# МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Чувашский государственный университет имени ИОН. Ульянова»** (ФГБОУ ВО «ЧГУ им. И. Н. Ульянова»)

Факультет прикладной математики, физики и информационных технологий. Кафедра дискретной математики и информатики.

**Операционные системы и оболочки**

Доклад на тему

«Автоматизация рутинных задач (FOR, IF, COPY, PATH)»

Выполнил:

студент группы ФМ-12-22  Волков Б.А.

Проверил: кандидат педагогических наук, доцент

 Речнов А. В.

г. Чебоксары 2024

**ВВЕДЕНИЕ**

В современном информационном мире, где объемы данных растут с каждым днем, автоматизация играет ключевую роль в оптимизации рабочих процессов и повышении производительности. Одним из инструментов, облегчающих повседневные задачи для пользователей операционных систем Windows, являются команды «FOR, IF, COPY, PATH». Эти команды предоставляют простое, но мощное средство для работы с файлам или папкам в формате, легко воспринимаемом программами на языках программирования, особенно в условных конструкциях типа «IF». Мы рассмотрим принцип работы этих команд, их преимущества и области применения, а также примеры использования, демонстрирующие их эффективность в автоматизации рутинных задач на платформе Windows.

**Команда FOR**

**FOR**- организация циклической обработки результатов выполнения других команд, списков, и строк в текстовых файлах. Команда FOR используется для выполнения команды, заданной в виде параметра, для каждого элемента из набора. В качестве элементов могут использоваться файлы, каталоги, наборы строк.

**Формат командной строки:**

FOR %переменная IN (набор) DO команда [параметры]

**Параметры:**

* %переменная - Однобуквенный подставляемый параметр.
* (набор) - Определяет набор, состоящий из одного или нескольких элементов, обрабатываемых командой.
* команда - Команда, которую следует выполнить для каждого элемента набора.
* параметры - Параметры для команды, выполняемой по отношению к элементам набора.

В пакетных файлах для команды FOR используется запись %%переменная вместо %переменная. Имена переменных учитывают регистр букв (%i отличается от %I).

**Поддерживаются также дополнительные форма команды FOR:**

* FOR /D %переменная IN (набор) DO команда [параметры] - Ключ /D задает в качестве набора имена каталогов (не файлов).
* FOR /R [[диск:]путь] %переменная IN (набор) DO команда [параметры] - Ключ /R задает выполнение команды для каталога [диск:]путь, а также для всех подкаталогов этого пути. Если после ключа /R не указано имя каталога, используется текущий каталог. Если набор - это одиночный символ точки (.), команда просто перечисляет дерево каталогов).
* FOR /L %переменная IN (начало,шаг,конец) DO команда [параметры]- Ключ /L задает обработку набора из последовательности чисел с заданными началом, концом и шагом приращения. Так, набор (1,1,5) раскрывается в (1 2 3 4 5), а набор (5,-1,1) - в (5 4 3 2 1)
* FOR /F ["ключи"] %переменная IN (набор-файлов) DO команда [параметры]
* FOR /F ["ключи"] %переменная IN ("строка") DO команда [параметры]
* FOR /F ["ключи"] %переменная IN ('команда') DO команда [параметры]

Ключ /F задает обработку файлов, строковых значений или результатов стандартного вывода другой команды. Набор файлов - содержит имена одного или нескольких файлов, которые по очереди открываются, читаются и обрабатываются. Обработка состоит в чтении файла, разбивке его на отдельные строки текста и разборе каждой строки в ноль или более подстрок. Затем вызывается тело цикла "for", при выполнении которого каждая найденная подстрока используется в качестве значения переменной. По умолчанию ключ /F выделяет из каждой строки каждого файла первую отделенную пробелами подстроку. Пустые строки в файле пропускаются. Необязательный параметр "ключи" служит для переопределения правил разбора по умолчанию. Он представляет собой заключенную в кавычки строку, содержащую одно или несколько ключевых слов для определения параметров разбора. Ключевые слова:

* eol=символ - знак начала комментария в конце строки ( признак конца обрабатываемых данных строки). Задается в виде одиночного символа.
* skip=n - число пропускаемых при обработке строк от начала файла.
* delims=xxx - набор разделителей между обрабатываемыми элементами строк. По умолчанию, в качестве разделителей используются пробелы и знаки табуляции.
* tokens=x,y,m-n- номера подстрок из каждой строки, передаваемые в тело цикла "for" для каждой итерации.

Например, для обычного текстового файла, подстроками будут слова, а разделителями подстрок - пробелы или знаки табуляции. При использовании этого ключа выделяются дополнительные имена переменных. Формат m-n представляет собой диапазон подстрок с номерами от m по n. Если последний знак в строке tokens= является звездочкой, то создается дополнительная переменная, значением которой будет весь оставшийся текст в строке после разбора последней подстроки.

usebackq - режим обработки кавычек. Строка, заключенная в обратные кавычки, выполняется как команда, строка, заключенная в прямые одиночные кавычки, является строкой символов, а двойные кавычки могут использоваться для задания имен файлов, содержащих пробелы. В команде FOR возможно использование ссылок на переменные.

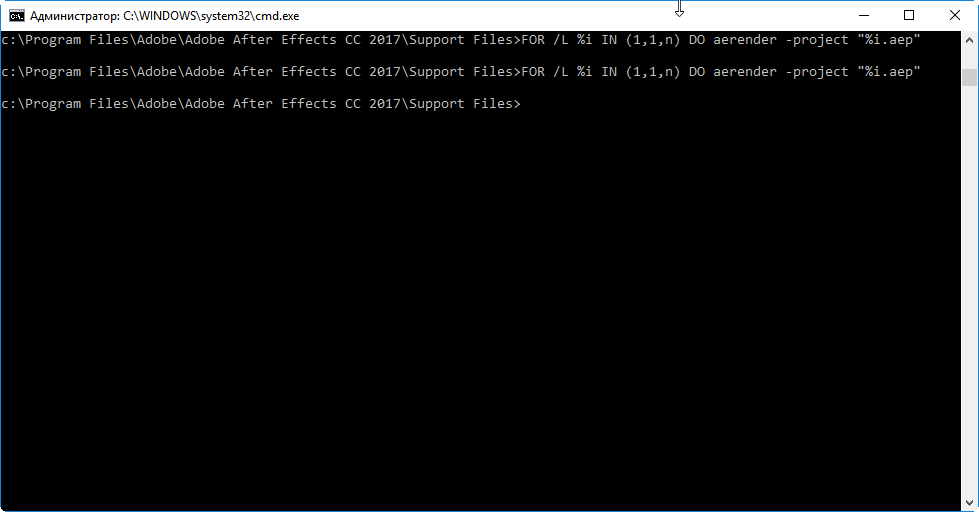
**Допускается применение следующих синтаксических конструкций:**

* %~I - из переменной %I удаляются обрамляющие кавычки (")
* %~fI - переменная %I расширяется до полного имени файла
* %~dI - из переменной %I выделяется только имя диска
* %~pI - из переменной %I выделяется только путь к файлу
* %~nI - из переменной %I выделяется только имя файла
* %~xI - из переменной %I выделяется расширение имени файла
* %~sI - полученный путь содержит только короткие имена
* %~aI - переменная %I расширяется до атрибутов файла
* %~tI - переменная %I расширяется до даты /времени файла
* %~zI - переменная %I расширяется до размера файла
* %~$path:I - проводится поиск по каталогам, заданным в переменной среды path, и переменная %I заменяется на полное имя первого найденного файла. Если переменная path не определена или в результате поиска не найден ни один файл, то этот модификатор заменяется на пустую строку.

**При объединении нескольких операторов можно получить следующие результаты:**

* ~dpI - переменная I раскрывается в имя диска и путь
* ~nxI - переменная I раскрывается в имя файла и его расширение
* ~fsI - переменная I раскрывается в полный путь с короткими именами
* ~dp$path:I - проводится поиск по каталогам, заданным в переменной среды path, и переменная I раскрывается в имя диска и путь к первому найденному файлу.
* ~ftzaI - переменная I раскрывается в строку, подобную выдаваемой командой DIR

Пример использование команды for в консоли(поочередная передача файлов):



**Команда If**

**IF** – оператор условного выполнения команд в командных файлах Windows

**Формат командной строки:**

* IF [NOT] ERRORLEVEL число команда
* IF [NOT] строка1==строка2 команда
* IF [NOT] EXIST имя\_файла команда

**Параметры:**

* NOT - Указывает, что Windows должна выполнить эту команду, только если условие является ложным.
* ERRORLEVEL число - Условие является истинным, если код возврата последней выполненной программы не меньше указанного числа.
* строка1==строка2 - Условие является истинным, если указанные строки совпадают.
* EXIST имя\_файла - Условие является истинным, если файл с указанным именем существует.
* команда - Задает команду, выполняемую при истинности условия. За этой командой может следовать ключевое слово ELSE. В случае, если указанное условие ложно, будет выполнена команда, находящаяся после слова ELSE.

Предложение ELSE должно располагаться в той же строке, что и команда, следующая за ключевым словом IF. Например:

IF EXIST имя\_файла. (

del имя\_файла.

) ELSE (

echo имя\_файла. missing.

)

**Изменение команды IF при включении расширенной обработки команд:**

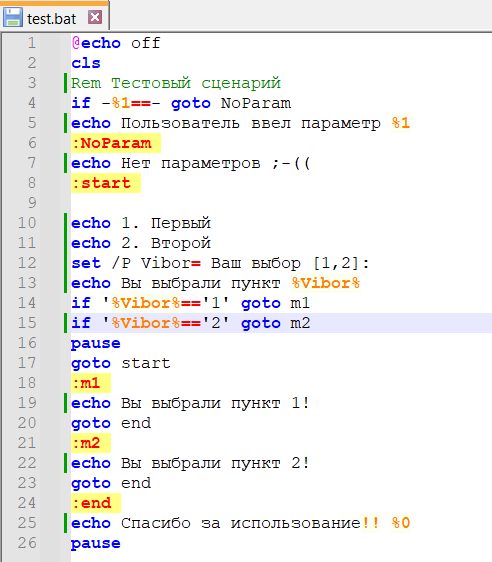
* IF [/I] строка1 оператор\_сравнения строка2 команда
* IF CMDEXTVERSION число команда
* IF DEFINED переменная команда,

**где оператор\_сравнения принимает следующие значения:**

* EQU - равно
* NEQ - не равно
* LSS - меньше
* LEQ - меньше или равно
* GTR - больше
* GEQ - больше или равно,

а ключ /I, если он указан, задает сравнение текстовых строк без учета регистра. Ключ /I можно также использовать и в форме строка1==строка2 команды IF. Сравнения проводятся по общему типу данных, так что если строки 1 и 2 содержат только цифры, то обе строки преобразуются в числа, после чего выполняется сравнение чисел.

Пример использования команды if:



**Команда Copy**

**Copy** – копирование одного или нескольких файлов в командной строке Windows.

**Формат командной строки:**

* COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/A | /B] источник [/A | /B] [+ источник [/A | /B] [+ ...]] [результат [/A | /B]]

**источник** - Имена одного или нескольких копируемых файлов.

**результат** - Каталог и/или имя для конечных файлов.

* /A - Файл является текстовым файлом ASCII.
* /B - Файл является двоичным файлом.
* /D - Указывает на возможность создания зашифрованного файла
* /V - Проверка правильности копирования файлов.
* /N - Использование, если возможно, коротких имен при копировании файлов, чьи имена не удовлетворяют стандарту 8.3.
* /Y - Подавление запроса подтверждения на перезапись существующего конечного файла.
* /-Y - Обязательный запрос подтверждения на перезапись существующего конечного файла.
* /Z - Копирование сетевых файлов с возобновлением. Ключ /Y можно установить через переменную среды COPYCMD.
* Ключ /-Y командной строки переопределяет такую установку.

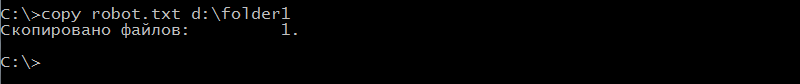
По умолчанию требуется подтверждение перезаписи существующих файлов, если только команда COPY не выполняется в пакетном файле.

Команда copy позволяет объединить несколько файлов в один. Чтобы объединить файлы, укажите один конечный и несколько исходных файлов, используя подстановочные знаки или формат "файл1+файл2+файл3+...".

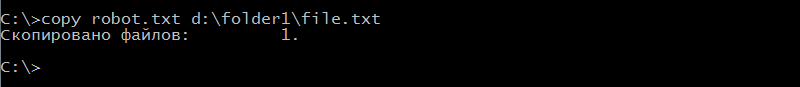
По умолчанию, используется копирование файлов как двоичных "один в один", поэтому параметр /B можно не задавать. Параметр /A необходим, если в результатах копирования должны обрабатываться специальные символы, применяемые в текстовых файлах.

Ключ /N используется для совместимости с программами MS-DOS, не поддерживающих длинные имена файлов.

Для примера, создадим на диске “C” файл robot.txt, а на диске “D” папку folder1. Тогда команда для копирования файла robot.txt в папку folder1 будет выглядеть так:



Если необходимо скопировать файл, изменив его имя, то необходимо прописать эту команду:



В этом случае файл robot.txt будет скопирован в папку folder1, но уже с именем file.txt.

**Команда Path**

**Path**- отобразить или изменить пути поиска исполняемых

файлов.

Команда PATH используется для указания или просмотра путей поиска исполняемых файлов. Пути поиска представляют собой строки, определяющие перечень каталогов файловой системы, в которых находятся исполняемые файлы (файлы с расширением .bat, .cmd, .exe, .vbs и т.п. ), разделенные точкой с запятой ; Например, C:\windows;C:\windows\system32 - определяет пути поиска C:\windows и C:\windows\system32. Если вы в командной строке набираете program.exe без явного указания пути, то для запуска файла program.exe выполняется его поиск в текущем каталоге, и если он не найден, то в каталоге C:\windows, если и там не найден – в каталоге C:\windows\system32. Если же исполняемый файл будет в обоих каталогах, то выполнится запуск из того, что определен ранее - C:\windows. Значение переменной среды PATH содержит пути поиска исполняемых файлов определенный на данный момент времени.

**Формат командной строки:**

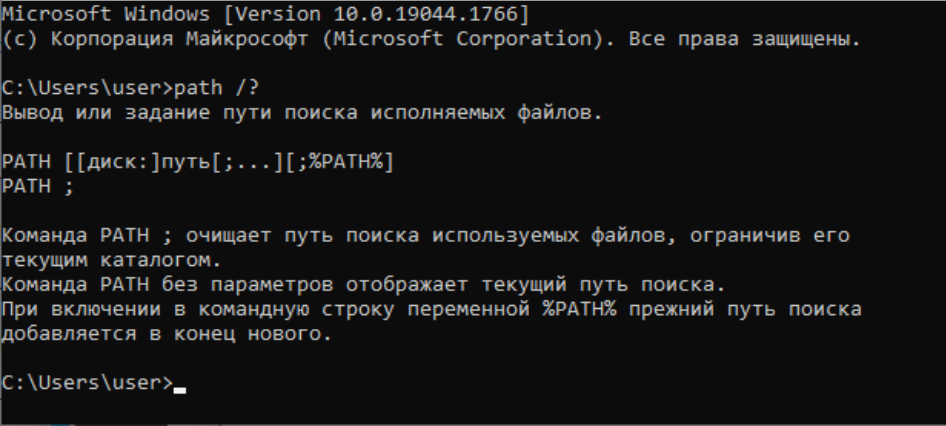
* PATH ; - очистить путь поиска используемых файлов, ограничив его текущим каталогом.
* PATH [[диск:]путь[;...][;] - установит ь пути поиска исполняемых файлов.

Команда PATH без параметров отображает текущий путь поиска. В командную строку допускается включение переменной %PATH% , задающей прежний путь поиска.

**Примеры использования:**

* path /? - отобразить подсказку по использованию команды.
* path - отобразить пути поиска исполняемых файлов
* path %PATH%;C:\Scripts - добавить путь C:\Scripts в конец существующего списка каталогов для поиска исполняемых файлов.
* path C:\scripts;%PATH% - добавить путь C:\Scripts в начало существующего списка каталогов для поиска исполняемых файлов.

При выполнении команды PATH, значение передаваемых ей параметров не анализируется и воспринимается как обычная строка символов, поэтому, например, трижды выполнив команду path C:\scripts;%PATH% вы создадите 3 записи для пути C:\Scripts. Значение переменной PATH, измененное командой действует только на момент текущего сеанса командной строки. Для постоянного изменения системных и пользовательских переменных среды, в том числе, и путей поиска, используется команда SetX. В постоянно действующих путях поиска не стоит указывать каталоги сменных носителей (дискет, CD/DVD, карты памяти и т.п.)



**ЛИТЕРАТУРА**

1. Командная строкаCMD. Учебное пособие | Арьков Валентин Юрьевич, 2021г.
2. Операционные системы. Учебное пособие | Арьков Валентин Юрьевич, 2021г.
3. Microsot Windows 11. Первое знакомство | Колисниченко Денис Николаевич. БХВ, 2022г.
4. Операционные системы и оболочки. Учебное пособие | Малахов Сергей Валерьевич. Лань, 2023г.
5. Современная PowerShell | Попов Андрей Владимирович, 2022г.
6. Самоучитель Windows 11 | Колисниченко Денис Николаевич, 2022г.
7. Операционные системы | Бубнов Сергей Алексеевич, Бубнов Алексей Алексеевич, 2024г.