

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 12. «ПРОГРАММИРОВАНИЕ КОМАНДНЫХ BAT- ФАЙЛОВ В WINDOWS»

### Теоретическая часть

#### Командный интерпретатор `command.com`

В операционных системах MS-DOS и Windows команды выполняются с помощью специальной программы – интерпретатора команд `command.com`. Некоторые команды распознаются и выполняются самим командным процессором – они называются *внутренними*, другие представляют собой отдельные программы – их называют *внешними (утилитами)*.

Внутренними командами являются:

<b>Команда</b>	<b>Описание</b>
BREAK	прерывание
CALL	вызов командного файла
CHCP	вывод или смена текущей кодовой страницы
CHDIR	смена каталога
CLS	очистка экрана
COPY	копирование файлов
CTTY	смена устройства вывода – экран, принтер и т.д.
DATE	просмотр и установка даты
DEL	удаление файла
DIR	вывод информации о содержимом дисков и каталогов
ECHO	вывод строки на экран
EXIT	выхода из интерпретатора <code>command.com</code>
FOR	оператор цикла в bat-файлах
GOTO	оператор перехода в bat-файлах
IF	условный оператор в bat-файлах
LOADHIGH	загрузка программы в область верхней памяти
MKDIR	создание каталога
PATH	указание списка путей для автоматического поиска

## Операционные системы

PAUSE	пауза до нажатия любой клавиши
PROMPT	установка формата командной строки
REM	комментарий в командных файлах
RENAME	переименование файла
RMDIR	удаления каталога
SET	установка переменных окружения
SHIFT	сдвиг параметров для пакетного файла
TIME	просмотр и установка времени
TYPE	просмотр файла
VER	вывод версии
VERIFY	контроль записи
VOL	информация о томе

Все остальные команды являются внешними и хранятся в виде исполняемых файлов в папке *C:\Windows\Command*.

Для работы с командной строкой в Windows можно выполнить *Пуск/Программы/Сеанс Ms-Dos*, запустить файл *command.com* непосредственно или использовать файловые менеджеры FAR, Norton Commander. При запуске *command.com* можно использовать ключи (кстати, их мы уже упоминали в параметре SHELL файла *config.sys*). Кроме указанных там параметров /P (загрузка в память) и /E (размер области переменных окружения) интересны параметры /Y (выполнение пакетного файла по шагам – для отладки), /C (исполнение указанной далее программы и возврат), /K (исполнение команды с продолжением). Используя ключ /C, например, можно создавать ярлыки для выполнения внутренних команд.

Отметим сначала несколько особенностей опеределения путей к файлам Windows. Файловая система имеет древовидную структуру и имена файлов задаются в формате *[диск:] [путь] [имя\_файла]*. Если путь начинается с символа «\», то маршрут вычисляется от корневого каталога, иначе от текущего.

Существуют особые обозначения для текущего каталога («.») и трех его верхних уровней («..» — родительский, «...» — второго уровня и «....» — третьего уровня). Например, для текущего каталога *C:\Windows\Media\Office97* путь к файлу *autoexec.bat* в корневом каталоге диска C: может быть записан в виде

....\autoexec.bat.

В именах файлов можно применять шаблоны \* (произвольное кол-во любых символов) или ? (один произвольный символ или его отсутствие). Для того чтобы использовать длинные имена файлов при работе с командной строкой, их нужно заключать в двойные кавычки.

В синтаксисе командной строки особое место занимает перенаправление ввода-вывода с помощью символа «>» на стандартное устройство (PRN, COM1 и т. п) или файл. Например, **DIR /? > helpdir.txt** выведет справку по команде DIR в файл. Символ «>>» позволяет не создавать файл заново, а дописать в него. По аналогии символ «<<» позволяет читать данные не с клавиатуры, а с файла. Например, **DATE < date.txt** вводит новую дату из файла.

Кроме того, командная строка поддерживает конвейеризацию с помощью конструкции: команда1 | команда2, когда сообщения выводимые первой командой используются в качестве входных данных для второй. Например, команда MORE выводит информацию частями, не превышающим размер экрана. Строка **TYPE имя\_ файла | MORE** выводит файл по экранам с паузой.

### Переменные окружения

При загрузке ОС Windows в оперативной памяти постоянно хранится набор т.н. *переменных окружения* (environment variables). Хотя в Windows есть более совершенный способ для хранения системных значений – реестр, многие программы по-прежнему используют переменные окружения.

Наиболее важные переменные хранят системный путь для поиска (PATH), каталог запуска Windows (WINDIR), место хранения временных файлов (TEMP).

Переменные устанавливаются с помощью команды

**SET** [переменная[строка]]

Запуск SET без параметров приводит к выводу списка переменных среды. Для получения их значений (всегда строки) нужно имя соответствующей переменной заключить в символы «%», например: %TEMP%.

### Основные команды

Остановимся подробнее на командах работы с файловой системой: ATTRIB, COPY, XCOPY, DIR, MKDIR, RMDIR, DEL,

DELTREE, REN, MOVE, SUBST, VOL, LABEL.

**ATTRIB** [+R | -R] [+A | -A] [+S | -S] [+H | -H]  
 [[диск:][путь]имя\_файла] [/S] - просмотра или установка атрибутов файлов.

В Windows 9x можно использовать 4 атрибута:

- Read-Only (R) – только для чтения;
- System (S) – системный;
- Archive (A) – архивный;
- Hidden (H) – скрытый.

Установка атрибута производится ключом «+», снятие «-». Ключ «/S» применяется для обработки файлов во всех подкаталогах указанного пути. Например, сделать все Word-файлы в каталоге «Мои документы» доступными только для чтения: **ATTRIB +R "C:\Мои документы \\*.doc"**. Атрибуты файлов можно просмотреть/поменять в файловых менеджерах или проводнике (пункт «Свойства» в контекстном меню по правой клавише).

**COPY** [/A | /B] источник [/A | /B] [+источник [/A | /B] [+ ...]]  
 [результат [/A | /B]] [/V] [/Y | /-Y] – копирование файлов.

Описание ключей сведено в таблицу:

Параметр	Описание
/A	файл является текстовым файлом ASCII (символ + в конце)
/B	файл является двоичным, результат - каталог для результата копирования и/или имя создаваемого файла
/V	проверка правильности копирования путем сравнения источника и приемника
/Y	отключение режима запроса подтверждения на замену файлов
/-Y	включение режима запроса подтверждения на замену файлов

В качестве источника или результата при копировании можно указывать имена не только файлов, но и устройств компьютера:

- PRN – принтер,
- LPT1-LPT3 – соответствующие параллельные порты;
- AUX – устройство, присоединяемое к последовательному

порту 1;

- COM1-COM3 — соответствующие последовательные порты;
- CON – терминал (при вводе – это клавиатура, а при выводе – монитор);
- NUL – пустое устройство (все операции игнорируются).

Например, для печати файла на принтере можно ввести команду **COPY abc.txt PRN**, для создания файла **COPY CON my.txt** (для конца файла ввести +).

Команда COPY может объединять (склеивать) несколько файлов путем использования групповых знаков «\*» и «?» (**COPY /B \*.dat all.dt**) или знака «+» (**COPY /B 1.txt+2.txt 3.txt**). Здесь ключ /B используется для предотвращения усечения соединяемых файлов, т.к. по умолчанию файлы считаются текстовыми.

Среди недостатков команды COPY надо отметить невозможность копирования скрытых системных файлов, замены файлов с атрибутом «Read Only», а также копирования открытых (занятых) файлов (в этом случае процесс просто прерывается).

Некоторые из этих проблем можно решить с помощью утилиты XCOPY.

**XCOPY** источник результат [/A | /M] [/D[:дата]] [/P] [/S] [/E] [/W] [/C] [/I] [/Q] [/F] [/L] [/H] [/R] [/T] [/U] [/K] [/Y | /-Y]

Команда XCOPY работает только с файлами и каталогами (а не с устройствами).

Используемые ключи рассмотрим в табличном виде:

Ключ	Описание
/A	Копирование только файлов с установленным архивным атрибутом. Сам атрибут при этом не изменяется.
/M	Копирование только файлов с установленным архивным атрибутом. После копирования атрибут снимается.
/D	Копирует только файлы, измененные не ранее указанной даты. Если дата опущена – то если источник новее результата.
/S	Копирование каталогов с подкаталогами (кроме пустых).

## Операционные системы

/E	Копирование каталогов с подкаталогами, в том числе пустых
/W	Режим запроса на нажатие любой клавиши до начала копирования
/C	Продолжение в случае возникновения ошибок
/I	Копирование нескольких файлов, когда файл назначения отсутствует. В этом случае считает, что файл назначения д.б. каталогом и создает его без дополнительных запросов
/V	Сравнение конечных файлов с исходными
/P	Вывод запросов перед созданием каждого нового файла
/Q, /F, /L	Запрет вывода имен копируемых файлов, /L – вывод имен, /F – полных
/G (2000)	Копирование зашифрованных файлов в каталог результата, не поддерживающий шифрование
/H	Копирование скрытых и системных файлов (среди прочих)
/R	Разрешение замены файлов, предназначенных только для чтения
/T	Создание структуры каталогов (кроме пустых каталогов) без копирования файлов. Если требуется создать пустые каталоги и подкаталоги — /T /E.
/U	Копирование только файлов, уже имеющих в конечном каталоге
/K	Копирование атрибута «Только чтение» (обычно он сбрасывается)
/N	Использование коротких имен при копировании

/O (NT)	Копирование сведений о владельце и данных ACL
/X (NT)	Копирование параметров аудита файлов (требует /O)
/Y /-Y	Подавление (обязательный) запроса подтверждения на перезапись существующего конечного файла
/Z (NT)	Копирование сетевых файлов с возобновлением
/EXCLUDE:файл1[+файл2]... (NT)	Исключение определенных файлов из операции копирования

При использовании команды XCOPY внутри пакетных файлов по переменной **ERRORLEVEL** можно анализировать код выхода (завершения) команды (0 – без ошибок, 1 – файлы не найдены, 2 – прерывание по +, 4 – не хватает места, ошибка в назначении или синтаксисе, 5 – ошибка записи на диск).

**DIR** [диск:][путь][имя\_ файла] [/A[:]атрибуты]] [/B] [/C] [/D] [/L] [/N] [/O[:]порядок] [/P] [/Q] [/S] [/T[:]время]] [/W] [/X] [/4] - вывод информации о содержимом дисков и каталогов.

При использовании без ключей выводит метку диска, имена (в коротком и длинном вариантах) файлов и подкаталогов, находящихся в текущем подкаталоге, а также дату и время их последней модификации. Также выводятся общее число файлов в каталоге, их объем и размер свободного пространства.

Ключи команды DIR позволяют задать различные режимы расположения, фильтрации и сортировки выводимой информации:

Ключ	Описание
/A	Вывод файлов с указанными атрибутами (D-каталоги, A, R, S, H – как обычно. Префикс «-» будет иметь значение НЕ.
/B	Вывод только имен файлов (переопределяет /W)
/C	Применение разделителя групп разрядов для вывода размеров файлов (по умолчанию). Для отключения /-C.

## Операционные системы

/D (NT)	Вывод списка в несколько столбцов с сортировкой по столбцам
/L	Использование нижнего регистра для имен файлов
/N	Отображение имен файлов в крайнем правом столбце
/O	Сортировка списка отображаемых файлов (префикс «-» обращает порядок): N — по имени (алфавитная) S — по размеру (сперва меньшие) E — по расширению (алфавитная) D — по дате (сперва более старые) G — начать список с каталогов A — по дате загрузки (начиная с более старых)
/P	Пауза после заполнения каждого экрана
/Q (NT)	Вывод сведений о владельце файла
/V (9x)	Вывод расширенных сведений о файлах и каталогах
/S	Вывод списка файлов из указанного каталога и его подкаталогов
/T	Выбор поля времени для отображения и сортировки C — создание A — последнее использование W — последнее изменение
/W	Вывод списка в несколько столбцов
/X	Отображение коротких имен для файлов, чьи имена не соответствуют стандарту
/4 (9x)	Вывод номера года в четырехзначном формате

Стандартный набор ключей можно записать в переменную среды DIRCMD. Для отмены их действия введите в команде те же ключи с префиксом «-», например: /-W. Команда DIR (так же как и другие) поддерживает перенаправление ввода-вывода (символы



«>>» и «>>>»), например, команда

**DIR "C:\Мои документы" /W/O:N > PRN**

выполняет печать в широком формате в алфавитном порядке на принтер.

**MKDIR (MD)** [диск:]путь и **RMDIR (RD)** [диск:]путь - создание и удаление каталога. Команда MKDIR не будет выполнена, если каталог или файл с заданным именем уже существуют, а RMDIR – если удаляемый каталог не пустой. В Windows NT появилась возможность удалять подкаталоги (**RD /S**), и создавать полный путь – **MD \A\B\C\D**.

**DEL** [диск:][путь]имя\_ файла [/P] и **ERASE** [диск:][путь]имя\_ файла [/P] - удаление файлов. Ключ /P означает выдачу запроса на удаление (при задании удаления всех файлов **DEL \*.\*** или **DEL .** запрос будет всегда). В Windows NT добавлены параметры для удаления файлов с атрибутом «Только для чтения» (/F), подкаталогов (/S), отмены подтверждений (/Q), а также удаления файлов, не имеющих заданных атрибутов (/A:атрибуты).

**DELTREE** [/Y] [диск:]путь - удаление каталога вместе со всем содержимым. Ключ /Y используется для отключения запроса на подтверждение.

**RENAME (REN)** [диск:][путь][каталог1 |файл1] [каталог2 | файл2] - переименование файла или каталога. Так же можно использовать групповые символы «\*» и «?», например: **REN \*.txt \*.doc**. В этой команде нельзя указать другой диск или каталог – для этих целей надо использовать команду MOVE.

**MOVE** [/Y | /-Y] [диск:][путь]имя\_файла1[,...] рез\_ файл - перемещение файлов или каталогов.

**SUBST** [диск1: [диск2:]путь] - сопоставление заданному пути имя виртуального диска. Часто нужно для инсталляции программ (эмуляции корневого каталога CD). Например: **SUBST F: C:\INSTALL\DELPHI5**. Ключ /D используется для удаления ранее созданного виртуального диска: **SUBST F: /D**. SUBST без параметров выводит текущий список виртуальных дисков (нельзя назначать их на сетевые каталоги).

**VOL** [диск:] и **LABEL** [диск:][метка] – вывод и задание метки тома.

### Командные bat-файлы

Командный (пакетный) файл в Windows – это обычный текстовый файл с расширением *bat*, в котором записаны допустимые команды ОС, а также некоторые инструкции (ключевые слова) для алгоритмизации действий.

Например, *deltmp.bat* удалит все временные файлы в каталоге Windows\Temp:

```
C: |
CD %TEMP%
ATTRIB -R *.tmp
DEL *.TMP
```

Основной командой для вывода информации в пакетных файлах служит **ECHO [сообщение]**.

По умолчанию команды пакетного файла перед исполнением выводятся на экран. С помощью команды **ECHO OFF** можно отключить дублирование вывода команд на экран, **ECHO ON** восстанавливает режим дублирования. Кроме этого, можно отключить дублирование отдельной строки, если предварить ее символом «@». И, наконец, может вывести пустую строку комбинацией **ECHO.** (с точкой в конце).

```
@ECHO OFF
ECHO Привет !
ECHO.
ECHO Пока ...
```

При запуске командных файлов в командной строке можно указывать произвольное число параметров, которые затем можно использовать внутри пакетного файла.

Для доступа к параметрам применяются символы %0 – имя файла, %1-%9 – значения первых девяти параметров соответственно. Например, имеется командный файл *copier.bat* следующего содержания:

```
@ECHO oFF
CLS
ECHO Файл %0 копирует каталог %1 в %2
XCOPY %1 %2 /S
```

При запуске его из командной строки с 2 параметрами:

```
copier.bat C:\Programs D:\Backup
```

на экран выводится сообщение «Файл *copier.bat* копирует каталог C:\Programs в D:\Backup» и происходит соответствующее

копирование.

При необходимости можно использовать более девяти параметров командной строки с помощью команды **SHIFT**, которая изменяет значения замещаемых параметров с %0 по %9, копируя каждый параметр в предыдущий. Значение %1 – в %0, %2 в %1 и т.д., 10 параметр в %9. Команда, обратная SHIFT, отсутствует, поэтому восстановить параметры уже не удастся.

В командных файлах можно использовать переменные окружения и объявлять собственные с помощью команды SET. Все переменные рассматриваются как строки и в Windows 9x над ними нельзя производить арифметические действия (в отличие от NT), а только конкатенацию (просто слитно 2 переменные без знака «+»). Например, в результате выполнения командного файла:

```
SET A=Первый
```

```
SET B=Второй
```

```
SET C=%A%%B%
```

```
ECHO Переменная C=%C%
```

на экран выведется «Переменная C= Первый Второй».

Для управления выполнением командных файлов существуют команды приостановки, перехода, условного ветвления, циклов и вызова внешних командных файлов.

Для того, чтобы прервать выполнение командного файла, надо нажать + или + (в Windows NT есть команда выхода EXIT /B). Для приостановки с выдачей запроса на нажатие любой клавиши есть команда PAUSE, которую рекомендуется использовать перед выполнением потенциально опасных действий. Например:

```
ECHO Сейчас будут удалены все файлы в каталоге C:|Мои документы
```

```
ECHO Для отмены нажмите Ctrl-C
```

```
PAUSE
```

```
DEL "C:|Мои документы|*.*"
```

Для организации циклов используется конструкция: **FOR** %%переменная **IN** (множество) **DO** команда [параметры]

Отличие от классических языков программирования в том, что не происходит регулярного приращения счетчика, а вместо этого используется заданный список значений в параметре *множество*. Скобки здесь обязательны, внутри которых одно или несколько строковых значений, разделенных запятыми. Например:

```
@ECHO OFF
```

```
FOR %%i IN (Раз, Два, Три) DO ECHO %%i
```

напечатает следующее:

```
Раз
```

*Два*

*Три*

В качестве переменных цикла можно использовать лишь имена, состоящие из одной буквы. Весь цикл должен быть записан в одной строке, вложенные циклы не допускаются.

В параметре *множество* можно представить одну или несколько групп файлов. Например, для вывода всех файлов с расширениями \*.doc и \*.txt:

```
FOR %%f IN(C:\TEXT\*.doc C:\TEXT\*.prn) DO ECHO %%f
>> list.txt
```

Из одного командного файла можно вызвать другой, просто указав его имя. Однако в этом случае управление назад не вернется, для этого существует команда **CALL**, с помощью которой можно в командных файлах организовать подобие подпрограмм. Например, командный файл proc.bat:

```
@ECHO OFF
```

```
ECHO Записываем файл %1.txt
```

```
ECHO Параметр вызова: %1 > %1.txt
```

можно вызвать в другом пакетной файле:

```
FOR %%i IN (Первый, Второй, Третий) DO CALL proc.bat
%%i
```

в результате proc.bat вызывается 3 раза и создает 3 файла – Первый.txt, Второй.txt и Третий.txt с соответствующим текстом.

Командный файл может содержать метки, начинающиеся с двоеточия («:») и команды **GOTO** - перехода к этим меткам. Имя метки задается набором символов, следующих за двоеточием до первого пробела или конца строки. Например:

```
@ECHO OFF
```

```
GOTO Label1
```

```
ECHO Эта строка никогда не выполнится
```

```
:Label1
```

```
REM Продолжение выполнения
```

```
DIR
```

С помощью команды **IF** в пакетных файлах можно выполнять обработку условий 3 типов:

1) **IF [NOT]** строка1==строка2 команда

Условие считается истинным (обратите внимание на 2 знака равно) при точном совпадении обеих строк (регистр имеет значение). Строки могут быть литеральными (кавычки для них не требуются) или представлять значения переменных. Например:

```
IF %1%==Петя ECHO Привет, Петя !
```

Для предотвращения синтаксических ошибок в случае от-

сутствия переменных или параметров, рекомендуется при сравнении строк приписывать им вначале какой-нибудь символ, например:

```
IF -%MyVar%==-C:| ECHO OK !
```

2) **IF [NOT] EXIST** файл команда

Проверка существования заданного файла (кавычки для имени также не требуются, кроме длинных). Например:

```
IF NOT EXIST C:\autoexec.bat ECHO у вас нет файла автозагрузки !
```

```
IF EXIST "C:\Мои документы\Работа.doc" ECHO Все в порядке !
```

3) **IF [NOT] ERRORLEVEL** число команда

Условие считается истинным, если последняя запущенная программа или команда завершилась с кодом возврата, равным либо превышающим указанное число. Например:

```
xcopy my.txt c:| > nul
```

```
IF errorlevel 1 goto ErrOccurred
```

```
echo Копирование прошло без проблем.
```

```
goto EndBatch
```

```
:ErrOccurred
```

```
echo При выполнении команды возникла ошибка !
```

```
:EndBatch
```

К сожалению, в командных файлах нет возможности организовать полноценный диалог с пользователем путем ввода строк с клавиатуры. Единственное средство – команда CHOICE, которая выводит подсказку и ждет выбора пользователем варианта из указанного набора клавиш.

**CHOICE** [/C:]варианты [/N] [/S] [/T:]с,nn] [текст]

Если в команде не задается текст, то пользователь видит на экране только подсказку. Рассмотрим подробно ключи этой команды:

Ключ	Описание
/C	задает варианты ответа, по умолчанию YN
CHOICE /C:ync	Yes, No, Cancel [Y, N, C]
/N	выводит только текст без вариантов ответа (клавиши для ответа работают)
CHOICE /N	

/S	учет регистра символов (по умолчанию верхний и нижний воспринимаются одинаково)
/T	c – символ по умолчанию, который вводится после указанной в <i>nn</i> секунд паузы

После выполнения команды CHOICE переменная ERRORLEVEL приобретает значение, равное номеру варианта ответа, что позволяет использовать команду IF для организации ветвления. Если происходит ошибка, CHOICE возвращает 255, прерывание по + или < CTRL >+ — 0. В качестве примера рассмотрим простейшее меню:

```
@ECHO OFF
ECHO Выберите режим
ECHO 1 — Простой
ECHO 2 — Расширенный
ECHO.
CHOICE /c:12 Введите пункт меню
IF ERRORLEVEL 2 goto Choice2
IF ERRORLEVEL 1 goto Choice 1
ECHO Выход из меню
GOTO Done
:Choice1
ECHO Выбран пункт 1
GOTO Done
:Choice2
ECHO Выбран пункт 2
GOTO Done
:Done
```

### Особенности командных файлов в Windows NT-XP

Фактически в составе Windows NT имеются два командных интерпретатора – *command.com* во встроенной виртуальной машине MS-Dos (аналогичный по возможностям рассмотренному ранее) и специальный интерпретатор команд Windows NT – *cmd.exe (%System Root%\System32)*.

Для запуска командного интерпретатора можно вызвать ярлык «Пуск/Программы/Командная строка», файловый менеджер типа FAR или загрузить *cmd.exe* непосредственно. При запуске *cmd.exe* можно использовать, как и в *command.com* ключи /C, /K, а также задание цвета и кодировки. Ключи /X и /Y соответ-

ственно включают и выключают расширенный режим работы интерпретатора (по умолчанию установлен).

Работа с переменными среды текущего командного окна осуществляется, как и в Windows 9x, с помощью команды SET. Естественно, изменения, которые вносятся в переменные среды этой команды, актуальны только в текущем командном окне.

Новые, весьма полезные, возможности у команды SET появляются при включении расширенной обработки команд. Теперь переменные могут рассматриваться как числа и с ними можно производить арифметические вычисления. Для этой цели имеется дополнительный ключ /A:

**SET /A** переменная= выражение

Использование ключа /A указывает, что стоящая справа от знака равенства строка является числовым выражением, значение которого вычисляется.

Например, если задать команду

*SET /A M=1+2*

то значение переменной M будет равно трем.

Обработчик выражений, входящих в команду SET, очень прост и поддерживает следующие операции, перечисленные в порядке убывания приоритета:

- группировка с помощью круглых скобок ( );
- арифметические операторы умножения (\*), целочисленного деления (/), остатка от деления (%);
- арифметические операторы сложения (+) и вычитания (-);
- двоичный сдвиг влево (<<) и вправо (>>);
- двоичное И (&);
- двоичное исключающее ИЛИ (^);
- двоичное ИЛИ (|);
- операторы присваивания =\*, =/, =%, =+, =-, =, &=, ^=, |=, <<= и >>=;
- разделение операторов с помощью запятой (,).

При использовании любых логических или двоичных операторов необходимо заключить строку выражения в кавычки. Можно использовать префиксы для систем счисления – 0x -16, 0b — 2, 0 — 8. Любые нечисловые строки в выражении рассматриваются как имена переменных среды, значения которых преобразуются в числовой вид перед использованием. Если переменная с указанным именем не определена, вместо нее подставляется нулевое значение. Например, если переменная X не была предварительно задана, то в результате выполнения команды

*SET /A N=X+5*

значение N будет равно пяти.

Таким образом, применение ключа /A позволяет выполнять арифметические операции со значениями переменных среды, причем не нужно вводить знаки % для получения их значений. Кроме того, усовершенствована работа с переменными среды как со строками (кроме конкатенации – замена вхождений и выделение подстрок).

Командные файлы в Windows NT, как и в Windows 9x, являются обычными текстовыми файлами, однако для них в операционной системе зарезервированы не одно, а два расширения: bat и cmd. Новые возможности командных файлов связаны с командами SETLOCAL, ENDLOCAL, PUSHD, POPD, а также изменениями в GOTO, CALL, IF, FOR.

В Windows NT имеется возможность локализовать изменения переменных среды внутри пакетного файла, т. е. автоматически восстанавливать значения всех переменных в том виде, в каком они были до начала запуска данного файла. Команда SETLOCAL определяет начало области локальных установок переменных среды, т.е. изменения среды, внесенные после выполнения SETLOCAL, будут являться локальными относительно текущего пакетного файла. Каждая команда SETLOCAL должна иметь соответствующую команду ENDLOCAL для восстановления прежних значений переменных среды.

При работе с параметрами командного файла можно обозначить все аргументы через %\*, а также использовать синтаксический анализ:

Оператор	Описание	Пример
%~Fn	Полное имя файла	%~F1=C:\TEXT\Рассказ.doc
%~Dn	Имя диска	%~D1=C:
%~Pn	Путь к файлу	%~P1=\TEXT\
%~Nn	Имя файла	%~N1=Рассказ
%~Xn	Расширение файла	%~X1=doc

Можно задавать в качестве метки перехода строку :EOF, которая передает управление в конец текущего пакетного файла. Это позволяет легко выйти из пакетного файла без определения каких-либо меток в самом его конце (**GOTO** :EOF).

В качестве адресата команды **CALL** можно использовать метки внутри текущего командного файла (аналог подпрограмм). Три вызове такой команды создается новый контекст текущего пакетного файла с заданными аргументами и управление переда-



ется на инструкцию, расположенную сразу после метки. Для выхода из такого пакетного файла необходимо два раза достичь его конца (первый выход возвращает управление на инструкцию, расположенную сразу после CALL, а второй выход завершает выполнение пакетного файла).

Дополнительно вводятся еще три варианта команды **IF**:

1) **IF** [/I] строка1 оператор\_сравнения строка2 команда

В качестве операторов\_сравнения используются: EQ – равно, NEQ – не равно, LSS – меньше, GTR – больше, LEQ – меньше или равно, GEQ – больше или равно. Ключ /I задает сравнение без учета регистра.

2) **IF** CMDEXVERSION число команда

Служит для определения внутреннего номера версии расширенной обработки команд (какая на 2000 и XP — ?)

3) **IF** DEFINED переменная команда

Возвращает истинное значение, если переменная среды определена.

В Windows NT доступны еще пять разновидностей циклов

**FOR**, которые обеспечивают следующие функции:

1) **FOR /D** %переменная IN (набор) DO команда [параметры]

Выполнение заданной команды для всех подходящих имен каталогов, например получение списка всех каталогов на диске C:

```
FOR /D %%f IN (C:) DO ECHO %%f
```

2) **FOR /R** [[диск:]путь] %переменная IN (набор) DO команда [параметры]

Организует рекурсивное выполнение заданной команды для определенного каталога, а также всех его подкаталогов. Например, для распечатки всех файлов с расширением \*.txt:

```
FOR /R %%f IN (*.txt) DO PRINT %%f
```

3) **FOR /L** %переменная IN (начало, шаг, конец) DO команда [параметры]

Классический цикл с параметром (заданными началом, концом и шагом приращения). Например:

```
FOR /L %%f IN (1,1,5) DO ECHO %%f
```

4) **FOR /F** ["ключи"] %переменная IN (набор) DO команда [параметры]

Чтение и обработка строк из набора текстовых файлов. Ключи позволяют выделять слова и подстроки (токены) по разделителям (DELIMS) или начальным символам.

5) **FOR /F** ["ключи"] %переменная IN (^строка) DO команда [параметры]

Позволяет обработать строку вывода определенной команды (вместо набора файлов в предыдущем варианте – строка вызов команд в апострофах). Например, для вывода имен всех переменных среды:

```
FOR /F "DELIMS==" %%i IN ('SET') DO ECHO %%i
```

Кроме того, для переменных команды FOR разрешены также синтаксические операции (выделение имени, пути, диска — аналогично рассм. выше переменным окружения).

Команда **PUSHD** сохраняет имя текущего каталога для команды **POPD** и осуществляет переход в другой каталог. Ее синтаксис имеет вид:

```
PUSHD [путь | ..]
```

Вновь сделать текущим каталог, сохраненный командой **PUSHD**, можно с помощью команды **POPD**.

### Примеры выполнения заданий

1. Пусть имеется текстовый файл `protokol.txt`, в котором хранится журнал обработанных файлов в следующем формате:

```
<имя файла> <дата> <время>
```

Слово дата здесь начинается в каждой строке с двадцатой позиции.

Необходимо написать командный файл, с помощью которого сделать выборку из этого файла (т. е. создать новый текстовый файл с нужной информацией) за заданный месяц (мм) и год (гггг) в файл `ммгггг-.txt`, сформированный файл упорядочить по дате обработки. Нужные месяц и год указать как параметры командной строки.

Решение:

Практически все нужные действия выполняются с помощью конвейеризации команд **FIND** и **SORT**:

```
@ECHO OFF
```

**REM Проверка наличия параметров командной строки**

```
IF %1==- GOTO NoParam
```

```
IF %2==- GOTO NoParam
```

```
REM Выделение нужных строк из файла protokol.txt
```

```
FIND «%1.%2» protokol.txt | SORT /+26 > %1%2.txt
```

```
GOTO End
```

```
:NoParam
```

*ECHO Не заданы необходимые параметры командной строки!*

```
PAUSE
```

*:End*

2. Написать командный файл, который будет копировать из текущего каталога все файлы с расширением txt, кроме одного файла, указанного в качестве второго параметра командной строки, в каталог, указанный первым параметром. Если имя каталога, в который должно производиться копирование, не задано, то вывести сообщение об этом и прервать выполнение файла.

Решение:

Для выполнения поставленной задачи можно перебрать в цикле все файлы с расширением txt, проверяя перед копированием имя каждого из этих файлов:

```
@ECHO OFF
REM Проверка наличия параметра командной строки
IF -%!=-- GOTO NoDir
REM Копирование нужных файлов
FOR %%f IN (*.txt) DO IF NOT -%%f==-%2 COPY %%f %1
GOTO End
:NoDir
ECHO Не указан каталог для копирования!
PAUSE
:End
```

3. Задание аналогично второму упражнению. Дополнительные требования:

- а) переписывать только те файлы, которые новее одноименных в каталоге — приемнике,
- б) не прерывать копирование в случае возникновения ошибки,
- в) записывать в файл logcopy.txt имя каждого копируемого файла и результат выполнения операции для него.

Другими словами, файл logcopy.txt должен быть примерно таким:

```
Успешно: a.txt
Ошибка : b.txt
Успешно: c.txt
```

Решение:

В этом упражнении нужно создать два пакетных файла. В основном файле работает такой же цикл FOR, как и в предыдущем упражнении, однако вместо непосредственного выполнения команды COPY здесь вызывается командный файл 3\_1.bat:

```
@ECHO OFF
```

```

REM Проверка наличия параметра командной строки
IF -%1==- GOTO NoDir
REM Вызов в цикле файла 3_1.bat для копирования нужного
файла
FOR %%f IN (*.txt) DO IF NOT -%%f=-%2 CALL 3_1.bat %%f
%1
GOTO End
:NoDir
ECHO Не указан каталог для копирования!
PAUSE
:End
Вызываемый в цикле файл 3_1.bat имеет следующее со-
держимое:
@ECHO OFF
REM Копирование файла
XCOPY %1 %2 /D /C > NUL
REM Проверка успешности копирования
IF ERRORLEVEL 0 GOTO Success
REM Запись в файл отчета информации об ошибке при ко-
пировании
ECHO Ошибка: % 1 >> logcopy.log
GOTO End
:Success
REM Запись в файл отчета информации об успешном копи-
ровании
ECHO Успешно: %1 >> logcopy.log
:End
    
```

4. Создать командный файл, который выводил бы содержи-  
мое каталога, указанного в качестве параметра командной стро-  
ки, причем пользователю должна быть предоставлена возмож-  
ность выбора с помощью меню устройства для вывода: на экран  
(информация выводится по одному экрану), в текстовый файл  
c:\catalog.txt или на принтер.

Решение:

Следующий простой пакетный файл не требует дополни-  
тельных пояснений:

```

@ECHO OFF
CLS
IF -%1==- GOTO NoDir
REM Вывод меню на экран
ECHO A — На экран
    
```

```
ECHO Б — В файл C:\catalog.txt
ECHO В — На принтер
KEM Вывод подсказки для ввода
CHOICE /C:АБВ Куда выводить содержимое %1
CLS
KEM Определение сделанного выбора
IF ERRORLEVEL 3 GOTO DirToPrn
IF ERRORLEVEL 2 GOTO DirToFile
IF ERRORLEVEL 1 GOTO DirToCon
ECHO Выбор не был сделан.
GOTO End
:DirToCon
DIR %1 | MORE
GOTO End
:DirToFile
DIR %1 > C:\catalog.txt
GOTO End
:DirToPrn
DIR %1 > prn
GOTO End
:NoDir
ECHO Не указан каталог для сканирования!
PAUSE
:End
```

### Задания

В соответствии с номером студента по списку в журнале преподавателя разработать пакетный bat-файл. В пакетных файлах предусмотреть сообщение имени, назначения, применения и автора пакетного файла (при пустой командной строке и по ключу /?), контроль верности командной строки, наличие требуемых файлов и сохранность имени пакетного файла. Текущий каталог не изменять, если это специально не оговорено. Там, где необходимо, имена файлов указывать с полным путем и диском. С клавиатуры при работе пакетного файла вводить только числа, строковые данные выбирать либо из меню, либо передавать в командной строке.

### Задания для самостоятельного выполнения.

1. Разработать пакетный файл для обновления архива. Выбор архиватора осуществляется из меню. Имя архива

передается в командной строке.

2. Разработать пакетный файл для очистки подкаталога с подтверждением.

3. Разработать пакетный файл для проверки дисков (каталогов) на вирусы. Диск выбирается из меню. Имя антивирусной программы — в командной строке (drwebw.exe).

4. Разработать пакетный файл для выбора из меню на запуск одного из нескольких редакторов (notepad, ncedit и др.) для редактирования требуемого файла. Имя файла передается в командной строке.

5. Разработать пакетный файл для архивации файлов в каталогах и подкаталогах (по отдельности в каждом каталоге) и удаления архивных файлов. Имена каталога и архиватора — в командной строке.

6. Разработать пакетный файл для построения системы студенческих каталогов с запросом на создание каталогов требуемых курсов, групп и запросом максимального числа пользователей в группе. Номера курсов и шифры групп - в командной строке.

7. Разработать пакетный файл для перехода студента в личный каталог. Группа выбирается из меню, курс передается в командной строке.

8. Разработать пакетный файл для установки даты и времени (параметры – в командной строке).

9. Разработать пакетный файл для очистки студенческих каталогов: удаления файлов \*.BAK, \*.TMP и др. Номера очищаемых курсов передаются в командной строке.

10. Разработать пакетный файл для вывода (на экран, файл, принтер) списка файлов из каталогов студентов. Шифр группы — из командной строки, направление вывода — из меню.

11. Разработать пакетный файл для перехода в каталог студента, если он существует и его архивирования.

12. Разработать пакетный файл для вывода текстового файла на экран по страницам. Имя файла передается в командной строке.

13. Разработать пакетный файл для перезаписи файлов документов (\*.doc, \*.txt) из одного каталога в другой с обновлением.

14. Разработать пакетный файл для копирования всех файлов документов (\*.doc, \*.txt) из всех студенческих подкаталогов в директорию «Мои документы».

15. Разработать пакетный файл для проверки наличия фай-

## Операционные системы

лов документов (\*.doc, \*.txt) в данном подкаталоге (имя – в параметрах). В случае положительного ответа – вывести их список.