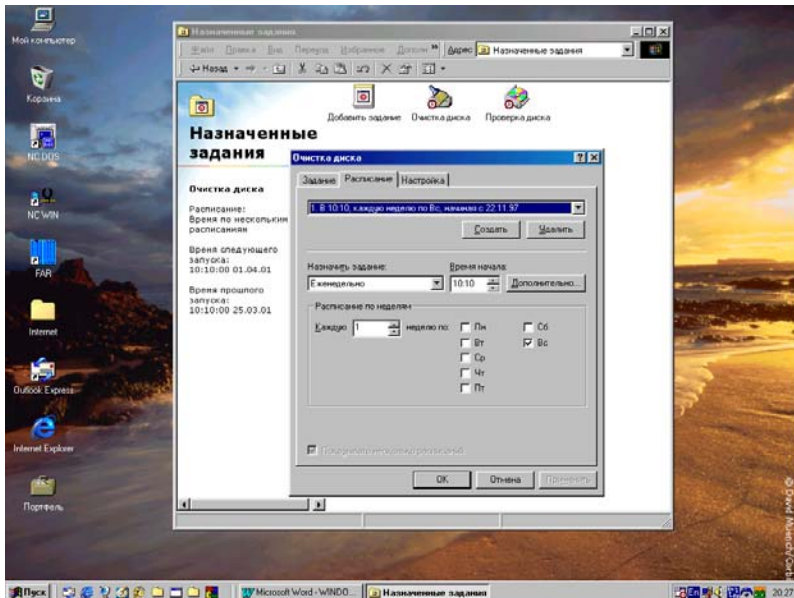


Рис.110.Установка расписания выполнения программы.

Изменение назначенного задания

Для изменения назначенного ранее задания выполните действия:

1. Откройте мышкой папку **Назначенные задания** в окне **Мой компьютер**.
2. Щелкните правой кнопкой задание, которое требуется изменить, и в Контекстном меню выберите команду **Свойства**.
3. Выполните далее одно из следующих действий:
 - 1) Для того, чтобы сменить запускаемую программу, введите путь к программе в поле **Запуск** вкладки **Задание**.

2) Для того, чтобы изменить расписание задания, выберите вкладку **Расписание** и выполните необходимые изменения.

3) Для того, чтобы изменить настройку задания, выберите вкладку **Настройка**.

После внесения нужных изменений в работу планировщика, щелкните мышкой кнопку **ОК** в его окне, которое показано на рис.110.

Удаление назначенного задания

Для удаления задания выполните действия:

1. Откройте мышкой папку **Назначенные задания** в окне **Мой компьютер**.

2. Щелкните правой кнопкой задание, которое требуется удалить, и в Контекстном меню выберите команду **Удалить**.

После удаления назначенного задания его значок удаляется из списка заданий.

Прекращение выполнения текущего задания

Для прекращения работы, выполняемого в данный момент назначенного задания выполните следующие действия:

1. Откройте мышкой папку **Назначенные задания** в окне **Мой компьютер**.

2. Щелкните правой кнопкой задание, которое требуется остановить, и в Контекстном меню выберите команду **Завершить**.

Если в назначенном задании запускается другая программа, то данная команда приведет к остановке только исходного назначенного задания, а второе задание будет выполняться.

Для того, чтобы снова запустить остановленное задание, щелкните правой кнопкой значок задания и в Контекстном меню выберите команду **Запустить**.

Временное отключение всех назначенных заданий

Для временной приостановки назначенных заданий планировщика выполните:

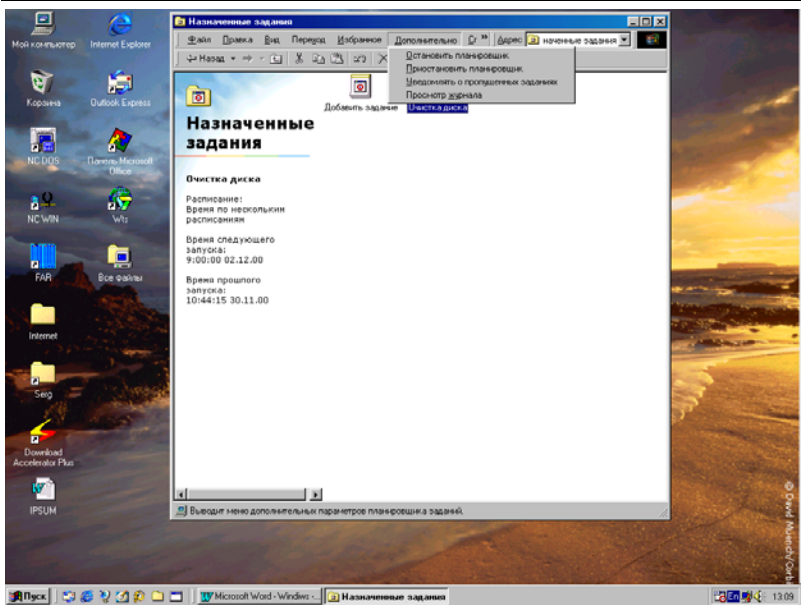


Рис.111. Окно выбора режимов работы Планировщика заданий.

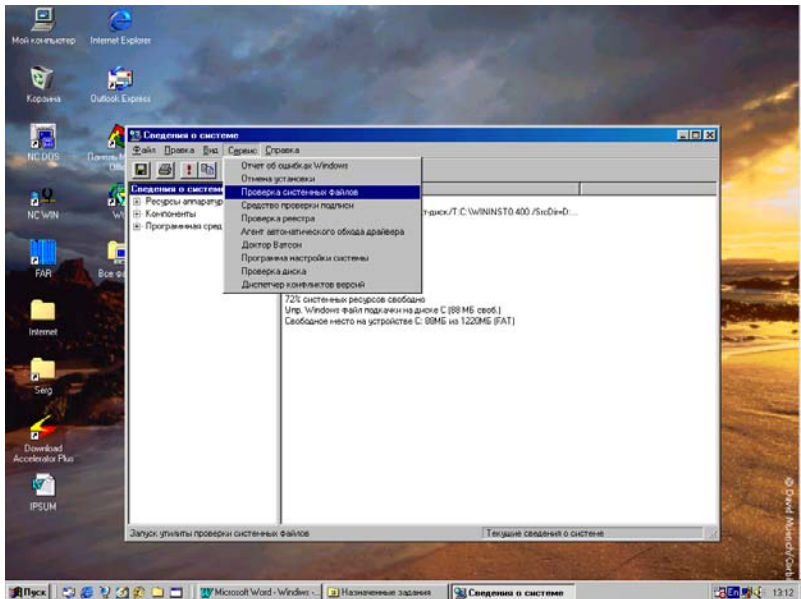


Рис.112. Запуск программы проверки системных файлов.

1. Откройте мышкой папку **Назначенные задания** в окне **Мой компьютер**.
2. В меню **Дополнительно** этого окна выберите команду **Приостановить планировщик** (рис.111).

Это действие бывает полезно при выполнении другой программы (например, игровой) или при установке программного обеспечения, когда назначенное задание может оказаться помехой.

Задания, которые не были выполнены во время приостановки планировщика заданий, будут запущены в следующее время выполнения по графику после запуска планировщика.

Для того, чтобы возобновить выполнение заданий, выберите в меню этого окна пункт **Дополнительно**, команду **Возобновить работу планировщика**.

Остановка планировщика заданий

Для полной остановки планировщика выполните последовательность действий:

1. Откройте мышкой папку **Назначенные задания** в окне **Мой компьютер**.
2. В меню **Дополнительно** выберите команду **Остановить планировщик**.

После выключения (остановки) планировщика заданий, назначенные задания не выполняются и планировщик заданий автоматически не запускается при следующем запуске системы Windows. Заново включить планировщик заданий можно командой **Запустить планировщик** из меню **Дополнительно**.

Проверка системных файлов

Программу **Проверки системных файлов** используют для проверки целостности и работоспособности файлов операционной системы Windows, их восстановления в случае повреждения, а также для распаковки сжатых файлов (таких, как, например, драйверы устройств) с установочных, дистрибутивных дисков.

Программа **Проверки системных файлов** позволяет также резервировать существующие файлы перед восстановлением исходных файлов. Кроме того, пользователь имеет возможность задать на-

стройку условий поиска по именам папок и расширениям имен файлов.

Для запуска программы проверки системных файлов нужно нажать кнопку **Пуск**, выбрать команды **Программы**, **Стандартные**, **Служебные** и **Сведения о системе**, а затем в меню **Сервис** щелкнуть команду **Проверка системных файлов** (рис.112). В появившемся на экране новом окне следует нажать кнопку **Настройка** и установить нужные параметры проверки (рис.113).

Для получения дополнительных и более полных сведений о системной информации следует использовать раздел **Справка** в Главном меню программы.

Отображение системных данных

Программа **Сведения о системе** выполняет сбор сведений о конфигурации системы и содержит меню, позволяющее открывать связанные разделы описания системы. Окно **Сведения о системе** позволяет быстро собрать данные, необходимые для устранения неполадок.

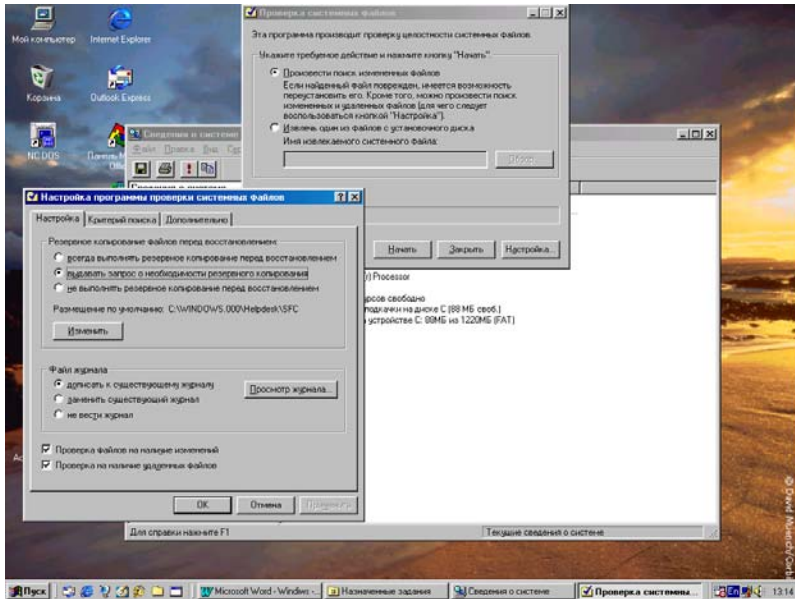


Рис.113. Окно программы **Проверки системных файлов**.

Для отображения системных данных щелкните значок "плюс" в левой области окна программы, чтобы развернуть категорию системных сведений, а затем выберите элемент просмотра. В зависимости от выбранного раздела, может быть предложен выбор основных, дополнительных или журнальных сведений.

Для запуска программы **Сведения о системе** нужно нажать кнопку **Пуск**, в открывшемся **Главном меню** выбрать разделы **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть команду **Сведения о системе** (рис.114).

Пользователь имеет возможность сохранить системные данные в файле системных сведений и в дальнейшем снова просмотреть этот файл в данном приложении. Кроме того, допускается возможность сохранения этих данных в текстовом файле, который можно будет отправить с компьютера через факс или по электронной почте, или просмотреть в другом приложении.

Если сбор данных занимает больше времени, чем ожидается, пользователь имеет возможность прервать процесс сбора данных. Меню **Сервис** обеспечивает быстрый доступ к другим системным средствам, в том числе к программе проверки системных файлов, описанной выше.

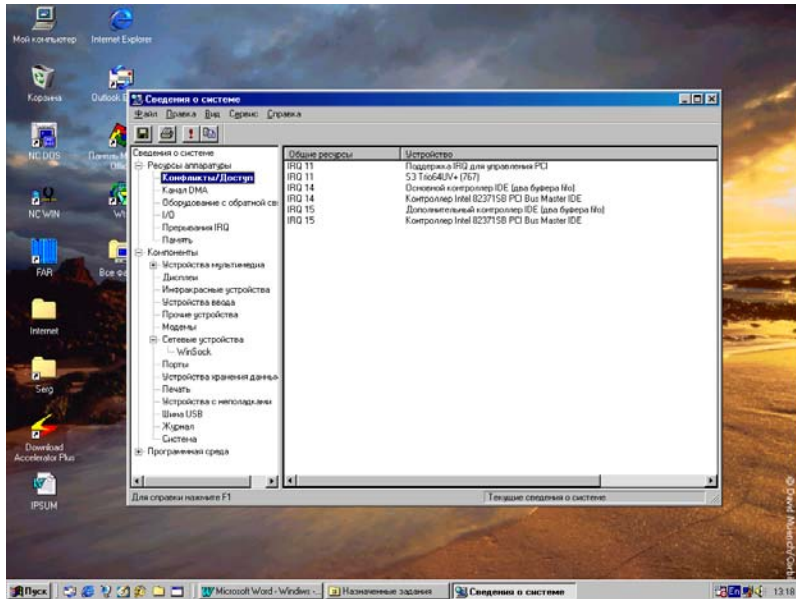


Рис.114. Окно программы **Сведения о системе**.

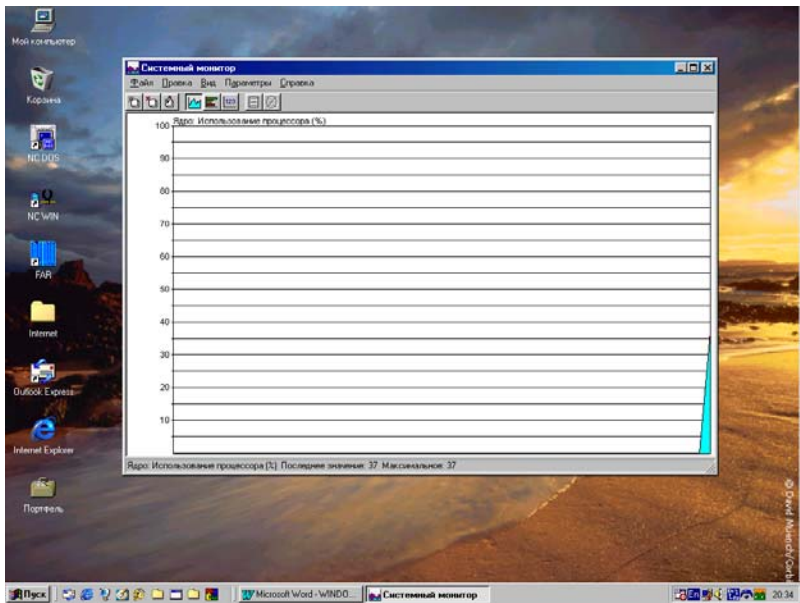


Рис.115. окно программы **Системный монитор**.

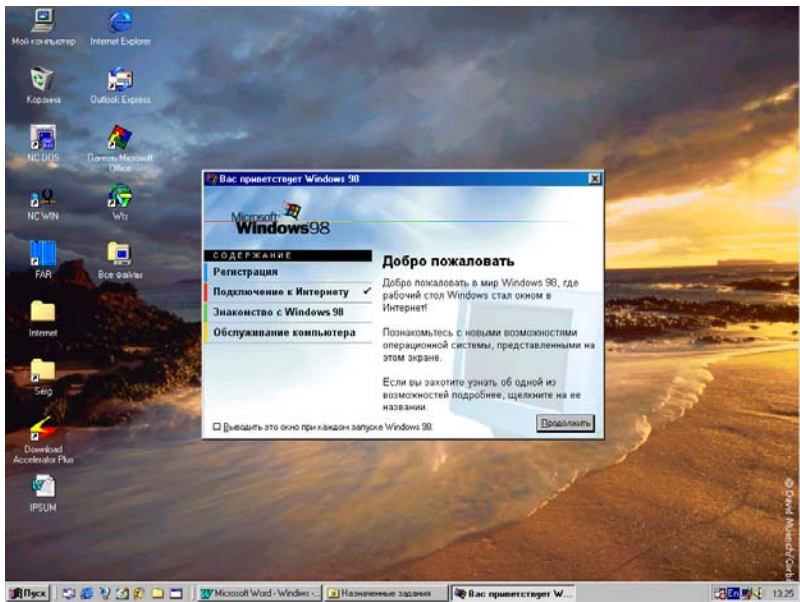


Рис.116. Окно **Приветствия системы**.

Системный монитор

Программа **Системный монитор** использует для наблюдения за быстродействием компьютера или локальной сети. Каждый выбранный показатель отображается в специальном окне на диаграмме, которая обновляется через 5 секунд.

Для запуска **Системного монитора** нужно нажать кнопку **Пуск**, выбрать разделы **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть команду **Системный монитор** (рис.115).

Меню Справка позволяет получить более подробные сведения о работе этой программы.

Приветствие Windows

Программа **Вас приветствует Windows - 98** открывает на рабочем столе целый мир систем Windows. С помощью программы можно познакомиться с новыми приложениями и возможностями компьютера. В меню этого окна можно выбирать интересующие вас разделы и получить краткую справку по основным возможностям работы Windows.

Для запуска программы нужно нажать кнопку **Пуск**, последовательно выбрать пункты **Программы**, **Стандартные** и **Служебные**, а затем щелкнуть мышкой команду **Вас приветствует Windows - 98**, окно которой показано на рис.116.

ПРИКЛАДНЫЕ ПРОГРАММЫ

Под прикладными программами понимается некий набор программ пользователя компьютера, предназначенных для решения тех или иных задач, например, набора текста, редактирования рисунков, сканирование текста и фотографий, перевода с одного языка на другой и т.д. Ниже кратко опишем основные способы запуска программ с открытием их окон, свертывание этих окон и полное выключение программы с закрытием ее окна на экране компьютера.

Запуск прикладных программы

Для запуска любой программы с открытием ее окна на экране компьютера:

1. Нажмите кнопку **Пуск**, выберите команду **Программы** и щелкните по названию прикладной программы, как показано на рис.117.
2. Если программа отсутствует в меню и не имеет **Ярлыка на Рабочем столе**, откройте **Проводником** или через **Мой компьютер** содержащую ее папку и щелкните по ее имени.

Любой запущенной программе соответствует кнопка на **Панели задач**.

Для того, чтобы перейти от одной выполняющейся программы к другой, достаточно нажать соответствующую ей кнопку. Если программа не включена ни в меню **Программы**, ни в одно из его подменю и на **Рабочем столе** нет ее ярлыка - значка, а вы не знаете, где она находится на жестком диске, нажмите кнопку **Пуск** и выберите команды **Найти**, раздел **Файлы и папки**. Найдите исполняемый файл программы с помощью диалогового окна **Найти**. Однако, это возможно сделать только, если вам известно имя такой программы.

Если нужная программа не установлена на компьютер, требуется ее установка с дистрибутивного пакета на лазерном диске или с гибких дискет. После такой установки, в меню **Программы** кнопки **Пуск** или на **Рабочем столе** обычно появляется пункт или **Ярлык** этой программы.

Свертывание всех открытых окон

Для свертывания всех открытых на экране окон выполните следующие действия:

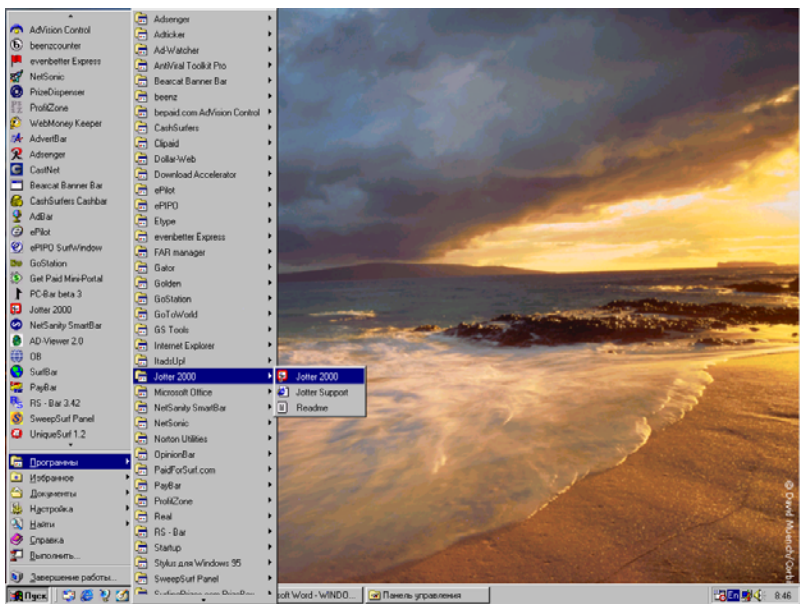


Рис.117. Запуск прикладных программ

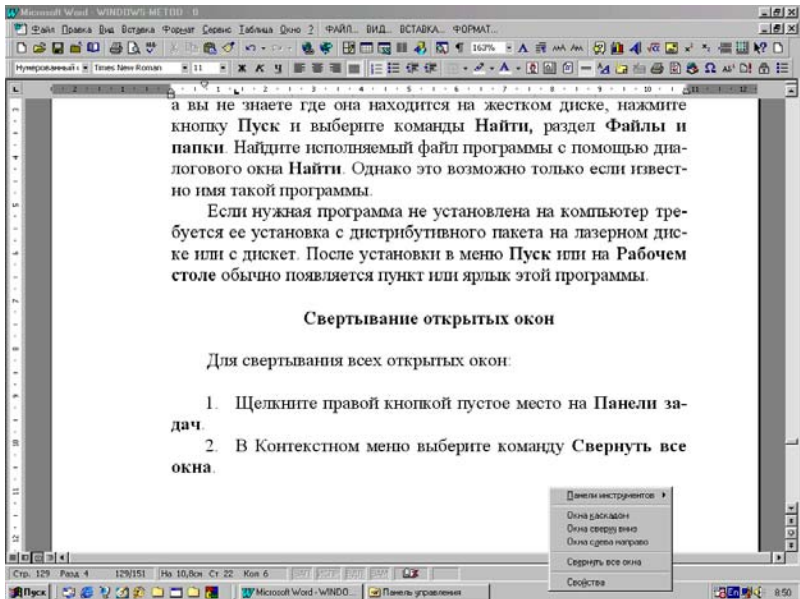


Рис.118. Свертывание всех открытых окон.

1. Щелкните правой кнопкой мышки пустое, свободное место на **Панели задач**.
2. В появившемся Контекстном меню выберите команду **Свернуть все окна** (рис.118).

Можно также последовательно щелкнуть по кнопке со знаком минус "-" в правом верхнем углу окна каждой программы. Для того, чтобы открыть свернутое окно, нажмите соответствующую ему кнопку на **Панели задач**. Для изменения размеров окон можно использовать кнопку с двумя квадратами (**Восстановить**) или с одним квадратом (**Развернуть**) в правом верхнем углу окна программы.

Для того, чтобы восстановить исходные размеры и размещение всех окон после сворачивания, снова щелкните правой кнопкой пустое место на **Панели задач** и в Контекстном меню выберите команду **Отменить Свернуть все**.

Выход из программы

Для завершения работы любой программы и закрытия ее окна в меню **Файл** этой программы выберите команду **Выход**. Можно также закрыть программу нажатием кнопки с крестиком "x" у правого верхнего края строки заголовка ее окна. Если в программе есть не сохраненные данные будет предложено сохранить их.

Для закрытия окна программы при ее зависании нажмите клавиши Ctrl + Alt + Del. В появившемся диалоговом окне **Завершение работы программы** выберите программу, которая не отвечает на системные запросы (справа от нее будет написано "Не отвечает") и нажмите кнопку **Завершить задачу**, а затем, если появилось новое окно подтверждения снова щелкните кнопку **Завершить задачу** (рис.119). При этом все не сохраненные предварительно данные будут потеряны.

Для выключения компьютера можно нажать клавиши Ctrl + Alt + Del, а затем кнопку **Завершить работу** в диалоговом окне **Завершение работы программы**. Если компьютер не выключился автоматически можно отключить его от электрической сети, используя кнопку на корпусе системного блока.

Примеры прикладным программ

Приведем несколько примеров прикладным программ, работающих на базе системы Windows. В частности, кратко рассмотрим программы Norton Commander, Word, Excel, Internet Explorer, Outlook Express, Front Page, Photo Editor и т.д.

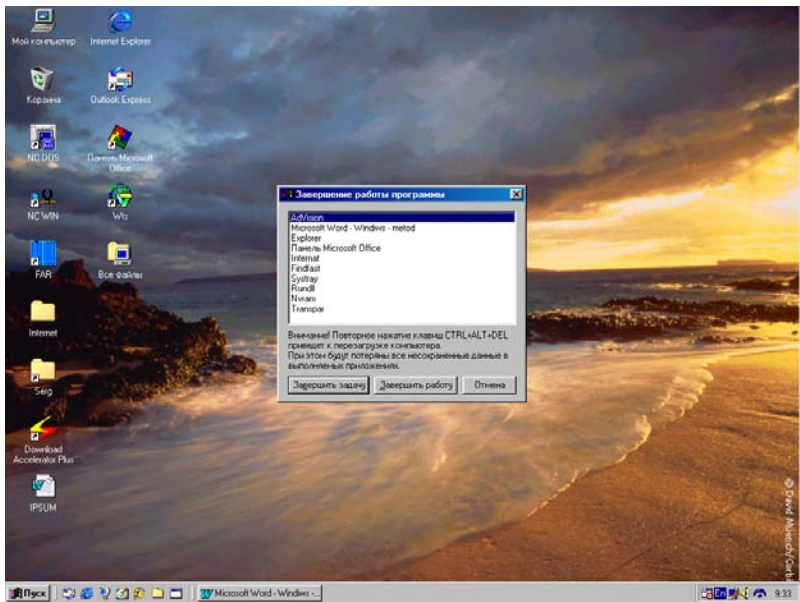


Рис.119. Окно для завершения работы программы.

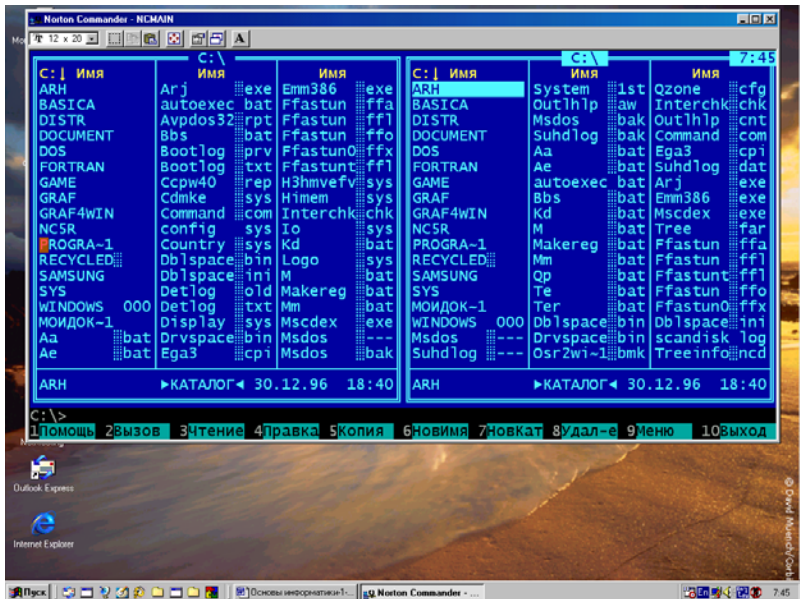


Рис.120. Основное окно программы Norton Commander.

Программа оболочка Norton Commander

Программа Norton Commander (NC) может работать и на основе системы Windows. Эта программа используется для работы с файлами и папками (директориями), и является некоторым аналогом программы **Проводник**. NC устанавливалась на компьютеры еще тогда, когда не существовало программы Windows и являлась, по сути, единственным файловым менеджером для компьютеров IBM PC AT.

Вызов программы может осуществляться с помощью ее **Ярлыка** на **Рабочем столе**, который устанавливается, описанными ранее методами. Вид экрана после вызова программы NC показан на рис.120. Возможен и переход в полноэкранный режим работы, выполняемый нажатием комбинации клавиш Alt + Enter. Повторное нажатие этих клавиш возвращает экран в исходное положение.

Для того, чтобы при выходе из программы по клавише F10 закрывалось ее окно, нужно щелкнуть по **Ярлыку** NC правой кнопкой мыши и в появившемся на экране Контекстном меню выбрать команду **Свойства**. Затем перейти на вкладку **Программы** и установить "птичку" в окне **Закрывать окно по завершению сеанса работы**, как показано на рис.121.

В меню свойств можно сменить значок программы, установить некоторые дополнительные параметры и определить рабочий каталог (например, диск C:), который будет открываться при запуске NC. Другие вкладки этого окна позволяют менять шрифты, используемые программой, задавать распределение памяти и определять вид экрана.

Программа Word

Программа MS Word является основным текстовым редактором, которой используется на основе операционной системы Windows. Она позволяет набирать и редактировать тексты, создавать произвольные таблицы и рисунки, строить диаграммы по заданным таблицам, набирать сложные математические формулы и многое другое.

Запускается программа обычно с панели MS Office или из раздела **Программы Главного меню** кнопки **Пуск**. Вид экрана после ее запуска показан на рис.122.

Программа Excel

Программа используется для создания различных таблиц, в которых можно выполнять некоторые типы математических вычислений.

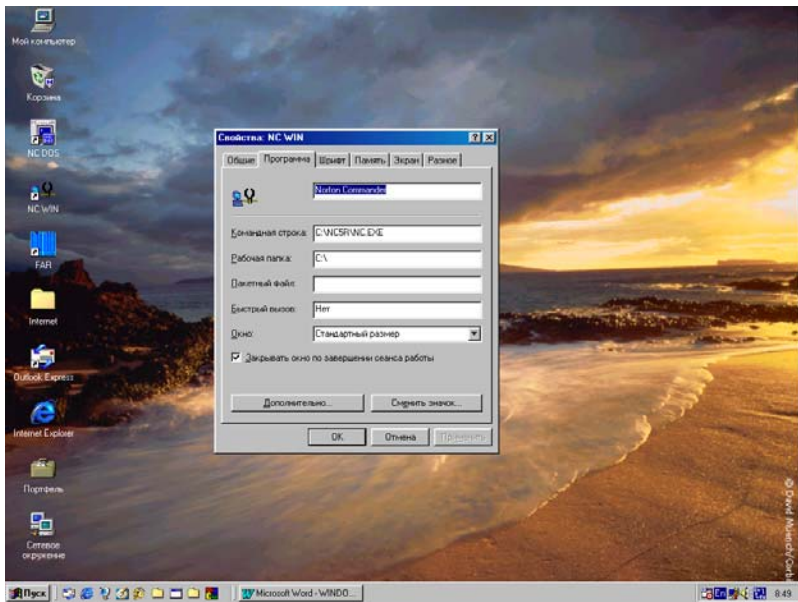


Рис.121. Меню **Свойства** программы NC.

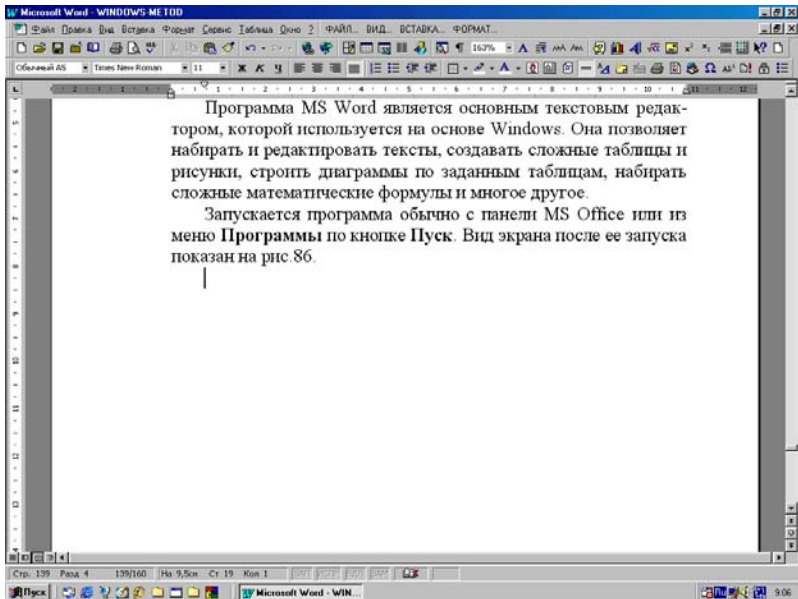


Рис.122. Вид экрана после запуска программы Word.

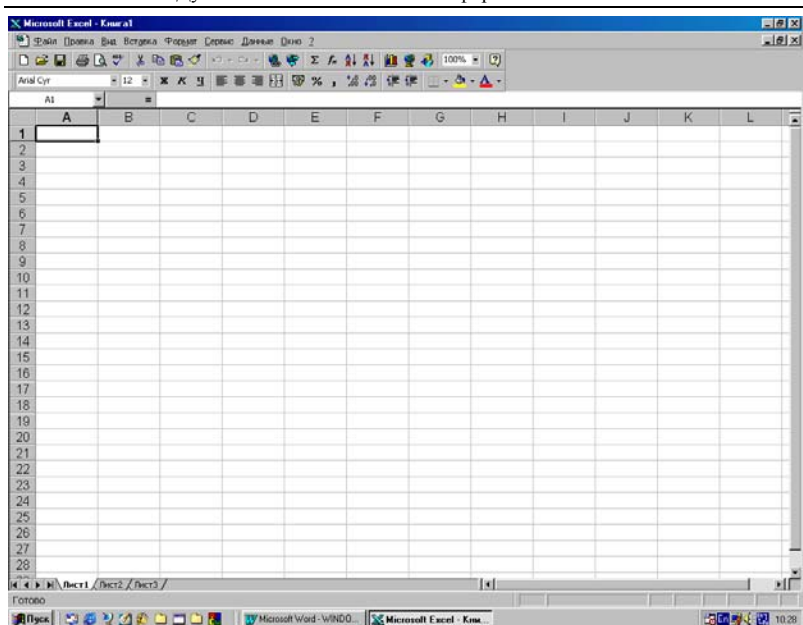


Рис.123. Окно программы Excel.

Разделы **Главного меню** и **Панели инструментов** самой программы позволяют оформлять таблицу и задавать вид математических операций, выполняемых над числами, которые предварительно заносятся в определенные клетки таблицы.

Запускается программа с панели MS Office или из раздела **Программы** кнопки **Пуск**. Вид окна после ее запуска показан на рис.123.

Программа Internet Explorer

Программа позволяет пользователю работать в Интернете, просматривая содержимое Web - узлов. Возможно использование этой программы и для работы с электронной почтой на серверах, допускающих прямой доступ к вашему почтовому ящику, например, сервер Казахстана - <http://www.itte.kz>.

Запускается программа обычно с помощью значка на **Рабочем столе** или из меню **Быстрого запуска** на **Панели задач**. Вид экрана после запуска программы показан на рис.124. В основном окне программы показано оформление сервера Lorton компании S&G Communications. Левое окно содержит список папки **Избранное**, куда заносятся все наиболее часто используемые адреса Интернета. Более

подробное описание этой программы будет дано во второй части книги. Ги.

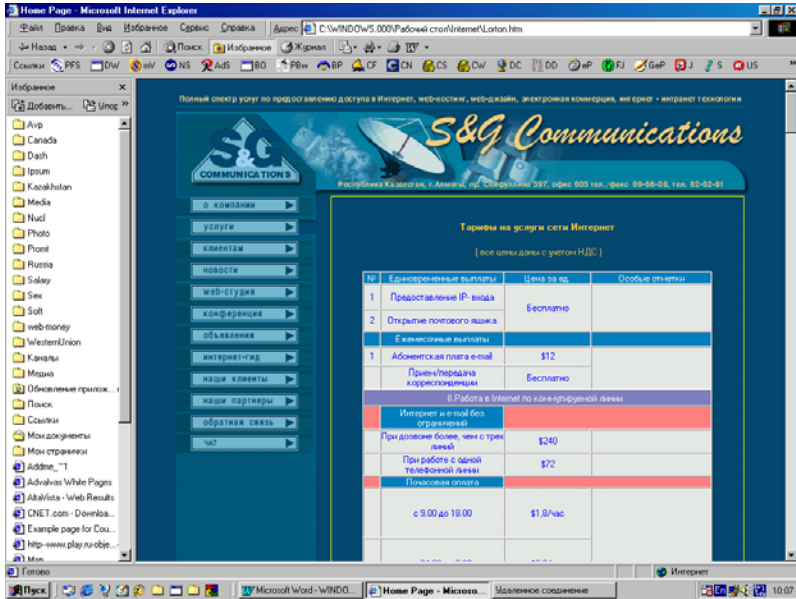


Рис.124. Окно программы Internet Explorer.

Программа Outlook Express

Программа используется для получения, написания и отправки или пересылки писем электронной почты. Запускается программа значком на **Рабочем столе** или из меню **Быстрого запуска Панели задач**. Вид окна программы после ее запуска показан на рис.125.

В левой части окна показан список папок, в которых хранятся входящие (полученные) и исходящие (отправленные) письма. Верхнее окно справа содержит список писем, выделенной слева папки, в данном случае это папка **Входящие**, а в нижней части окна приводится текст, выделенного сверху письма.

Панель инструментов и **Главное меню** программы позволяют работать с письмами, выполнять настройку соединения с Интернетом и своим почтовым ящиком для отправки и получения корреспонденции. Эта программа, также как обозреватель Интернета Internet Explorer входит в комплект Windows и может быть установлена на компьютер вместе с установкой самой системы. Более подробное описание этой программы будет дано во второй части данной книги.

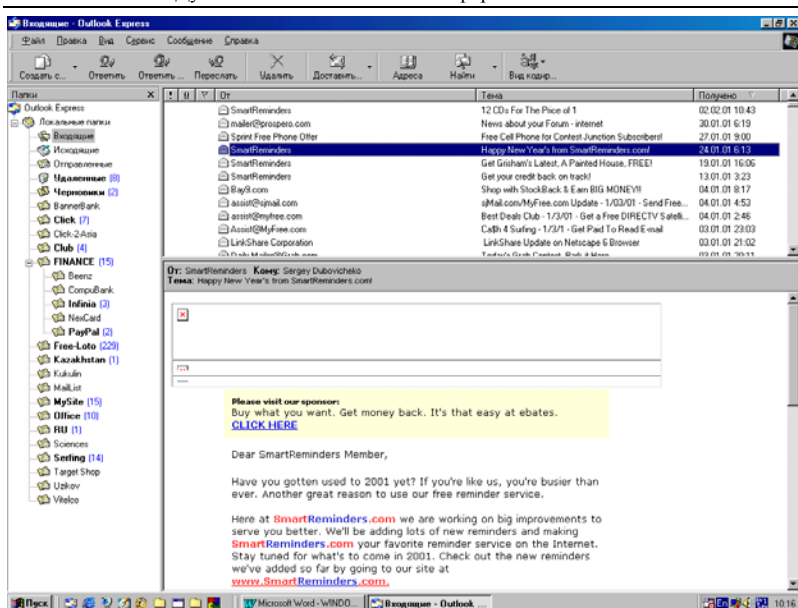


Рис.125. Окно почтовой программы Outlook Express.

Программа Photo Editor

Программа используется для редактирования графических изображений в файлах некоторых форматов. Допускается масштабирование изображения, изменение реальных размеров, выделение произвольных частей рисунков с последующим копированием или переносом этих частей.

Программа обычно запускается из меню **Программы** кнопки **Пуск**. Вид окна программы показан на рис.126. В окне программы показано меню выбора начальных параметров при создании нового рабочего окна для загрузки и редактирования рисунков.

Программа для Web - дизайна Front Page Express

Программа используется для разработки и создания Интернетовских страничек в HTML формате. Написанную, таким образом, страничку можно размещать на любом Интернетовском сервере, если он допускает размещение личных страничек. Такое размещение выполняется не только с помощью возможностей самого Интернетовского сервера, но и благодаря возможностям программы Front Page.

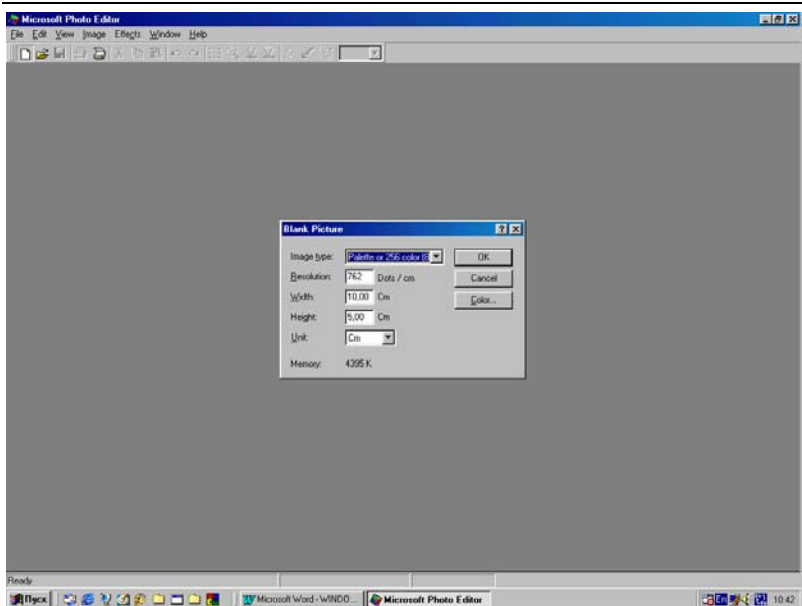


Рис.126. Окно программы MS Photo Editor.

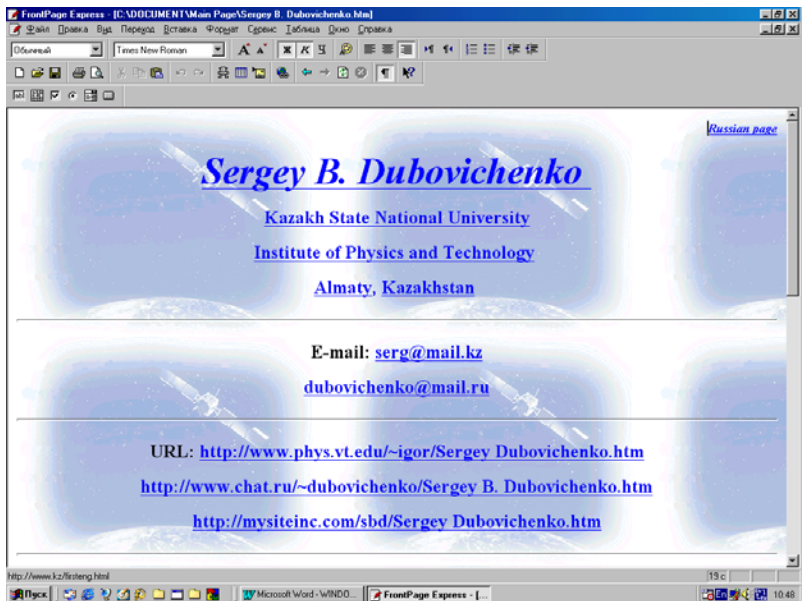


Рис.127. Окно программы Front Page Express.

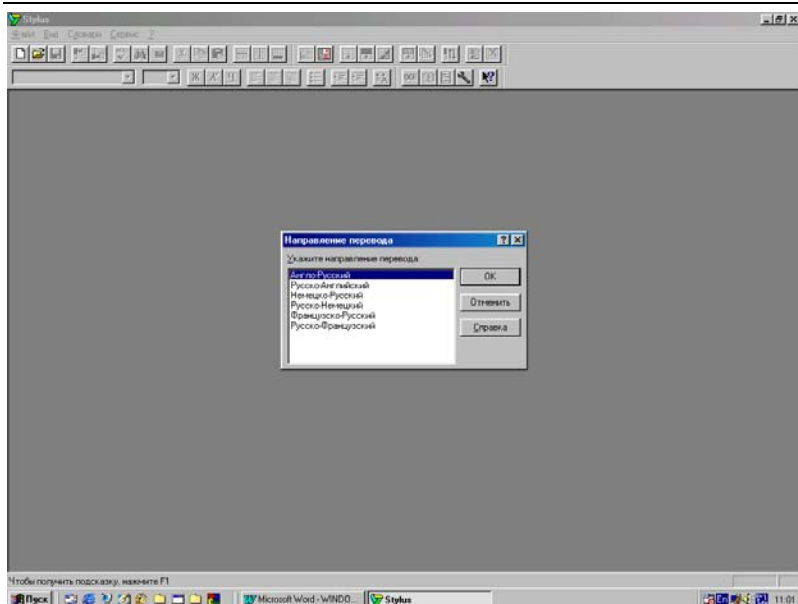


Рис.128. Окно программы переводчика Stylus.

Программа обычно запускается из меню **Программы** кнопки **Пуск**. Вид окна программы после ее запуска приведен на рис.127. В окне программы показан пример создания личной Интернетовской странички, которая размещена по адресам URL, также приведенным на рисунке.

Программа переводчика текста Stylus.

Программа применяется для переводов на и с трех языков - английского, немецкого и французского. Может работать совместно с текстовым редактором Word, что оказывается очень удобным при работе с текстами на иностранных языках. Дополнительно к программе Stylus, которая сама имеет сравнительно большой словарный запас, могут устанавливаться дополнительные специализированные словари по разным направлениям естественных и гуманитарных наук.

Запускается программа из меню **Программы** или с помощью значка на **Рабочем столе** (рис.128). На рисунке видно также окно

задания направления перевода при создании нового рабочего окна программы.

На рис.129 показано основное окно программы с примером перевода блока текста, приведенного в верхней части окна. Перевод обычно выполняется дословно, но в большинстве случаев программа сама оформляет перевод (в данном случае на английский) в соответствии с правилами грамматики языка перевода.

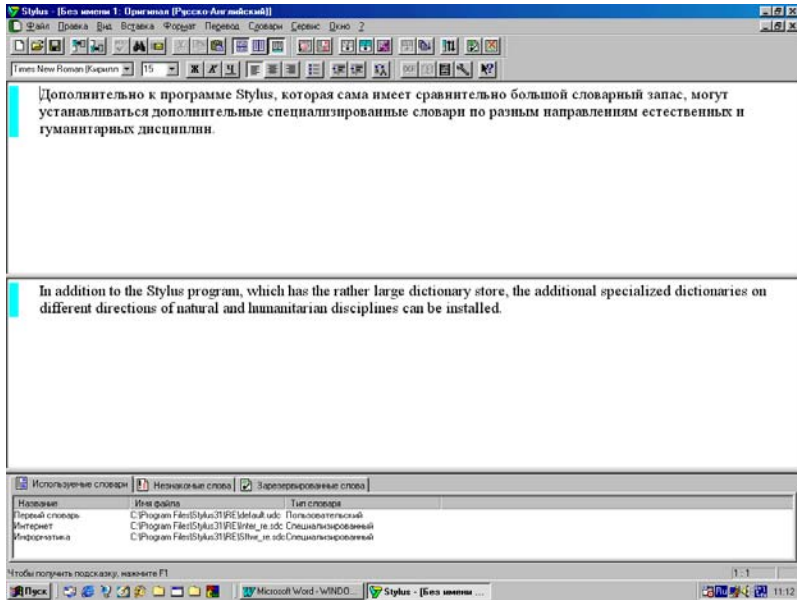


Рис.129. Пример перевода набранного блока текста.

Антивирусная программа AVP - 32

Программа AVP - 32 одна и лучших антивирусных программ (если не самая лучшая), используемая для поиска и уничтожения вирусов на компьютере. Программа распознает и вылечивает более 43 тысяч различных и, в том числе, скрытых вирусов.

Запускается программа с помощью значка - **Ярлыка** на **Рабочем столе** компьютера или из **Главного меню Программы** кнопки **Пуск**. Вид окна программы после ее запуска показан на рис.130, где внизу видна дата обновления программы и общее число вирусов, которые могут распознаваться в процессе работы.

Программа допускает обновление антивирусных баз через Интернет на сервере авторов программы. Возможна работа в локальных компьютерных сетях любой конфигурации и проверка любых дисков, в том числе, лазерных CD дисков. Однако, в последнем случае допустима только диагностика вирусов, но не их лечение, поскольку обычные устройства CD - ROM предназначены только для чтения. После запуска программа автоматически проверяет оперативную память компьютера, а после нажатия кнопки **Пуск** начинается проверка жесткого диска, отмеченного в окне выбора дисков на вкладке **Область**.

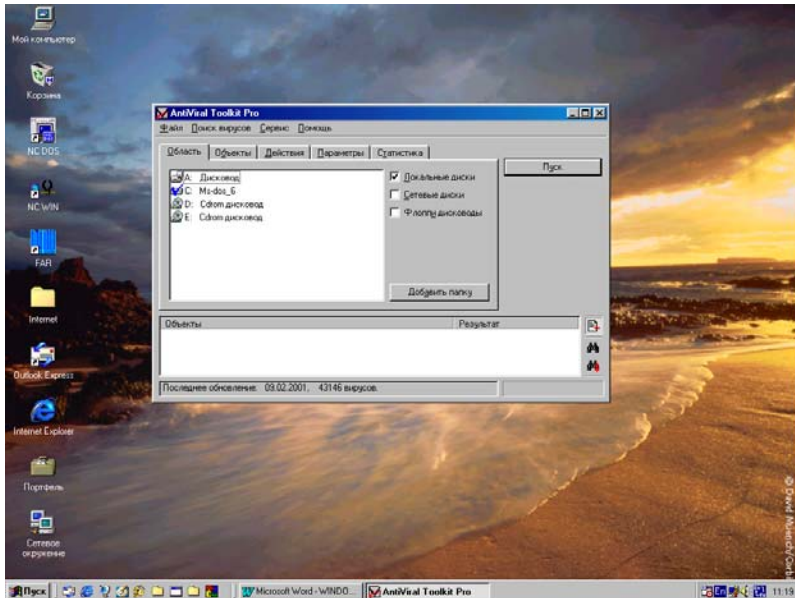


Рис.130. Антивирусная программа.

Программа просмотра графических файлов ACDSSee 32

Программа позволяет просматривать множество графических файлов различных форматов и типов. В окне слева вверху показан путь к файлу, а слева внизу дается предварительный просмотр файла, выделенного в основном окне программы (справа). Для полного просмотра рисунка нужно дважды щелкнуть по имени его файла.

Для запуска программы можно использовать раздел **Программы Главного меню** кнопки **Пуск** или ее **Ярлык** на **Рабочем столе**. Окно программы после ее запуска с включенным режимом предвари-

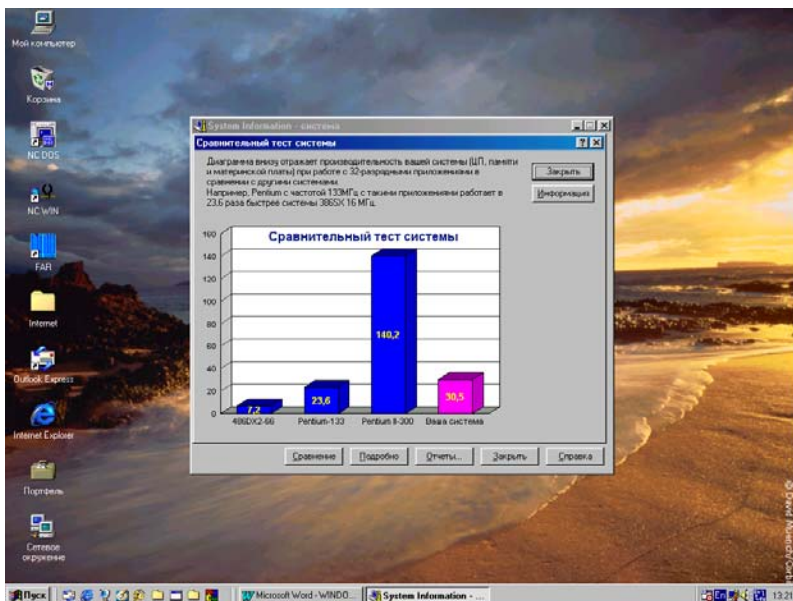


Рис.132. Окно программы Системной информации.

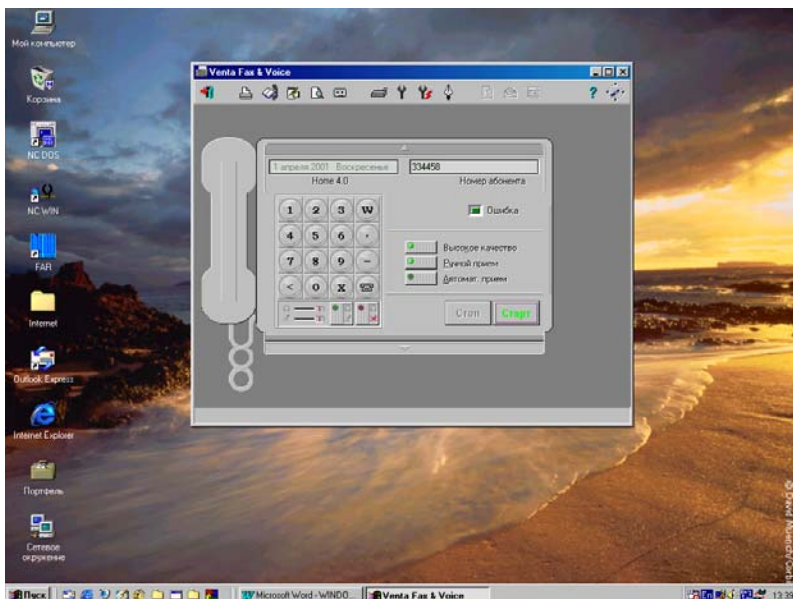


Рис.133. Программа для работы с факсами.

Программа для факсов

Программа Venta Fax предназначена для отправки и приема факсов непосредственно с компьютера. Перед отправкой факс должен быть определенным образом подготовлен.

Программа способна посылать в виде факса текстовые файлы, сделанные в различных программах - приложениях системы, после их преобразования в факсовский формат. Автоматический режим работы программы позволяет принимать факсы с других факсовых аппаратов или другого компьютера, на котором установлена какая - либо программа для факсов.

Запустить программу, если она установлена на компьютер можно из меню **Программы** кнопки **Пуск**. Окно программы Venta Fax после ее запуска показано на рис.133.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. MS DOS 6.2. - Руководство для пользователей компьютеров IBM PC., М., ВА Принт, 1994.
2. Зверев С.А. - Практическая работа в MS DOS., М., Воениздат, 1991.
3. Нортон П. - Персональный компьютер IBM и операционная система DOS., М., Радио и связь, 1991.
4. Федор Ж. - Операционные системы от PC до PS/2., М., Мир, 1992.
5. Симрин С. - Библия MS - DOS., М., Мир, 1985.
6. Цикритзис Д. - Операционные системы., М., Мир, 1977.
7. Help - файл операционной системы DOS., Microsoft Corp., 1995.
8. Левин А. Самоучитель работы на компьютере. - М, 1998.
9. Фигурнов В.Э. IBM PC для пользователей. - М, 1995.
10. Файл описания новых возможностей программы NC-5.0. - whats.new. Symantec Corp., 1995.
11. Файл описания программы NC-5.0 - readme.txt. Symantec Corp., 1995.
12. McNamara T. Файл описания программы - nc_exit. doc. Symantec Corp., 1995.
13. Help файл программы Norton Commander. - nc.hlp. Symantec Corp., 1995.
14. Смирнов Н.Н. - Программные средства персональных ЭВМ, Л., 1990, 270с.
15. Моррил Г. - Бейсик для ПК ИБМ, М., ФиС, 1987, 260с.
16. Пярнпу А.А. - Программирование на современных алгоритмических языках, М., Наука, 1990, 383с.
17. Лемешко Е.В., Романчук В.Г. - Основы программирования на языке Бейсик, М. МИУ, 1988, 49с.
18. Блэнд Г. - Основы программирования на языке Бейсик, М., ФиС, 1989, 207с.
19. Геворкян Г.Х., Семенов В.Н. - Бейсик - это просто, М., Радио и Связь, 1989, 143с.
20. Дьяконов В.П. - Применение персональных ЭВМ и программирование на языке Бейсик, М., Радио и Связь, 1989, 286с.
21. Дьяконов В.П. - Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ, М., Наука, 1989, 239с.
22. Кетков Ю.Л. - Диалог на языке Бейсик для мини - и микро ЭВМ, М., Наука, 1988, 367с.
23. Кетков Ю.Л. - Программирование на Бейсике, М., Статистика, 1978, 158с.

24. Кэтлин Э. - Программирование на языке Бейсик, М., Мир, 1990, 288с.
25. Вашкевич Ю.Ф. и др. - Справочник по программированию на Бейсике, М. Радио и Связь, 1987. 335с.
26. Уорт Т. - Программирование на языке Бейсик, М., Машиностроение, 1981. 255с.
27. Трояновский В.М., Шаньгин В.Ф. - Бейсик для начинающих и будущих профессионалов, М., Высшая школа, 1992. 239с.
28. Башмаков Е.С., и др. - Программирование микро ЭВМ на языке Бейсик, М., Радио и Связь, 1991. 239с.
29. Джордейн Р. - Справочник программиста персональных компьютеров типа IBM PC, XT и AT, Пер. с англ. и предисл. Н.И. Гайского, М., Финансы и статистика, 1991, 554с.
30. Очков В.Ф., Пухначев Ю.В. - 128 советов начинающему программисту, М., Энергоатомиздат, 1991, 256с.
31. Зельднер Г.А. - Quick BASIC для носорога, М., ABF, 1994, 461с.
32. Корчак А.Е. - Справочник. БЕЙСИК - версии для MS-DOS, М., МЦНТИ, 1996, в 2-томах, 463с.
33. Ясинский В. - Turbo, Quick и PDS Basic - алгоритмы и программы, М., 1997.
34. Данилина И.И. - Численные методы, М., Высшая школа, 1976, 367с.
35. Попов Б.А., Теслер Г.С. - Вычисление функций на ЭВМ, Киев, Наукова думка, 1984, 597с.
36. Смирнов Н.Н. - Программные средства персональных ЭВМ., Л., 1990.
37. Help - файл помощи Windows., Microsoft Co. 1998.
38. Абрамов С.А. - Начала Информатики., М., Наука, 1990.
39. Алтухов Е.В. - Основы информатики и вычислительной техники., М., ВШ, 1992.
40. Фролов Д. - Элементы Информатики., М., ВШ, 1989.
41. Шрайберг Я.Л. - Справочное руководство по основам Информатики., М., Финансы и статистика, 1990.
42. Дубовиченко С.Б. - Некоторые версии алгоритмического языка Бейсик., Алматы, АЭиП, 2001, 166с.
43. Дубовиченко С.Б. - Русский Norton Commander -5., Алматы, УЭиП, 2000, 96с.

Сергей Борисович Дубовиченко

Член - корреспондент Казахстанской Международной
Академии Информатизации

Основы информатики Часть 1

Подписано к печати 25.12.2001. Формат 60x84 1/16. Бумага офсетная.
Печать офсетная. Уч. - изд. л. 27,4. Заказ №113. Тираж 500.
Цена договорная.

Издательство Института международного права и международного бизнеса
480060, г. Алматы, ул. Ходжанова, 9,
Институт международного права и международного бизнеса
Отпечатано в типографии Института международного права и
международного бизнеса