

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное автономное образовательное учреждение высшего образования
«Севастопольский государственный университет»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к лабораторным работам по дисциплине
“ОПЕРАЦИОННЫЕ СИСТЕМЫ”, часть 2,
для студентов ОФО и ЗФО направлений подготовки:
09.03.01 «Информатика и вычислительная техника»
09.03.02 «Информационные системы и технологии»
27.03.04 «Управление в технических системах»**

Севастополь
2018

УДК 004.451

Методические указания к выполнению лабораторных работ по дисциплине «Операционные системы», часть 1, для студентов ОФО и ЗФО направлений подготовки: 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии», 27.03.04 «Управление в технических системах» / Сост. ст. преподаватель кафедры ИТиКС С.Н. Фисун, Е.М. Шалимова. – Севастополь: Изд-во СевГУ, 2018. – 28с.

Целью методических указаний является оказание помощи студентам в выполнении лабораторных работ по дисциплине «Операционные системы».

Приведены теоретические сведения, необходимые для выполнения лабораторных работ, варианты заданий, рекомендации по выполнению, требования к отчетам.

Методические указания рассмотрены и утверждены на заседании кафедры «Информационные технологии и компьютерные системы», протокол № 1 от 31.08.2018г.

Рецензент: доцент кафедры «Информационные технологии и компьютерные системы», к.т.н. Е.В. Козлова.

СОДЕРЖАНИЕ

Введение.....	4
1. Лабораторная работа №5. Исследование алгоритмов замещения страниц.....	5
2. Лабораторная работа №6. Интерпретатор командной строки ОС WINDOWS	10
3. Лабораторная работа №7. Управление файлами в ОС WINDOWS	21
Библиографический список.....	28

1. ЛАБОРАТОРНАЯ РАБОТА №5. ИССЛЕДОВАНИЕ АЛГОРИТМОВ ЗАМЕЩЕНИЯ СТРАНИЦ

Цель работы: исследовать стратегии и методы управления виртуальной памятью в современных ОС.

1.1 Основные теоретические положения

1.1.1 Виртуальная память

Страничная организация памяти является одним из основных способов реализации виртуальной памяти. Множество виртуальных адресов, как правило, значительно превышает диапазон адресов первичной памяти. Суть концепции виртуальной памяти заключается в том, что виртуальные адреса, на которые ссылается выполняющийся процесс, отделяются от адресов, существующих в первичной памяти, т.е. реальных адресов. Диапазон виртуальных адресов, к которым может обращаться процесс, называется виртуальным адресным пространством V этого процесса. Диапазон реальных адресов, существующих в конкретной вычислительной машине, называется реальным адресным пространством R этой машины.

Перевод виртуальных адресов в реальные называется динамическим преобразованием адресов. Динамическое преобразование адресов выполняет сама система, причем это делается прозрачно (невидимо для пользователя). Искусственная смежность, или непрерывность, означает, что адреса, смежные в виртуальном адресном пространстве V , не обязательно будут смежными в реальном адресном пространстве R .

Только небольшая часть процедур и данных каждого процесса, как правило, размещается в первичной памяти одновременно. Остальная часть хранится на устройствах внешней памяти с быстрым доступом.

Системы виртуальной памяти требуют наличия таблиц отображения виртуальных адресов на реальные адреса. Главная проблема для разработчиков систем виртуальной памяти – это минимизация количества информации отображения, которую необходимо держать в первичной памяти, гарантируя в то же время удовлетворительные скоростные характеристики системы. Достижению этой цели способствует так называемое поблочное отображение. Блоки фиксированного размера называются страницами; блоки переменного размера называются сегментами. В некоторых системах оба вида блоков комбинируются, причем, как правило, применяются сегменты, длина которых выражается целым числом страниц фиксированного размера. В схемах блочного отображения адреса могут рассматриваться как упорядоченные пары $\vartheta=(b,d)$, где b – блок, в котором находится виртуальный адрес ϑ , а d – смещение адреса ϑ относительно начала блока b .

В системе с блочным отображением виртуальные адреса преобразуются в реальные при помощи таблицы отображения блоков. Из скоростных соображений эта таблица зачастую размещается в быстродействующей кэш-памяти или ассоциативной памяти. Виртуальный адрес при страничной организации – это упорядоченная пара $\vartheta=(p,d)$, где p – номер страницы содержащей ϑ , а d – смещение адреса ϑ относительно начала страницы p .

Первичная память разделяется на страничные кадры того же самого размера, что и виртуальные страницы. Поступающая в первичную память страница может быть помещена в любой свободный страничный кадр.

Динамическое преобразование адресов при страничной организации памяти предусматривает отображение номера страницы на страничный кадр p . Отображение может быть прямым – в этом случае используется полная таблица страниц, размещаемая либо в первичной памяти, либо в высокоскоростной кэш-памяти. Отображение может быть ассоциативным – в этом случае полная таблица страниц размещается в высокоскоростной ассоциативной памяти. Ввиду высокой стоимости ассоциативной памяти и кэш-памяти отображение может быть комбинированным, когда только самые последние по времени

обращения страницы содержатся в ассоциативной памяти, а полные таблицы прямого отображения используются, если поиск по ассоциативной памяти дает отрицательный результат. Виртуальная память чаще реализуется с использованием кэш-памяти, чем ассоциативной памяти.

Для коллективного использования (разделения) информации в страничной системе в строках таблиц страниц различных процессов указывается один и тот же страничный кадр. Совместное использование, реализуемое таким способом, связано с неудобствами, поскольку по смыслу процессы коллективно используют такие логические объекты, как процедуры и данные, которые могут зачастую занимать по несколько страниц и расширяться и сокращаться во время выполнения процесса.

1.1.2. Стратегии управления виртуальной памятью

В лабораторной работе рассматриваются следующие стратегии управления виртуальной памятью: стратегия вталкивания, стратегия размещения, стратегия замещения. Стратегия замещения (выталкивания)

Стратегия вталкивания. Цель стратегии – определить, в какой момент следует переписать страницу или сегмент из вторичной памяти в первичную. Вталкивание по запросу (по требованию) предполагает, что система ждёт ссылки на страницу или сегмент от выполняющегося процесса и только после появления ссылки начинает переписывать данную страницу или сегмент в первичную память. Вталкивание с упреждением (опережением) предполагает, что система пытается заблаговременно определить, к каким страницам или сегментам будет обращаться процесс. Если вероятность обращения высока и в первичной памяти имеется свободное место, то соответствующие страницы или сегменты будут переписываться в основную память ещё до того, как к ним будет явно производиться обращение.

Стратегия размещения. Цель стратегии – определить, в какое место первичной памяти помещать поступающую страницу или сегмент. В системах со страничной организацией решение о размещении принимается достаточно тривиально, поскольку поступающая страница может быть помещена в любой свободный страничный кадр. Системы с сегментной организацией требуют стратегий размещения, аналогичных тем, которые применяются в системах мультипрограммирования с переменными разделами.

Стратегия замещения (выталкивания). Цель стратегии – решить, какую страницу или сегмент следует удалить из первичной памяти, чтобы освободить место для помещения поступающей страницы или сегмента, если первичная память полностью занята.

1.1.3. Алгоритмы замещения страниц

1.1.3.1 Принцип оптимальности (OPT)

Принцип оптимальности означает, что для обеспечения оптимальных скоростных характеристик и эффективного использования ресурсов следует заменять ту страницу, к которой в дальнейшем не будет новых обращений в течение наиболее длительного времени. Теоретически можно доказать, что подобная стратегия действительно оптимальна, однако реализовать ее на практике нельзя.

В связи с этим для обеспечения высоких скоростных характеристик и эффективного использования ресурсов мы попытаемся наиболее близко подойти к принципу оптимальности, применяя различные методы выталкивания страниц, приближающиеся к оптимальному.

1.1.3.2 Выталкивание случайной страницы (RANDOM)

Если вам нужно иметь стратегию выталкивания страниц, которая характеризовалась бы

малыми издержками и не являлась бы дискриминационной по отношению к каким-либо конкретным пользователям, то можно пойти по очень простому пути – выбирать случайную страницу. В этом случае все страницы, находящиеся в основной памяти, могут быть выбраны для выталкивания с равной вероятностью, в том числе даже следующая страница к которой будет производиться обращение (и которую, естественно, удалять из памяти наиболее нецелесообразно). Поскольку подобная стратегия по сути как бы рассчитана на “слепое” везение, в реальных системах она применяется редко.

1.1.3.3 Выталкивание первой пришедшей страницы (FIFO)

При выталкивании страниц по принципу FIFO мы присваиваем каждой странице в момент поступления в основную память временную метку. Когда появляется необходимость удалить из основной памяти какую-нибудь страницу, мы выбираем ту, которая находилась в памяти дольше других. Интуитивный аргумент в пользу подобной стратегии кажется весьма весомым, а именно: у данной страницы уже были возможности “использовать свой шанс”, и пора дать подобные возможности и другой странице. К сожалению, стратегия FIFO с достаточно большой вероятностью будет приводить к замещению активно используемых страниц, поскольку тот факт, что страница находится в основной памяти в течение длительного времени, вполне может означать, что она постоянно в работе. Например, для крупных систем разделения времени стандартна ситуация, когда многие пользователи во время ввода и отработки своих программ совместно используют одну копию текстового редактора. Если в подобной системе выталкивать страницы по принципу FIFO, это может привести к удалению из памяти какой-либо интенсивно используемой странице редактора. А это будет безусловно нецелесообразно, поскольку её почти немедленно придется снова переписывать в основную память.

1.1.3.4 Выталкивание дольше всего не использовавшейся страницы (LRU)

Эта стратегия предусматривает, что для выталкивания следует выбирать ту страницу, которая не использовалась дольше других. Здесь мы исходим из эвристического правила, говорящего о том, что недавнее прошлое – хороший ориентир для прогнозирования ближайшего будущего. Стратегия LRU требует, чтобы при каждом обращении к странице ее временная метка обновлялась. Это может быть сопряжено с существенными издержками, и поэтому стратегия LRU, хотя она и кажется весьма привлекательной, в современных системах реализуется редко. Чаще применяются близкие к LRU стратегии, для которых характерны меньшие издержки.

Разработчики операционных систем должны всегда с большой осторожностью применять эвристические правила и рассуждения. Например, при реализации стратегии LRU может быть так, что страница, к которой дольше всего не было обращений, в действительности станет следующей используемой страницей, если программа к этому моменту очередной раз пройдет большой цикл, охватывающий несколько страниц. Таким образом, выталкивая страницу, к которой дольше всего не было обращений, мы можем оказаться вынужденными почти немедленно возвращать её обратно.

1.1.3.5 Выталкивание реже всего используемой страницы (LFU)

Одной из близких к LRU стратегий является стратегия, согласно которой выталкивается наименее часто (наименее интенсивно) использовавшаяся страница (LFU). Здесь мы контролируем интенсивность использования каждой страницы. Выталкивается та страница, которая наименее интенсивно используется или обращения к которой наименее часты. Подобный подход опять-таки кажется интуитивно оправданным, однако в то же время велика вероятность того, что удаляемая страница будет выбрана нерационально. Например,

наименее интенсивно используемой может оказаться та страница, которую только что переписали в основную память и к которой успели обратиться только один раз, в то время как к другим страницам могли уже обращаться более одного раза. Теперь работающий по принципу LFU механизм вытолкнет эту страницу, а она, скорее всего, сразу же будет использоваться.

Таким образом, практически любой метод выталкивания страниц, по-видимому, не исключает опасности принятия нерациональных решений. Это действительно так просто потому, что мы не можем достаточно точно прогнозировать будущее. В связи с этим необходима такая стратегия выталкивания страниц, которая обеспечивала бы принятие рациональных решений в большинстве случаев и в то же время не требовала больших накладных расходов.

1.1.3.6 Выталкивание не использовавшейся в последнее время страницы (NUR)

Один из распространенных алгоритмов, близких к стратегии LRU и характеризующихся малыми задержками, – это алгоритм выталкивания страницы, не использовавшейся в последнее время (NUR) к страницам, которая в последнее время не использовалась, вряд ли будут обращения и в ближайшем будущем, так что их можно заменять на вновь поступающие страницы.

Поскольку желательно заменять ту страницу, которая в период нахождения в основной памяти не изменялась, реализация стратегии предусматривает введение двух аппаратных битов – признаков на страницу. Это

$$\begin{aligned} \text{а) бит-признак обращения} &= \begin{cases} 0, & \text{если к странице не было обращений} \\ 1, & \text{если к странице были обращения} \end{cases} \\ \text{б) бит-признак модификации} &= \begin{cases} 0, & \text{если страница не изменялась} \\ 1, & \text{если страница изменялась} \end{cases} \end{aligned}$$

Бит-признак модификации часто называют так же “признаком записи” в страницу. Стратегия NUR реализуется следующим образом. Первоначально биты-признаки обращения и модификации для всех страниц устанавливаются в 0. При обращении к какой-либо странице её бит-признак обращения устанавливается в 1, а в случае изменения содержимого страницы устанавливается в 1 её бит-признак модификации. Когда нужно выбрать страницу для выталкивания, прежде всего мы пытаемся найти такую страницу, к которой не было обращений (поскольку мы стремимся приблизиться к алгоритму LRU). В противном случае у нас не будет другого выхода, как вытолкнуть страницу, к которой были обращения. Если к странице обращения были, мы проверяем, подверглась ли она изменению или нет. Если нет, мы заменяем ее из тех соображений, что это связано с меньшими затратами, чем в случае замены модифицированной страницы, которую необходимо будет физически переписывать во внешнюю память. В противном случае нам придется заменять модифицированную страницу.

Во многих системах основная память работает активно, так что рано или поздно у большинства страниц бит-признак обращения будет установлен в 1, и мы не сможем отличать те страницы, которые вытолкнуть наиболее целесообразно. Один из широко распространенных способов решения этой проблемы заключается в том, что все биты-признаки обращений периодически сбрасываются в 0, с тем, чтобы механизм выталкивания оказался в исходном состоянии, а затем снова разрешается установка этих битов-признаков в 1 обычным образом при обращениях. Правда, в этом случае существует опасность того, что могут быть вытолкнуты даже активные страницы, однако только в течение короткого периода после сброса битов-признаков, поскольку почти немедленно биты-признаки обращений для этих страниц будут снова установлены в 1.

Описанный выше алгоритм NUR предусматривает существование четырех групп

страниц:

- группа 1- обращений не было, модификаций не было
- группа 2- обращений не было, модификация была
- группа 3- обращения были, модификаций не было
- группа 4- обращения были, модификация была

Страницы групп с меньшими номерами следует выталкивать в первую очередь, а с большими – в последнюю. Отметим, что группа 2 означает на первый взгляд нереальную ситуацию, она включает страницы, к которым как бы не было обращений, но они оказались модифицированными. В действительности это просто результат того, что биты-признаки обращения (но не биты-признаки модификации) периодически сбрасываются, т.е. подобная ситуация вполне возможна.

1.3.7 FIFO с “пристрастием” (BIFO)

В системах с разделением времени часто используется стратегия BIFO (Biased FIFO), при которой каждому активному заданию i выделен набор из $n(i)$ страниц, таких, что $n(i) \neq n(j)$ при $i \neq j$. В пределах каждой группы из $n(i)$ страниц применяется стратегия FIFO. Через каждые t единиц времени значение $n(i)$ для всех i изменяются. Таким образом, в каждый момент времени операционная система покровительствует одному или нескольким заданиям.

1.1.3.8 Рабочие множества

Рабочее множество – это подмножество страниц, к которым процесс активно обращается.

Для обеспечения эффективного выполнения программы необходимо, чтобы её рабочее множество находилось в первичной памяти. В противном случае может возникнуть режим чрезмерно интенсивной подкачки страниц, так называемое пробуксовывание, поскольку программа будет многократно подкачивать одни и те же страницы из внешней памяти. Новые процессы можно инициировать только в случае, если в ОП имеется свободное место для размещения их рабочих множеств. В данном методе выталкивается та страница, которая не входит в рабочее множество.

1.2.Задание на лабораторную работу

Разработать программу, моделирующую работу двух заданных алгоритмов замещения страниц (таблица 5.1), реализации алгоритмов должны быть оформлены в виде подпрограмм. Считать, что виртуальная память имеет объем 1000 страниц. В оперативной памяти имеется 10 страниц, о которых известна вся необходимая для работы алгоритмов информация. Главная программа должна вводить исходные данные, вызывать подпрограммы, реализующие алгоритмы, и печатать результаты.

Исходные данные к программе:

- объём оперативной памяти в страницах ($V_{оп}$, к примеру, 10 страниц);
- состояние оперативной памяти (какие страницы в ней находятся)
- последовательность обращений к страницам **REF(1), REF(2), ..., REF(N)**, (можно генерировать случайным образом, НО следует обратить внимание на то, что оба алгоритма должны получать одинаковые исходные данные).

Результаты:

- состояние оперативной памяти после выполнения обращений в моменты времени t_1, \dots, t_N ;
- число страничных прерываний из-за отсутствия необходимой страницы ($N_{стр}$).

Замечание 1.

В алгоритме NUR дополнительно необходимо определить бит-признак модификации для каждой страницы **REF(i)**.

Замечание 2.

В алгоритме BIFO последовательность обращений к страницам следует определить для каждого задания **J₁, ..., J_M**.

Замечание 3.

В алгоритме “рабочие множества” для каждого задания **J_i** необходимо дополнительно задать подмножество страниц **АКТ_i**, к которым задание **J_i** активно обращается.

Таблица 1.1. - Варианты задания

№ вар.	Алгоритмы замещения страниц	№ вар.	Алгоритмы замещения страниц
1	RANDOM, BIFO	9	LFU, LRU
2	RANDOM, NUR	10	LFU, BIFO
3	FIFO, NUR	11	LFU, NUR
4	FIFO, BIFO	12	Рабочие множества, RANDOM
5	FIFO, LFU	13	Рабочие множества, FIFO
6	LRU, NUR	14	Рабочие множества, LRU
7	LRU, BIFO	15	Рабочие множества, LFU
8	LRU, LFU		

2. ЛАБОРАТОРНАЯ РАБОТА №6. ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ОС WINDOWS

Цель работы: изучение возможностей интерпретатора командной строки ОС Windows, приобретение практических навыков работы в командном режиме.

2.1. Основные теоретические положения

2.1.1. Запуск оболочки командной строки

Во всех версиях ОС Windows поддерживается интерактивная оболочка командной строки (command shell) и определенный набор утилит (количество и состав этих утилит зависит от версии ОС). Начиная с версии Windows NT, оболочка командной строки представляется интерпретатором Cmd.exe.

Запуск командного интерпретатора (открытия нового сеанса командной строки) можно выполнить разными способами, например:

через меню «Пуск»: «Пуск» - «Все приложения»- «Стандартные»-«Командная строка»;
комбинацией клавиш WINDOWS+X (в WINDOWS 10);

в окне ПОИСК набрать cmd;

в окне «Выполнить» (Windows+R) набрать cmd.

Единственный из способов, который не позволяет запустить командную строку от имени администратора, – это использование команды «Выполнить». В результате откроется окно, представленное на рисунке 2.1.

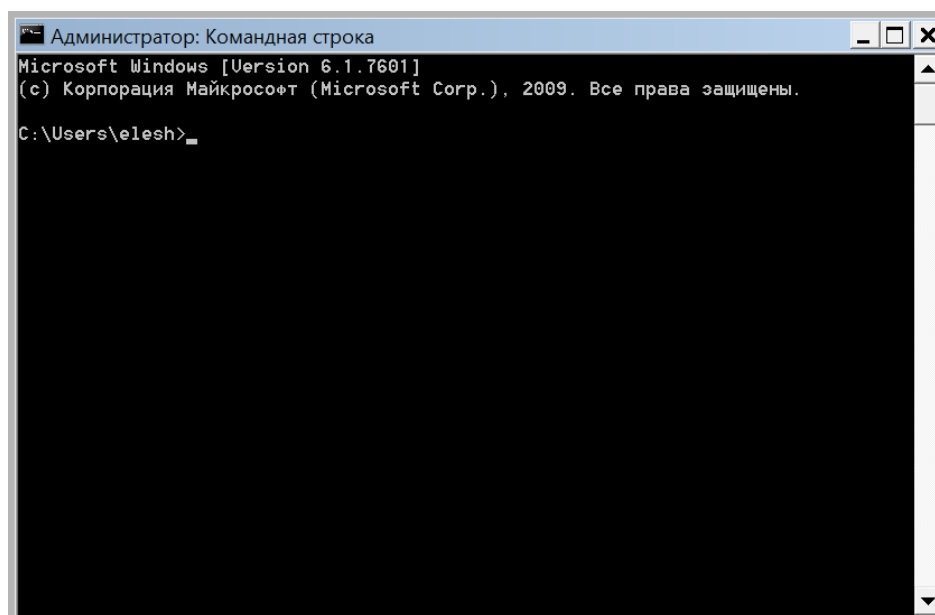


Рисунок 2.1 - Командное окно интерпретатора Cmd.exe в Windows 7

2.1.2. Команды

Все команды делятся на две группы: внутренние и внешние. Команды, которые распознаются и выполняются непосредственно самим командным интерпретатором, команды называются **внутренними** (например, COPY или DIR). Другие команды ОС представляют собой отдельные программы, расположенные по умолчанию в том же каталоге, что и Cmd.exe, которые Windows загружает и выполняет аналогично другим программам. Такие команды называются внешними (например, MORE или XCOPY).

Для того, чтобы выполнить команду, после приглашения командной строки (например, `C:\>`) следует ввести имя этой команды (регистр не важен), параметры и ключи (если они необходимы) и нажать клавишу `<Enter>`. Например: `C:\>COPY C:\myfile.txt A:\ /V`
Здесь `COPY` — имя команды здесь, `C:\myfile.txt` и `A:\` — параметры, а `/V` является ключом. Отметим, что в некоторых командах ключи могут начинаться не с символа `/`, а с символа `-` (минус), например, `-V`.

Многие команды Windows имеют большое количество дополнительных параметров и ключей. Большинство команд снабжено встроенной справкой, для доступа к которой следует ввести команду с ключом `/?`.

Для некоторых команд текст справки может быть довольно большим и не уместиться на одном экране. В этом случае информацию можно выводить последовательно по одному экрану с помощью команды `MORE` и символа конвейеризации `|`, например:

`XCOPY /? | MORE`

В этом случае после заполнения очередного экрана вывод помощи будет прерываться до нажатия любой клавиши. Кроме того, используя символы перенаправления вывода `>` и `>>`, можно текст, выводимый на экран, направить в текстовый файл для дальнейшего просмотра. Например, для вывода текста справки по команде `XCOPY` в текстовый файл `xcopy.txt`, используется следующая команда:

`XCOPY /? > XCOPY.TXT`

В командах вместо имени файла можно указывать обозначения устройств компьютера. В ОС Windows поддерживаются следующие **имена устройств**: `PRN` (принтер), `CON` (терминал: при вводе — это клавиатура, при выводе — монитор), `NUL` (пустое устройство, все операции ввода/вывода для него игнорируются).

2.1.3 Перенаправление ввода/вывода и конвейеризация команд

С помощью переназначения устройств ввода/вывода одна программа может направить свой вывод на вход другой или перехватить вывод другой программы, используя его в качестве своих входных данных. Таким образом, имеется возможность передавать информацию от процесса к процессу при минимальных программных издержках. Практически это означает, что для программ, которые используют стандартные входные и выходные устройства, ОС позволяет:

выводить сообщения программ не на экран (стандартный выходной поток), а в файл или на принтер (перенаправление вывода);

читать входные данные не с клавиатуры (стандартный входной поток), а из заранее подготовленного файла (перенаправление ввода);

передавать сообщения, выводимые одной программой, в качестве входных данных для другой программы (конвейеризация или композиция команд).

Из командной строки эти возможности реализуются следующим образом. Для того, чтобы перенаправить текстовые сообщения, выводимые какой-либо командой, в текстовый файл, нужно использовать конструкцию

команда `>` имя_файла

Если при этом заданный для вывода файл уже существовал, то он перезаписывается, если не существовал — создается. Можно также не создавать файл заново, а *дописывать* информацию, выводимую командой, в конец существующего файла. Для этого команда перенаправления вывода должна быть задана в виде:

команда `>>` имя_файла

С помощью символа `<` можно прочитать входные данные для заданной команды не с клавиатуры, а из заранее подготовленного файла:

команда `<` имя_файла

Рассмотрим примеры перенаправления ввода/вывода.

1) Вывод встроенной справки для команды COPY в файл copy.txt:

`COPY /? > copy.txt`

2) Добавление текста справки для команды XCOPY в файл copy.txt:

`XCOPY /? >> copy.txt`

3) Вывод текущей даты в файл date.txt (`DATE /T` — это команда для просмотра и изменения системной даты, `T` - ключ для получения только даты без запроса нового значения):

`DATE /T > date.txt`

Если при выполнении определенной команды возникает ошибка, то сообщение об этом выводится на экран. В случае необходимости сообщения об ошибках (стандартный поток ошибок) можно перенаправить в текстовый файл с помощью конструкции

команда `2>` имя_файла

В этом случае стандартный вывод будет производиться на экран.

Также имеется возможность информационные сообщения и сообщения об ошибках выводить в один и тот же файл. Делается это следующим образом:

команда `>` имя_файла `2>&1`

Например, в приведенной ниже команде стандартный выходной поток и стандартный поток ошибок перенаправляются в файл copy.txt:

`XCOPY A:\1.txt C: > copy.txt 2>&1`

С помощью конструкции команда1 | команда2 можно использовать сообщения, выводимые первой командой, в качестве входных данных для второй команды (конвейер команд).

Используя механизмы перенаправления ввода/вывода и конвейеризации, можно из командной строки посылать информацию на различные устройства и автоматизировать ответы на запросы, выдаваемые командами или программами, использующими стандартный ввод. Для решения таких задач служит команда

`ECHO [сообщение]`, которая выводит сообщение на экран. Пример использования этой команды.

Удаление всех файлов в текущем каталоге без предупреждения (автоматический положительный ответ на запрос об удалении):

`ECHO y | DEL *.*`

2.1.4. Команды MORE и SORT

Одной из наиболее часто используемых команд, для работы с которой применяется перенаправление ввода/вывода и конвейеризация, является `MORE`. Эта команда считывает стандартный ввод из конвейера или перенаправленного файла и выводит информацию частями, размер каждой из которых не больше размера экрана. Используется `MORE` обычно для просмотра длинных файлов. Возможны три варианта синтаксиса этой команды:

`MORE [диск:][путь]имя_файла`

`MORE < [диск:][путь]имя_файла`

имя_команды | `MORE`

Параметр `[диск:] [путь]имя_файла` определяет расположение и имя файла с просматриваемыми на экране данными. Параметр `имя_команды` задает команду, вывод которой отображается на экране (например, `DIR` или команда `TYPE`, используемая для вывода содержимого текстового файла на экран). Приведем два примера.

Для поэкранного просмотра текстового файла news.txt возможны следующие варианты команд:

`MORE news.txt`

`MORE < news.txt`

`TYPE news.txt | MORE`

Другой распространенной командой, использующей перенаправление ввода/вывода и конвейеризацию, является `SORT`. Эта команда работает как фильтр: она считывает символы в

заданном столбце, упорядочивает их в возрастающем или убывающем порядке и выводит отсортированную информацию в файл, на экран или другое устройство. Возможны два варианта синтаксиса этой команды:

```
SORT [/R] [/+n] [[диск1:][путь1]файл1] [> [диск2:][путь2]файл2]
```

или

```
[команда] SORT [/R] [/+n] [> [диск2:][путь2]файл2]
```

В первом случае параметр [диск1:] [путь1] файл1 определяет имя файла, который нужно отсортировать. Во втором случае будут отсортированы выходные данные указанной команды. Если параметры файл1 или команда не заданы, то SORT будет считывать данные с устройства стандартного ввода.

Параметр [диск2:] [путь2] файл2 задает файл, в который будет направляется сортированный вывод; если этот параметр не задан, то вывод будет направлен на устройство стандартного вывода.

По умолчанию сортировка выполняется в порядке возрастания. Ключ /R позволяет изменить порядок сортировки на обратный (от Z к A и затем от 9 до 0). Например, для постраничного просмотра отсортированного в обратном порядке файла price.txt, нужно задать следующую команду:

```
SORT /R < price.txt |MORE
```

Ключ /+n задает сортировку в файле по символам n-го столбца. Например, /+10 означает, что сортировка должна осуществляться, начиная с 10-й позиции в каждой строке. По умолчанию файл сортируется по первому столбцу.

2.1.5 Условное выполнение и группировка команд

В командной строке Windows можно использовать специальные символы, которые позволяют вводить несколько команд одновременно и управлять работой команд в зависимости от результатов их выполнения. С помощью таких символов условной обработки можно содержание небольшого пакетного файла записать в одной строке и выполнить полученную составную команду.

Используя символ амперсанда &, можно разделить несколько утилит в одной командной строке, при этом они будут выполняться друг за другом. Например, если набрать команду DIR & PAUSE & COPY /? и нажать клавишу <Enter>, то вначале на экран будет выведено содержимое текущего каталога, а после нажатия любой клавиши — встроенная справка команды COPY.

Условная обработка команд в Windows осуществляется с помощью символов && и || следующим образом. Двойной амперсанд && запускает команду, стоящую за ним в командной строке, только в том случае, если команда, стоящая перед амперсандами была выполнена успешно. Например, если в корневом каталоге диска C: есть файл plan.txt, то выполнение строки TYPE C:\plan.txt && DIR приведет к выводу на экран этого файла и содержимого текущего каталога. Если же файл C:\plan.txt не существует, то команда DIR выполняться не будет.

Два символа || осуществляют в командной строке обратное действие, т.е. запускают команду, стоящую за этими символами, только в том случае, если команда, идущая перед ними, не была успешно выполнена. Таким образом, если в предыдущем примере файл C:\plan.txt будет отсутствовать, то в результате выполнения строки TYPE C:\plan.txt || DIR на экран выведется содержимое текущего каталога.

Отметим, что условная обработка действует только на ближайшую команду, то есть в строке

```
TYPE C:\plan.txt && DIR & COPY /?
```

команда COPY /? запустится в любом случае, независимо от результата выполнения команды TYPE C:\plan.txt.

Несколько утилит можно сгруппировать в командной строке с помощью *круглых скобок*. **Рассмотрим две строки:**

```
TYPE C:\plan.txt && DIR & COPY /?
```

```
TYPE C:\plan.txt && (DIR & COPY /?)
```

В первой из них символ условной обработки && действует только на команду DIR, во второй — одновременно на две команды: DIR и COPY.

2.1.6. Команды для работы с файловой системой

Рассмотрим некоторые наиболее часто используемые команды для работы с файловой системой. Отметим сначала несколько особенностей определения путей к файлам в Windows.

2.1.6.1 Пути к объектам файловой системы

Файловая система логически имеет древовидную структуру и имена файлов задаются в формате [диск:] [путь\] имя_файла, то есть обязательным параметром является только имя файла. При этом, если путь начинается с символа "\", то маршрут вычисляется от корневого каталога, иначе — от текущего каталога. Например, имя C:123.txt задает файл 123.txt в текущем каталоге на диске C:, имя C:\123.txt — файл 123.txt в корневом каталоге на диске C:, имя ABC\123.txt — файл 123.txt в подкаталоге ABC текущего каталога.

Существуют особые обозначения для текущего каталога и родительского каталогов. Текущий каталог обозначается символом . (точка), его родительский каталог — символами .. (две точки). Например, если текущим каталогом является C:\WINDOWS, то путь к файлу autoexec.bat в корневом каталоге диска C: может быть записан в виде ..\autoexec.bat.

В именах файлов (но не дисков или каталогов) можно применять так называемые **групповые символы** или шаблоны: ? (вопросительный знак) и * (звездочка). Символ * в имени файла означает произвольное количество любых допустимых символов, символ ? — один произвольный символ или его отсутствие. Скажем, под шаблон text??1.txt подходят, например, имена text121.txt и text11.txt, под шаблон text*.txt — имена text.txt, textab12.txt, а под шаблон text.* — все файлы с именем text и произвольным расширением.

Для того, чтобы использовать длинные имена файлов при работе с командной строкой, их нужно заключать в двойные кавычки. Например, чтобы запустить файл с именем 'Мое приложение.exe' из каталога 'Мои документы', нужно в командной строке набрать "C:\Мои документы\Мое приложение.exe" и нажать клавишу <Enter>.

2.1.6.2 Команда CD

Текущий каталог можно изменить с помощью команды CD [диск:][путь\]. Путь к требуемому каталогу указывается с учетом приведенных выше замечаний. Например, команда CD \ выполняет переход в корневой каталог текущего диска. Если запустить CD без параметров, то на экран будут выведены имена текущего диска и каталога.

2.1.6.3 Команда COPY

Одной из наиболее часто повторяющихся задач при работе на компьютере является копирование и перемещение файлов из одного места в другое. Для копирования одного или нескольких файлов используется команда COPY.

Синтаксис этой команды:

```
COPY [/A/B] источник [/A/B] [+ источник [/A/B] [+ ...]]
```

```
[результат [/A/B]] [/V][Y|—Y]
```

Параметры, ключи и их описание для команды COPY представлены в таблице 6.1

Таблица 1.1-Параметры и ключи команды COPY

Параметр	Описание
источник	Имя копируемого файла или файлов
/A	Файл является текстовым файлом ASCII, то есть конец файла обозначается символом с кодом ASCII 26 (<Ctrl>+<Z>)
/B	Файл является двоичным. Этот ключ указывает на то, что интерпретатор команд должен при копировании считывать из источника число байт, заданное размером в каталоге копируемого файла
результат	Каталог для размещения результата копирования и/или имя создаваемого файла
/V	Проверка правильности копирования путем сравнения файлов после копирования
/Y	Отключение режима запроса подтверждения на замену файлов
/-Y	Включение режима запроса подтверждения на замену файлов

Примеры использования команды COPY.

Копирование файла abc.txt из текущего каталога в каталог D:\PROGRAM под тем же именем:

`COPY abc.txt D:\PROGRAM`

Копирование файла abc.txt из текущего каталога в каталог D:\PROGRAM под новым именем def.txt:

`COPY abc.txt D:\PROGRAM\def.txt`

Копирование всех файлов с расширением txt с диска A: в каталог 'Мои документы' на диске C:

`COPY A:*.txt "C:\Мои документы"`

Если не задать в команде целевой файл, то команда COPY создаст копию файла-источника с тем же именем, датой и временем создания, что и исходный файл, и поместит новую копию в текущий каталог на текущем диске. Например, для того, чтобы скопировать все файлы из корневого каталога диска A: в текущий каталог, достаточно выполнить такую команду:

`COPY A:*.*`

Пример 1. Создание нового текстового файла и запись в него информации без использования текстового редактора.

Для решения задачи необходимо ввести команду `COPY CON my.txt`, которая копирует вводимые с клавиатуры символы в файл my.txt (если этот файл существовал, то он перезапишется, иначе — создастся). Для завершения ввода необходимо ввести символ конца файла нажатием клавиш <Ctrl>+<Z>.

Команда COPY может также объединять (склеивать) несколько файлов в один. Для этого необходимо указать единственный результирующий файл и несколько исходных. Это достигается путем использования групповых знаков (? и *) или формата файл1 + файл2 + файл3. Например, для объединения файлов 1.txt и 2.txt в файл 3.txt можно задать следующую команду:

`COPY 1.txt+2.txt 3.txt`

Объединение всех файлов с расширением dat из текущего каталога в один файл all.dat может быть произведено так:

`COPY /B *.dat all.dat`

Ключ /B здесь используется для предотвращения усечения соединяемых файлов, так как при комбинировании файлов команда COPY по умолчанию считает файлы текстовыми.

Если имя целевого файла совпадает с именем одного из копируемых файлов (кроме первого), то исходное содержимое целевого файла теряется. Если имя целевого файла опущено, то в его качестве используется первый файл из списка.

Например, команда `COPY 1.txt+2.txt` добавит к содержимому файла 1.txt содержимое файла 2.txt. Командой `COPY` можно воспользоваться и для присвоения какому-либо файлу **текущей даты и времени** без модификации его содержимого. Для этого нужно ввести команду

`COPY /B 1.txt +,,`

Здесь запятые указывают на пропуск параметра приемника, что и приводит к требуемому результату.

Команда `COPY` имеет ряд особенностей: с ее помощью нельзя копировать скрытые и системные файлы, файлы нулевой длины, файлы из подкаталогов. Кроме того, если при копировании группы файлов `COPY` встретит файл, который в данный момент нельзя скопировать (например, он занят другим приложением), то процесс копирования прервется, и остальные файлы не будут скопированы.

2.1.6.4 Команда `XCOPY`

Указанные при описании команды `COPY` проблемы можно решить с помощью команды `XCOPY`, которая предоставляет намного больше возможностей при копировании. Однако, `XCOPY` может работать только с файлами и каталогами, но не с **устройствами**.

Синтаксис команды: `XCOPY` источник [результат] [ключи]

Команда `XCOPY` имеет множество ключей, далее приведены лишь некоторых из них. Ключ `/D[:[дата]]` позволяет копировать только файлы, измененные не ранее указанной даты. Если параметр дата не указан, то копирование будет производиться только если источник новее результата. Например, команда `XCOPY "C:\Мои документы*.*" "D:\BACKUP\Мои документы" /D` скопирует в каталог 'D:\BACKUP\Мои документы' только те файлы из каталога 'C:\Мои документы', которые были изменены со времени последнего подобного копирования или которых вообще не было в 'D:\BACKUP\Мои документы'.

Ключ `/S` позволяет копировать все непустые подкаталоги в каталоге-источнике. С помощью же ключа `/E` можно копировать вообще все подкаталоги, включая и пустые.

Если указан ключ `/C`, то копирование будет продолжаться даже в случае возникновения ошибок. Это бывает очень полезным при операциях копирования, производимых над группами файлов, например, при резервном копировании данных.

Ключ `/I` важен для случая, когда копируются несколько файлов, а файл назначения отсутствует. При задании этого ключа команда `XCOPY` считает, что файл назначения должен быть каталогом. Например, если задать ключ `/I` в команде копирования всех файлов с расширением txt из текущего каталога в несуществующий еще подкаталог TEXT, `XCOPY *.txt TEXT /I` то подкаталог TEXT будет создан без дополнительных запросов.

Ключи `/Q`, `/F` и `/L` отвечают за режим отображения при копировании. При задании ключа `/Q` имена файлов при копировании не отображаются, ключа `/F` — отображаются полные пути источника и результата. Ключ `/L` обозначает, что отображаются только файлы, которые должны быть скопированы (при этом само копирование не производится).

С помощью ключа `/H` можно копировать скрытые и системные файлы, а с помощью ключа `/R` — заменять файлы с атрибутом "Только для чтения". Например, для копирования всех файлов из корневого каталога диска C: (включая системные и скрытые) в каталог SYS на диске D:, нужно ввести следующую команду:

`XCOPY C:*.* D:\SYS /H`

Ключ `/T` позволяет применять `XCOPY` для копирования только структуры каталогов источника, без дублирования находящихся в этих каталогах файлов, причем пустые каталоги и подкаталоги не включаются. Для того, чтобы все же включить пустые каталоги и подкаталоги, нужно использовать комбинацию ключей `/T /E`.

Используя XCOPY можно при копировании обновлять только уже существующие файлы (новые файлы при этом не записываются). Для этого применяется ключ /U. Например, если в каталоге C:\2 находились файлы a.txt и b.txt, а в каталоге C:\1 — файлы a.txt, b.txt, c.txt и d.txt, то после выполнения команды:

```
XCOPY C:\1 C:\2 /U
```

в каталоге C:\2 по-прежнему останутся лишь два файла a.txt и b.txt, содержимое которых будет заменено содержимым соответствующих файлов из каталога C:\1. Если с помощью XCOPY копировался файл с атрибутом "Только для чтения", то по умолчанию у файла-копии этот атрибут снимется. Для того, чтобы копировать не только данные, но и полностью атрибуты файла, необходимо использовать ключ /K.

Ключи /Y и /-Y определяют, нужно ли запрашивать подтверждение перед заменой файлов при копировании. /Y означает, что такой запрос нужен, /-Y — не нужен.

2.1.6.5. Команда DIR

Команда: DIR [диск:][путь][имя_файла] [ключи] используется для вывода информации о содержимом дисков и каталогов. Параметр [диск:][путь] задает диск и каталог, содержимое которого нужно вывести на экран. Параметр [имя_файла] задает файл или группу файлов, которые нужно включить в список.

Например, команда DIR C:*.bat выведет на экран все файлы с расширением bat в корневом каталоге диска C:. Если задать эту команду без параметров, то выводится метка диска и его серийный номер, имена (в коротком и длинном вариантах) файлов и подкаталогов, находящихся в текущем каталоге, а также дата и время их последней модификации. После этого выводится число файлов в каталоге, общий объем (в байтах), занимаемый файлами, и объем свободного пространства на диске.

Например:

```

Том в устройстве C имеет метку PHYS1_PART2
Серийный номер тома: 366D-6107
Содержимое папки C:\aditor
.      <ПАПКА>    25.01.15  17:15 .
..     <ПАПКА>    25.01.15  17:15 ..
HILITE DAT       1 082 18.09.16      18:55 hilite.dat
TEMPLT01 DAT     48 07.08.16      1:00 templt01.dat
TTABLE DAT       357 07.08.16      1:00 ttable.dat
ADITOR EXE      461 312 01.12.15     23:13 aditor.exe
README TXT       3 974 25.01.15     17:26 readme.txt
ADITOR HLP       24 594 08.10.16     23:12 aditor.hlp
ТЕКСТО~1 TXT      0 11.03.15      9:02 Текстовый файл.txt
    11 файлов      533 647 байт
     2 папок      143 261 696 байт свободно
```

С помощью ключей команды DIR можно задать различные режимы расположения, фильтрации и сортировки. Например, при использовании ключа /W перечень файлов выводится в широком формате с максимально возможным числом имен файлов или каталогов на каждой строке.

С помощью ключа /A[:]атрибуты можно вывести имена только тех каталогов и файлов, которые имеют заданные атрибуты (R — "Только чтение", A — "Архивный", S — "Системный", H — "Скрытый", префикс "-" имеет значение НЕ). Если ключ /A используется более чем с одним значением атрибута, будут выведены имена только тех файлов, у которых все атрибуты совпадают с заданными. Например, для вывода имен всех файлов в корневом

каталоге диска C:, которые одновременно являются скрытыми и системными, можно задать команду

```
DIR C:\ /A:HS
```

а для вывода всех файлов, кроме скрытых — команду

```
DIR C:\ /A:-H
```

Отметим здесь, что атрибуту каталога соответствует буква D, и для того, чтобы, например, вывести список всех каталогов диска C:, нужно задать команду

```
DIR C: /A:D
```

Ключ /O[:]сортировка задает порядок сортировки содержимого каталога при выводе его командой DIR. Если этот ключ опущен, DIR печатает имена файлов и каталогов в том порядке, в котором они содержатся в каталоге. Если ключ /O задан, а параметр сортировка не указан, то DIR выводит имена в алфавитном порядке. В параметре сортировка можно использовать следующие значения: N — по имени (алфавитная), S — по размеру (начиная с меньших), E — по расширению (алфавитная), D — по дате (начиная с более старых), A — по дате загрузки (начиная с более старых), G — начать список с каталогов. Префикс "-" означает обратный порядок. Если задается более одного значения порядка сортировки, файлы сортируются по первому критерию, затем по второму и т.д.

Ключ /S означает вывод списка файлов из заданного каталога и его подкаталогов. Ключ /B перечисляет только названия каталогов и имена файлов (в длинном формате) по одному на строку, включая расширение. При этом выводится только основная информация. Например:

```
templt02.dat
UNINST1.000
hilite.dat
templt01.dat
UNINST0.000
ttable.dat
aditor.exe
readme.txt 23
aditor.hlp
Текстовый файл.txt
```

2.1.6.6 Команды MKDIR и RMDIR

Для создания нового каталога и удаления уже существующего пустого каталога используются команды MKDIR [диск:]путь и RMDIR [диск:]путь [ключи] соответственно (или их короткие аналоги MD и RD). Например:

```
MKDIR "C:\Примеры"
```

```
RMDIR "C:\Примеры"
```

Команда MKDIR не может быть выполнена, если каталог или файл с заданным именем уже существует. Команда RMDIR не будет выполнена, если удаляемый каталог не пустой.

2.1.6.7 Команда DEL

Удалить один или несколько файлов можно с помощью команды

```
DEL [диск:][путь]имя_файла [ключи]
```

Для удаления сразу нескольких файлов используются групповые знаки ? и *. Ключ /S позволяет удалить указанные файлы из всех подкаталогов, ключ /F — принудительно удалить файлы, доступные только для чтения, ключ /A[:]атрибуты — отбирать файлы для удаления по атрибутам (аналогично ключу /A[:]атрибуты] в команде DIR).

2.1.6.8 Команда REN

Переименовать файлы и каталоги можно с помощью команды RENAME (REN).
Синтаксис этой команды имеет следующий вид:

REN [диск:][путь][каталог1|файл1] [каталог2|файл2]

Здесь параметр каталог1|файл1 определяет название каталога/файла, которое нужно изменить, а каталог2|файл2 задает новое название каталога/файла. В любом параметре команды REN можно использовать групповые символы ? и *. При этом представленные шаблонами символы в параметре файл2 будут идентичны соответствующим символам в параметре файл1. Например, чтобы изменить у всех файлов с расширением txt в текущей директории расширение на doc, нужно ввести такую команду:

REN *.txt *.doc

Если файл с именем файл2 уже существует, то команда REN прекратит выполнение, и произойдет вывод сообщения, что файл уже существует или занят. Кроме того, в команде REN нельзя указать другой диск или каталог для создания результирующих каталога и файла. Для этой цели нужно использовать команду MOVE, предназначенную для переименования и перемещения файлов и каталогов.

2.1.6.9 Команда MOVE

Синтаксис команды для перемещения одного или более файлов имеет вид:

MOVE [/Y|/Y] [диск:][путь]имя_файла1[,...] результирующий_файл

Синтаксис команды для переименования папки имеет вид:

MOVE [/Y|/Y] [диск:][путь]каталог1 каталог2

Здесь параметр результирующий_файл задает новое размещение файла и может включать имя диска, двоеточие, имя каталога, либо их сочетание. Если перемещается только один файл, допускается указать его новое имя. Это позволяет сразу переместить и переименовать файл. Например, MOVE "C:\Мои документы\список.txt" D:\list.txt.

Если указан ключ /-Y, то при создании каталогов и замене файлов будет выдаваться запрос на подтверждение. Ключ /Y отменяет выдачу такого запроса.

2.2. Задание на лабораторную работу

1. Изучить теоретический материал.
2. Запустить интерпретатор командной строки
3. Увеличить размер окна интерпретатора и задать цвет фона и цвет шрифта (Для этого следует использовать пункт Свойства управляющего меню окна, рекомендуется синий фон и белый шрифт). Выполнить все примеры из п 2.1
4. Создать текстовый файл, содержащий справочные сведения по командам DIR, COPY и XCOPY.
5. Вывести содержимое указанного в таблице 2.2 каталога по указанному формату на экран и в файл.

Замечание 1. При создании текстового файла интерпретатор командной строки использует кодировку **кириллица (DOS)**. Поэтому рекомендуется переназначить вывод в файл с расширением **.txt**, а для просмотра содержимого файла использовать Internet Explorer, указав вид кодировки **кириллица (DOS)**. Пример вывода содержимого текстового файла приведен на рис. 3.

Замечание 2. Интерпретатор хранит историю введенных команд в буфере (размером 50 строк). Для просмотра содержимого буфера используйте клавиши клавиатуры СТРЕЛКА

ВВЕРХ и СТРЕЛКА ВНИЗ. Полученную команду можно отредактировать и выполнить снова.

Таблица 2.2.-Варианты индивидуальных заданий

Варианты заданий для бригад Номера бригад	Имя каталога	Что выводить	Сортировать по	Атрибуты фай-лов и каталогов
1	Студент	Только файлы	По размеру	Системный
2	Студент	Файлы и подкаталоги	По дате	Скрытый
3	Студент	Только подкаталоги	По имени	Только чтение
4	Студент	Только файлы bmp	По размеру	Только чтение
5	Студент	Только файлы jpg	По имени	Любые
6	Студент	Только подкаталоги	По дате	Любые
7	Студент	Файлы и подкаталоги	По размеру	Любые
8	Студент	Только файлы pdf	По имени	Только чтение

3. ЛАБОРАТОРНАЯ РАБОТА №7. УПРАВЛЕНИЕ ФАЙЛАМИ В ОС WINDOWS

Цель работы: изучение возможностей интерпретатора командной строки ОС Windows, приобретение практических навыков работы в командном режиме.

3.1. Часть 1. Командный язык ОС

Диалог с ОС осуществляется в форме команд. Команда состоит из имени команды и параметров, разделенных пробелами. Имя команды MS-DOS и параметры могут набираться как прописными, так и строчными латинскими буквами. Ввод каждой команды заканчивается нажатием клавиши [ENTER].

Когда MS-DOS готова к диалогу с пользователем выдается приглашение:

A:\>

C:\>

Основные параметры любой команды можно узнать с помощью встроенной справки:

<имя команды> /?

Cls – очистка содержимого экрана. По этой команде с экрана будет удалена вся информация, и останется одно приглашение операционной системы.

Ver – выводит на экран название и версию ОС.

Vol – выводит на экран метку и серийный номер тома для диска. Под томом подразумевается логический диск. Если жёсткий диск не разбит на логические диски, то подразумевается сам диск.

Date – вывод на экран текущей системной даты и приглашение для ввода новой даты. Дата вводится в том формате, который предлагает операционная система. Например, формат (дд-мм-гг) означает, что надо ввести арабскими цифрами день-месяц-год (например, 11-01-07). *Дата вводится без скобок!*

Time – вывод на экран текущего системного времени и приглашение для ввода нового времени в том формате, который предлагает операционная система. Практически достаточно ввести только часы и минуты через двоеточие (например, 12:45).

Некоторые команды имеют ключи, расширяющие возможности команды. Ключи записываются после имени команды и последующего за ним прямого слэша (символ /). Так, например, для всех команд существует ключ “?” (знак вопроса без кавычек). По данному ключу на экран выводится назначение команды, формат её записи и дополнительные ключи данной команды.

3.1.1. Работа с файлами и папками

Создание текстовых файлов: **copy con** (дискковод:) (\ путь\) имя_файла

После ввода команды построчно вводится текст, в конце каждой строки нажимается клавиша *Enter*, а после ввода последней строки – клавиши *F6* и затем *Enter*. Команда выведет сообщение: « 1 file(s) copied» (один файл скопирован) и на диске появится файл с указанным в команде именем.

Удаление файлов: **del** (дискковод:) (\ путь\) имя_файла

Перед удалением файла выводится имя файла и запрос «Delete (Y/N)?». При нажатии клавиши *Y* файл будет удалён, при нажатии *N* удаление отменяется.

Переименование файлов:

ren (дискковод:)(\ путь\) имя_файла_1 имя_файла_2

В результате выполнения команды имя файла_1 заменяется именем файла_2. Эта команда не обрабатывает файлы с атрибутом «скрытый».

Копирование файлов:

copy (дискковод:)(\путь\) **имя_файла_1** (дискковод:)(\путь\) **имя_файла_2**

В результате выполнения команды файл 1 копируется на место файла 2. Если файл с таким же именем, как у копии, уже существует, то он замещается без каких-либо предупреждений. Файлы с атрибутом «скрытый» не копируются. В этой команде вместо имён файлов можно использовать обозначения устройств *DOS* (клавиатуры, монитора, принтера). Если эта команда используется для объединения содержимого нескольких файлов, то вместо параметра «имя файла 1» перечисляются через знак «+» имена объединяемых файлов, а затем имя файла, в которое копируется содержимое объединяемых файлов.

Редактирование файла **edit** (дискковод:)(\путь\) **имя_файла**

Вывод файла на экран: **type** (дискковод:)(\путь\) **имя_файла**

Команда смены текущего дисквода: имя дисквода и двоеточие (**d:**)

Изменение текущего каталога: **cd** (дискковод:)(\путь\) **имя_каталога**

Если задан дискковод, то текущий каталог изменяется на этом дискводе, иначе - на текущем дискводе. Если в команде отсутствуют параметры, то в результате выполнения сообщаются текущие диск и каталог. Для перехода в каталог на другом диске нужно ввести команду перехода на другой диск и затем данную команду для изменения текущего каталога. Выход из каталога на уровень выше **cd..**

Просмотр каталога: **dir** (дискковод:)(\путь\) (**имя_файла**) (**параметры**)

Если имя файла не задано, то выводится всё оглавление каталога. Управление выводом сведений о каталоге осуществляется с помощью многочисленных параметров команды (например, **/P** – по-экранный вывод оглавления с паузами при заполнении экрана).

Параметры:

filename - имя файла или файлов;

/O:order - сортировка каталога: (**D** - сортировка по дате, **E** - сортировка по расширению; **N** - сортировка по имени; **S** - сортировка по размеру);

/B - вывести имена файлов и подкаталогов;

/P - поэкранный вывод;

/S - просмотр подкаталогов;

/W - вывод в широком формате.

Комментарий: ключ **/S** дает возможность искать файлы в пределах всего диска. Пример **DIR *.BAK /S /B** - вывод указанных файлов в пределах всего диска.

Создание каталога: **md** (дискковод:)(\путь\) **имя_каталога**

Удаление пустого каталога: **rd** (дискковод:)(\путь\) **имя_каталога**

Удаление каталога со всем содержимым:

rd /S (дискковод:)(\путь\) **имя_каталога**

Если параметр **/Q** не указан, то выдаётся запрос на подтверждение удаления каталога, иначе - удаление выполняется без запросов.

Переименование каталога: **ren** **имя_каталога** **новое_имя_каталога**

У старого имени можно указать диск и путь и тем самым переименовывать не только подкаталоги текущего каталога, но и других каталогов.

Структура папок: **tree** (дискковод:)(\путь\)- иерархическое отображение структуры папок заданного диска или заданной папки.

При подаче некоторых команд (**dir**, **copy**, **del**) в командной строке допускается использовать символы «?» и «*» (их можно использовать в имени и типе файла). При этом символ «?» будет рассматриваться как один из символов, допустимых в данной команде, а символ «*» заменяет собой произвольное число любых допустимых в данной команде символов. Например, командой «**del c:\dos\k1?.***» удалятся все файлы с любым расширением (в том числе и без него) в каталоге *DOS* на диске C:, имена которых состоят из 3-х символов (причем первые два – **k1**, а третий - произвольный из допустимых).

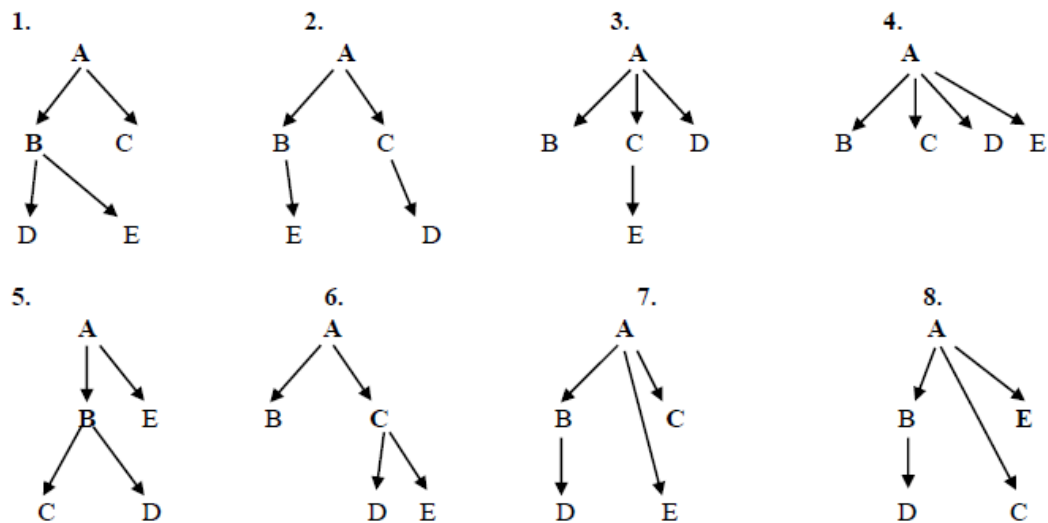
Вызов справки: *help*

Выводит справочную информацию о доступных внешних командах

3.2. Задание на лабораторную работу(часть 1)

1. Запустить командный интерпретатор.
1. Очистить экран (CLS).
- 2.1 Вывести на экран название и версию ОС.
- 2.2 Вывести на экран метки и серийные номера томов (логических дисков).
- 2.3 Вести настоящую дату и настоящее время и вывести их на экран, используя специальный ключ команды (определить его в результате запроса).
2. Через командную строку зайти в каталог Temp диска C.
3. В каталоге Temp создать дерево каталогов в соответствии с вариантом (рисунок 3.1). Отобразить графическую структуру на экране.
4. В каталоге A создать подкаталоги A1 и A2 , в каталоге B- подкаталог B2.
5. В каталоге B создать файл Name.txt, содержащий информацию о студентах (ФИО). Здесь же создать файл Date.txt, содержащий информацию о датах рождения студентов.
6. В каталоге C создать файл Name.txt, содержащий информацию о названии Вуза и специальность, на которой студент обучается. Здесь же создать файл Mark.txt с оценками на вступительных экзаменах и общей суммой баллов.
7. В каталоге E создать файл hobby.txt с информацией об увлечениях студентов.
8. Скопировать файл hobby.txt в каталог A2 и переименовать его в файл Lab_№варианта.txt.
9. Сделать копию файла Lab_№варианта.txt (например, copy_Lab_№варианта.txt) в этом же каталоге.
10. Вывести на экран информацию, хранящуюся в одном из файлов каталога C.
11. Отсортировать все файлы, хранящиеся в каталоге C, по имени.
12. Объединить все файлы, хранящиеся в каталоге C, в файл all.txt и вывести его содержимое на экран.
13. Отредактировать файл all.txt, добавив в него год вашего рождения, и вывести его содержимое на экран.
14. Скопировать файл all.txt в директорию D.
15. Удалить все директории, в названии которых есть буква A или цифра 2.
16. Удалить все оставшиеся файлы и папки.

Варианты иерархии каталогов



3.3. Часть 2. Теоретические сведения

Файл — это упорядоченная совокупность данных, хранящаяся на диске и занимающая поименованную область внешней памяти. Имя файла состоит из имени и расширения файла. Популярные форматы расширений в Windows:

- .txt, .doc – текстовые документы, в том числе документы приложения Word;
- .rar, .zip – архивные файлы;
- .exe, .com – программы и утилиты;
- .jpg, .bmp, .gif, .tiff – цифровые фотографии, изображения;
- .htm, .php – Web-страницы;
- .hlp – справочные файлы;
- .avi, .mpg, .wmv, .mkv – видеофайлы;
- .xls – электронные таблицы приложения Excel.

При установке или работе программ создаются временные файлы Windows с расширением .tmp. После завершения установки или работы такие временные файлы удаляются автоматически, но это происходит не всегда. Большое количество временных файлов может замедлять работу компьютера, поэтому их рекомендуется удалять вручную.

Для организации эффективной работы с данными, хранящимися в памяти, предназначена файловая система. Файловая система – это набор спецификаций и соответствующее им программное обеспечение, которое отвечает за создание, удаление, организацию, чтение, запись, модификацию и перемещение файлов информации, а также за управление доступом к файлам и за управление ресурсами, которые используются файлами. Файловая система определяет способ организации данных на диске и принципы хранения данных на физическом носителе. Файловая система должна обеспечивать:

- 1) безопасное и надежное хранение данных (т. е. защищенное от несанкционированного использования и различного рода сбоев и ошибок);
- 2) программный интерфейс доступа к файлам;
- 3) организацию файлов в виде иерархии каталогов.

С точки зрения операционной системы, жесткий диск – это набор кластеров. **Кластер** – область диска определенного размера для хранения данных. Том или логический диск – область внешней памяти, с которой операционная система работает как с единым целым.

В ОС Windows 7 доступны следующие файловые системы: NTFS, FAT32 и редко используемая система FAT (FAT16).

В NTFS (файловая система новой технологии, основная файловая система ОС Windows) размер кластера равен 4Кб, размер тома не более 2ТБ. Структурой NTFS предусмотрено хранение для каждого файла и каждой папки специального блока безопасности, который содержит следующую информацию:

- идентификатор (имя) пользователя, создавшего файл;
- список контроля доступа, в котором перечислены разрешения доступа к файлу или папке для пользователей и групп;
- системный список контроля доступа, в котором перечислено, какие действия (например, чтение, запись и т.п.) для каких пользователей и групп необходимо фиксировать в журнале аудита.

NTFS, поддерживает защиту файлов и каталогов; сжатие файлов; поддержка многопоточных файлов; отслеживание связей; шифрование.

FAT (таблица размещения файлов) - линейная табличная структура со сведениями о файлах – именами файлов, их атрибутами и другими данными, определяющими местоположение файлов или их фрагментов в среде FAT, размер кластера 32Кб. Элемент FAT определяет фактическую область диска, в котором хранится начало физического файла. В файловой системе FAT логическое дисковое пространство любого логического диска состоит из двух областей:

- системная область – создается при форматировании диска и обновляется при манипулировании файловой структурой;
- область данных – содержит файлы и каталоги, подчиненные корневому каталогу, доступна через пользовательский интерфейс.

3.4. Задание на лабораторную работу(часть 2)


3.4.1. Определение типа файловой системы.

Открыть папку «Компьютер», выбрать диск С:, кликнуть его правой кнопкой мышки и выбрать в контекстном меню пункт «Свойства». В открывшемся окне на вкладке «Общие» сверху будет указан тип диска и используемая файловая система. Например, для Windows 7 будет указан тип файловой системы NTFS.

Определить тип файловой системы диска С:. Вставить флэш-накопитель и определить тип диска и используемую файловую систему для съемного диска

3.4.2. Отображение расширений файлов

Если дважды щелкнуть мышью на имени файла со значком, будет запущена программа *Word*, в которой, в свою очередь, откроется выбранный файл. Задача *Windows* состоит в том, чтобы обработать файл согласно его расширению. При этом имя файла может быть любым. Расширения файлов не отображаются по умолчанию в *Windows 7*. Чтобы это сделать, необходимо выполнить следующие действия.

1. Открыть раздел «Параметры папок». Для этого нажать кнопку *Пуск* , выбрать последовательно компоненты *Панель управления*, и *Параметры папок*.

2. Перейти на вкладку *Вид* и в разделе *Дополнительные параметры* выполнить одно из следующих действий.

- Для отображения расширения имен файлов, снять флажок *Скрывать расширения для зарегистрированных типов файлов* и нажать кнопку ОК.
- Для сокрытия расширения имен файлов, установить флажок *Скрывать расширения для зарегистрированных типов файлов* и нажать кнопку ОК.

Выполнить действия по отображению расширений файлов.

3.4.3. Отображение скрытых папок и файлов

Наряду с обычными файлами в *Windows* есть системные файлы. Эти файлы содержат программы и данные, отвечающие за работу операционной системы. Поэтому изменять, переименовывать, удалять их ни в коем случае нельзя. Располагаются системные файлы в

папке *Windows*, в которую проводится установка операционной системы, а также в некоторых других папках. Основные системные файлы по умолчанию скрыты для просмотра. Чтобы их увидеть, в окне *Параметры папок*, на вкладке *Вид* в разделе *Дополнительные параметры* нужно установить флажок *Показывать скрытые папки, файлы и диски*. Системные файлы защищены от перезаписи.

ВНИМАНИЕ! Удаление или изменение системных файлов может привести к тому, что ОС перестанет запускаться или будет работать с ошибками.

Отобразить скрытые файлы и папки

3.4.4. Свойства файлов и папок

Чтобы просмотреть свойства определенного файла, необходимо щелкнуть правой кнопкой мыши на значке файла и в появившемся контекстном меню выбрать команду *Свойства*. При этом на экране появится окно с несколькими вкладками, каждая из которых позволит указать для файла различные параметры. Количество вкладок и их содержимое зависит непосредственно от параметров самого файла. Окно *Свойства* открывается на вкладке *Общие* (Рисунок 3.1).

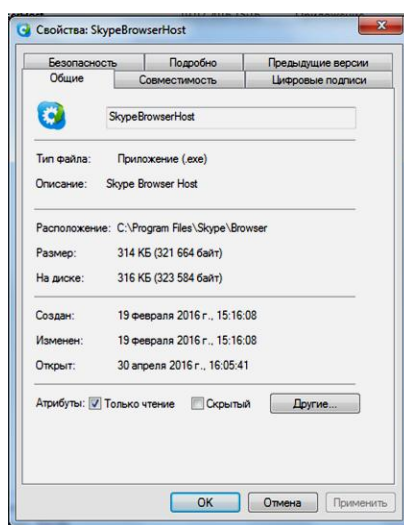


Рисунок 3.1. - Окно свойств файла, вкладка «Общие»

Вверху этой вкладки указывается имя файла, которое при желании можно изменить. В поле *Приложение* указывается программа, которая используется для открытия файла с данным расширением.

Если файл сам является программой, то для него этот раздел отсутствует. Далее на вкладке *Общие* последовательно указываются местоположение и размер файла, его размещение на диске, дата создания и последней модификации, а также дата последнего открытия.

Вкладка *Подробно* (Рисунок 3.2) окна *Свойства* содержит общие сведения о файле. То, какие сведения отображаются, зависит от типа файлов. Например, если это текстовый файл, то на вкладке *Подробно* будут отображена информация о количестве страниц, слов, знаков, имени автора и прочие сведения.

Внизу вкладки *Общие* с помощью флажков можно указать свойства (или, другими словами, атрибуты) файлов. Свойство *Только чтение* позволяет сделать файл доступным только для чтения, т.е. содержимое такого файла можно будет лишь просматривать или печатать. Удалить или изменить подобный файл нельзя. Если установить флажок *Скрытый*, то файл станет невидимым для обычного просмотра. Флажок *Архивный* будет присутствовать только в том случае, если используется файловая система FAT32. Если используется файловая система NTFS, то вместо флажка *Архивный* будет расположена кнопка *Другие* (атрибуты сжатия и шифрования). Щелкнув на этой кнопке, можно в новом

окне изменить различные параметры, связанные с файловой системой NTFS.

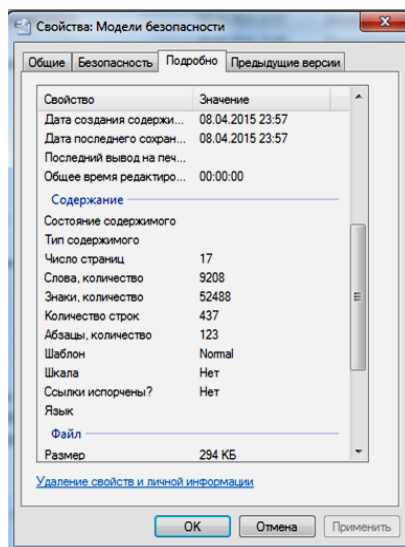


Рисунок 3.3.- Окно свойств файла, вкладка «Подробно»

Окно свойств папки вызывается из контекстного меню после щелчка правой кнопкой мыши по значку папки и содержит вкладки *Общие*, *Доступ*, *Безопасность*, *Настройка*.

Просмотреть и выписать свойства текстового файла Word, рабочей книги Excel, свойства папки, расположенные на разных вкладках окна «Свойства» .

3.4.5. Удаление временных файлов

Если система установлена на диск "C:", то адрес нахождения первой папки с временными файлами Windows - C:\Windows\Temp\ . Вторая папка Temp в Windows 7 находится по адресу: C:\Users\Имя пользователя\AppData\Local\AppData и имеет свойство скрытой папки. Увидеть её можно только включив отображение скрытых файлов и папок.

Самым простым способом избавиться от временных файлов является программа «Очистка диска», которая встроена в Windows. Чтобы запустить её, нужно нажать левой кнопкой мыши на диск C: в разделе «Компьютер», выбрать пункт «Свойства» и во вкладке «Общие» кликнуть по кнопке «Очистка диска». Поставить галочку «Временные файлы», затем нажать «ОК».

Удалить временные файлы из папок Temp можно вручную. Для того, чтобы найти вторую папку Temp, нужно отобразить скрытые файлы и папки. Находим в меню «Панель управления» и выбираем «Параметры папок». Во вкладке «Вид» в окне «Дополнительные параметры» выбираем «Показывать скрытые файлы, папки и диски» и жмем «ОК». Вторая папка находится по адресу C:\Users\Имя пользователя\AppData\Local\Temp.

Также можно зайти в «Пуск» — «Выполнить» и ввести %TEMP%, после чего нажать *Enter*. Выделить файлы в этой папке (Ctrl+A) и удалить. Причем может случиться так, что появится сообщение, что некоторые файлы нельзя удалить. Это нормально, потому что некоторые из них могут использоваться системой в данный момент. Нажать «ОК».

Удалить временные файлы с помощью программы *Очистка диска* и вручную.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Таненбаум Э. Современные операционные системы. / Э. Таненбаум. –СПб.: Питер, 2011. –1120 с.
2. Stallings W. Operatings Systems. Internals and Design Principles/ W. Stallings.–Sevents Edition. – 2012.– 768 с.
3. Дейтел Х.М. Операционные системы. Том 1. Основы и принципы/ Х.М. Дейтел, П.Дж.Дейтел, Д.Р.Чофнес .–М.: Изд. Бином – 2011. – 1024 с.
4. Информатика: базовый курс: учеб. пособие для студ. вузов/ Ред. С. В. Симонович. - СПб. и др. : Питер, 2008. - 640 с.
5. Лабораторный практикум по информатике и компьютерным технологиям: учеб. пособие для студ. вузов / Харьк. гос. экон. ун-т ; Ред. А. И. Пушкарь. - Харьков : ИНЖЭК, 2004. - 466 с.
6. Мезенцева, Е.М. Операционные системы. Лабораторный практикум / Е.М. Мезенцева, О.С. Коняева, С.В. Малахов.– Самара: ПГУТИ, 2017. – 216 с.
7. Методические указания к выполнению лабораторных работ по дисциплине «Безопасность операционных систем» для магистров ОФО по направлению 38.04.08 «Финансы и кредит», профиль подготовки «Финансовый мониторинг»/ Сост. ст. преподаватель кафедры ИТиКС М.А. Лебедева. – Севастополь: Изд-во СГУ, 2016.– 36с.
8. <https://okeygeek.ru/pyat-sposobov-otkryt-komandnuyu-stroku-v-windows-10/>