**ИИТ БГУИР**

**Факультет повышения квалификации и переподготовки**

**Кафедра МПСС**

**Дисциплина: Архитектура операционных систем**

**Отчет по лабораторной работе**

**Тема:** **Автоматизация работы в консоли Windows**

**с помощью командных файлов**

**Выполнил слушатель гр. 30333-2**

**Калевич Сергей Михайлович**

**Минск 2023**

**Содержание**

[Теоретические сведения 3](#_Toc153999104)

[Выполнение работы 5](#_Toc153999105)

[Простой пакетный файл 5](#_Toc153999106)

[Пакетный файл с параметрами 6](#_Toc153999107)

[Пакетный файл с обработкой списков параметров 8](#_Toc153999108)

[Пакетный файл с вызовом другого пакетного файла 10](#_Toc153999109)

[Комплексное упражнение 11](#_Toc153999110)

[Задания для самостоятельной работы 14](#_Toc153999111)

[Вариант 1 14](#_Toc153999112)

[Вариант 2 15](#_Toc153999113)

[Вариант 3 16](#_Toc153999114)

[Вариант 4 17](#_Toc153999115)

[Вариант 5 18](#_Toc153999116)

[Вариант 6 19](#_Toc153999117)

[Вариант 7 20](#_Toc153999118)

[Вариант 8 24](#_Toc153999119)

[Вариант 9 25](#_Toc153999120)

[Вариант 10 26](#_Toc153999121)

[Выводы 27](#_Toc153999122)

Теоретические сведения

**SET**

При объявлении переменных между переменной и знаком равно не должно быть пробелов, если это строка. Например:

* SET var = Hello World – неправильно (var=undefined);
* SET var= Hello World – правильно, но первый символ – пробел;
* SET /A var = 5 – правильно;
* SET var = 5 неправильно (5 - строка) (var=undefined).

**Комментарии**

Альтернативный способ комментирования: двойное двоеточие – **::** (:: *комментарий*)

**TYPE**

Команда TYPE может получать на вход данные из консоли и перенаправлять в файл, что позволяет не перезаписывать файл. Поведение схоже с командой COPY:

* copy con lists.txt – перезаписывает файл;
* TYPE CON >> lists.txt – не перезаписывает файл.

**@chcp 65001**

Позволяет корректно использовать русский язык, так как .bat файлы сохраняются в кодировке UTF-8. Команду следует писать в самой первой строке.

**Нахождение субстроки в строке**:

***if not x%str1:bcd=%==x%str1% echo It contains bcd***

*%str1:bcd=%* замещает *bcd* в *str1* пустой строкой, делая новую строку отличной от оригинала. Если оригинал не содержит *bcd* строку – измененная версия будет идентичной. *x* перед гарантирует, что строка *bcd* работает правильно. Это также дает защиту от некорректных первых символов.

**if exist c:mydir\nul**

Такая проверка работает только с папками не содержащими пробелы, так как требуется использование кавычек. Это относится и к файлам. Альтернативный способ – произвести проверку наличия атрибутов.

**if “%i”==”” do something**

Если значение параметра с пробелом, происходит ошибка. Альтернатива: **if [%i]==[]**.

**Enable and Disable Delayed Expansion**

*Enableelayeexpansion* указывает cmd распознавать синтаксис *!var!* который обращается к текущему значению *var*. Disabledelayedexpansion отключает эту возможность, поэтому *!var!* становится обычной строкой.

Внутри оператора блока (серии операторов в скобках) весь блок анализируется, а затем выполняется. Любой *%var%* внутри блока будет заменен значением этой переменной во время анализа блока - до того, как блок будет выполнен - то же самое относится и к *FOR ... DO* (блоку). Использование *!var!* вместо *%var%* получает доступ к измененному значению *var*.

Рассмотрим такой код:

***set getOption=yes***

***If "%getOption%" equ "yes" (***

***set /P option=Enter option:***

***echo Option read: %option%***

***)***

Предыдущий код НЕ будет работать, поскольку значение *%option%* заменяется только один раз при анализе команды *IF* (до ее выполнения). Необходимо «отложить» расширение значения переменной до тех пор, пока команда *SET /P* не изменит значение переменной:

***set getOption=yes***

***setlocal EnableDelayedExpansion***

***If "%getOption%" equ "yes" (***

***set /P option=Enter option:***

***echo Option read: !option!***

***)***

Также можно проверить:

***set var=Before***

***set var=After & echo Normal: %var% Delayed: !var!***

Вывод: ***Normal: Before Delayed: After***

Выполнение работы

Простой пакетный файл

Создайте пакетный файл с именем P1.bat, содержащий команды для формирования дерева директорий и копирования файлов в соответствии с заданием предыдущей лабораторной работы.



Выполните команду P1 и убедитесь, что она реализует поставленную задачу.



Выполните команду P1 еще раз, рассмотрите и объясните изменения в информации, выводимой при повторном выполнении.



Новая директория или файл не создаются, если директория либо файл с таким именем уже существуют.

Откройте в редакторе файл P1.bat и вставьте в начало файла команду @ECHO OFF. Сохраните файл и выполните команду Р1. Проанализируйте отличия в выполнении пакетного файла в этот раз от предыдущего.



Команда @ECHO OFF блокирует отображение команд в консоли, но они все равно выполняются в фоновом режиме.

Пакетный файл с параметрами

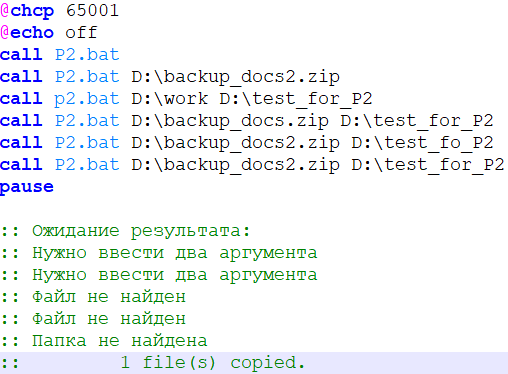
Создайте пакетный файл P2.bat, предусматривающий ввод параметров в командной строке запуска этого файла на выполнение.

Этот файл должен копировать под своим именем файл, указанный в качестве первого параметра в команде P2, в директорию, указанную в качестве второго параметра. Необходимо обеспечить проверку наличия необходимых параметров, проверку существования копируемого файла и существования директории назначения. Следует выводить необходимые сообщения и завершать работу, если продолжение невозможно.



Проверьте выполнение команды P2 с различными значениями параметров. Обеспечьте тестирование с полным перебором всех вариантов сочетаний параметров.

Тестируем при помощи файла **P2\_test.bat**.



Пакетный файл с обработкой списков параметров

Создайте пакетный файл P3 для запуска программ. Командные строки запускаемых программ передаются через параметры строки запуска P3. Количество параметров запускаемых программ заранее не определено. Например, запуск по очереди программ Edit, Dir и Mem, перечисленных со своими параметрами в командной строке запуска P3, как показано далее:

**d:\work\d1>P3.bat Edit f1.txt ; Dir d:\work ; Mem /d.**

Для выполнения задания требуется определить является ли значение параметра:

* исполняемым файлом (.exe);
* командой cmd.exe;
* Параметром исполняемого файла либо параметром команды cmd.exe.

Так как пакетный файл выполняется построчно, следует сначала определить является ли параметр исполняемым файлом или командой cmd.exe. Затем нужно определить следуют ли за ними их параметры или аргументы. В итоге формируется строка команды. Как только встречается новая команда, создается новая строка.

Таким образом создается следующий алгоритм:

1. первое значение параметра строки запуска образует первую строку;
2. если следующий аргумент команда, создается новая строка, иначе значение конкатенируется и т.д;

Это означает, что следует произвести парсинг строки запуска на отдельные строки. Можно использовать два варианта выполнения программы:

* записывать строки в .bat файл, а затем выполнить этот файл;
* записывать строки в массив, а затем выполнить каждый элемент массива.

**Этап 1**

Проверяем имеет ли значение расширение .exe:

* если да, то записываем первым элементом строки;
* если нет, то проверяем определяется ли значение в PATH;
  + если да, то записываем первым элементом строки;
  + если нет, то проверяем является ли значение командой cmd.exe.

**Этап 2.**

* если вышеуказанные проверки не проходятся, считаем значение аргументом (параметром) команды;
* конкатенируем значение;
* повторяем цикл;
* если встречается новая команда заканчиваем формирование строки;
* добавляем строку в .bat файл либо в массив;
* создаем новую строку;
* повторяем цикл.

**Этап 3.**

Когда все элементы строки запуска исчерпаны, переходим к выполнению программы:

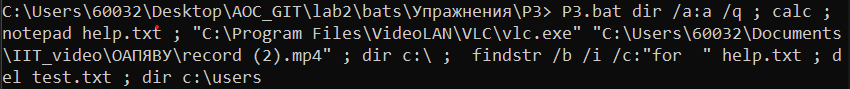
* либо вызываем созданный .bat файл;
* либо циклом вызываем каждый элемент массива.

**Тест.**

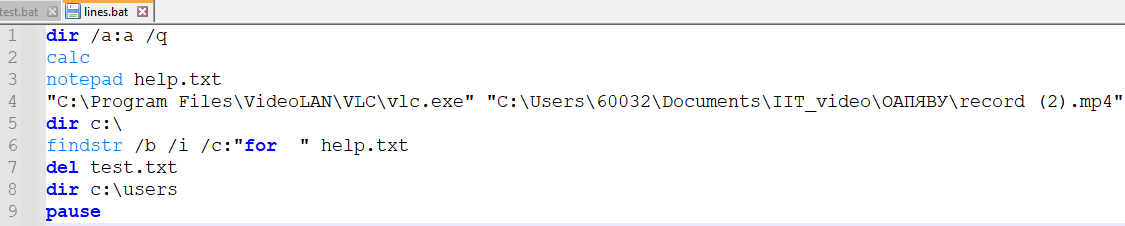
Тестируем на примере следующей строки:

**dir /a:a /q ; calc ; notepad help.txt ; "C:\Program Files\VideoLAN\VLC\vlc.exe" "C:\Users\60032\Documents\IIT\_video\ОАПЯВУ\record (2).mp4" ; dir c:\ ; findstr /b /i /c:"for " help.txt ; del test.txt ; dir c:\users**

**lab2\bats\Упражнения\P3\P3.bat** (создание и вызов файла **lines.bat**):

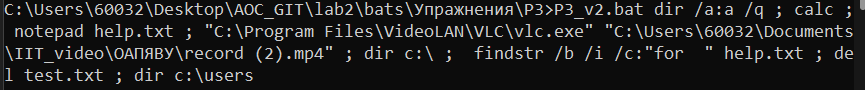


*Вызов P3.bat*



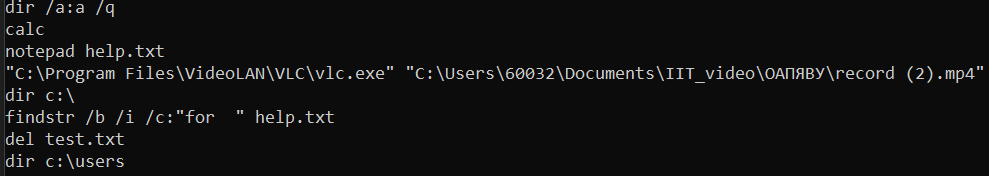
*Файл lines.bat*

**lab2\bats\Упражнения\P3\P3\_v2.bat** (создание массива и вызов каждого его элемента):



*Вызов P3\_v2.bat*

Также проверяем правильно ли заполнился массив:



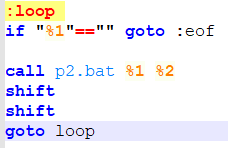
*Вывод всех элементов массива в консоль*

Как видно, парсинг прошел успешно в обоих случаях. Выполнение также прошло корректно.

Пакетный файл с вызовом другого пакетного файла

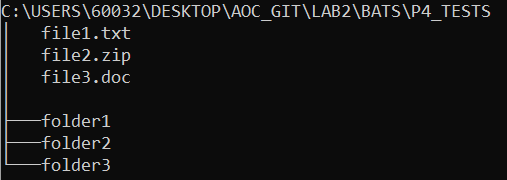
Создайте пакетный файл P4, осуществляющий копирование нескольких файлов в заданные директории. Файлы и директории задаются парами параметров: файл1 – директория1, файл2 – директория2 и т. д. Для выполнения копирования используйте пакетный файл P2.

Решением задачи является использование двойного сдвига в цикле:

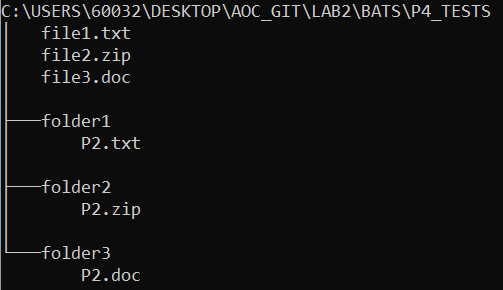


Тестируем при помощи файла **P4\_test.bat**.

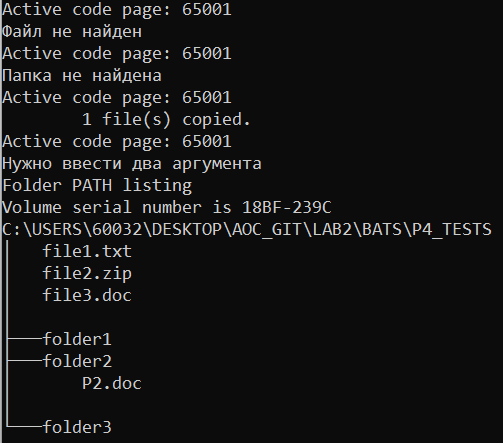
Для проверки используется следующее дерево данных:



Сначала проверяем программу с корректными параметрами:



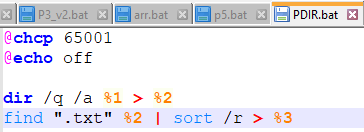
Затем проверяем обнаружение ошибок ввода:



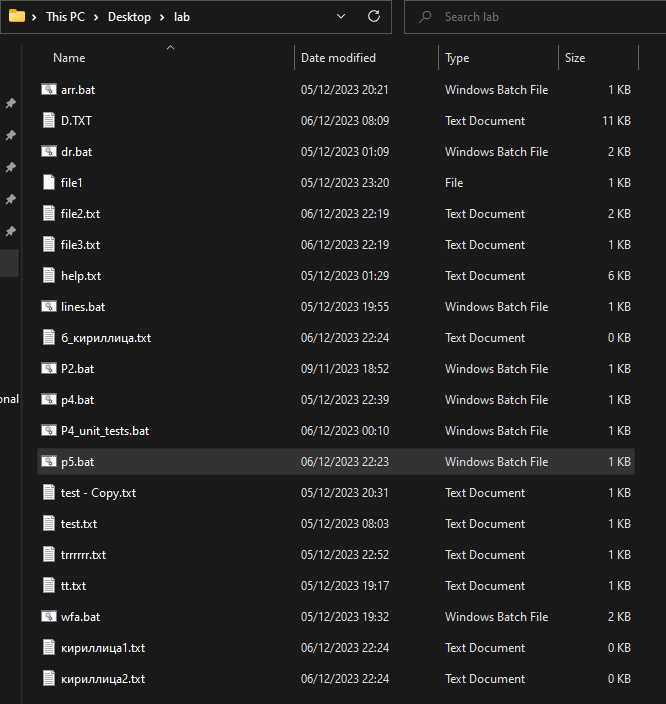
Комплексное упражнение

Создать пакетный файл PDIR для получения информации о директории, задаваемой параметром номер 1 в командной строке, записи полученной информации в файл (имя файла – параметр номер 2 в командной строке), отбора из сохраненного набора строк всех записей о файлах с расширением .txt и сохранения отобранных записей в файле с именем, задаваемым параметром номер 3 в командной строке. Отобранные записи сохранять в обратном алфавитному порядке имен файлов.

**Код пакетного файла**:



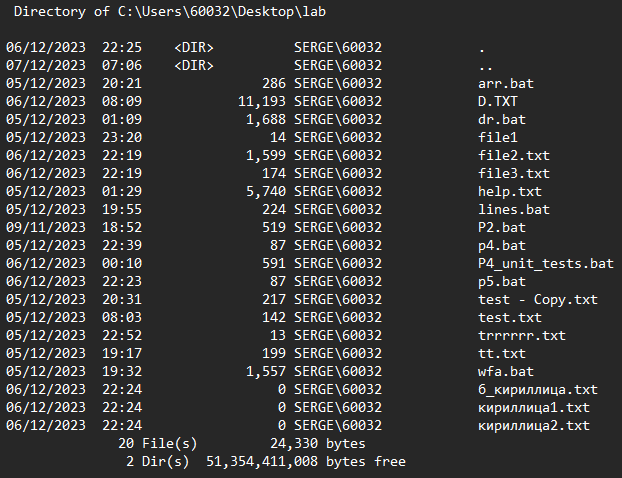
Для примера используем такую директорию:



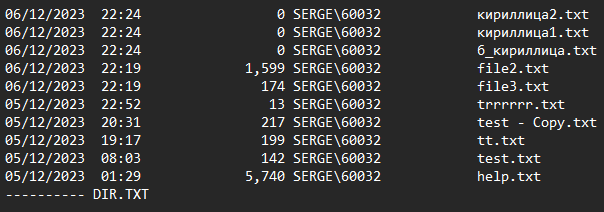
Проверка:



Запуск



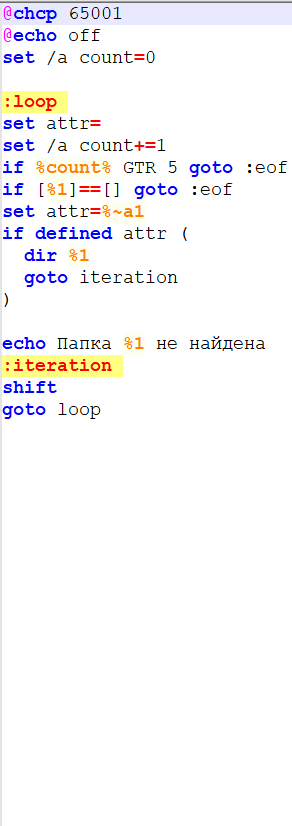
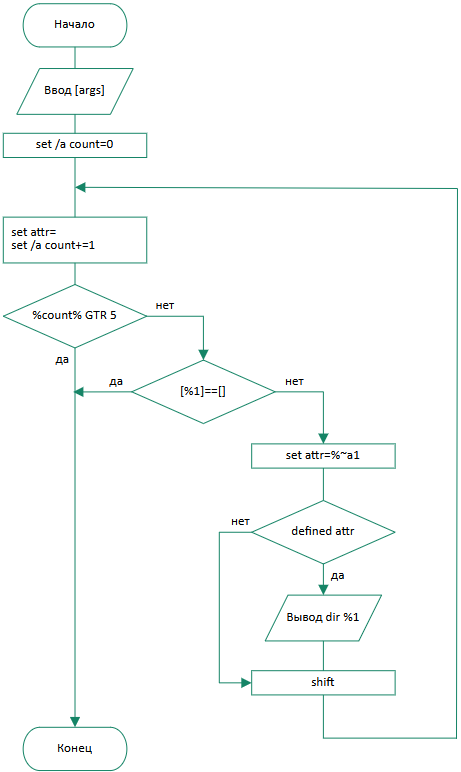
*%2 (dir.txt)*



*%3 (txt.txt)*

Задания для самостоятельной работы

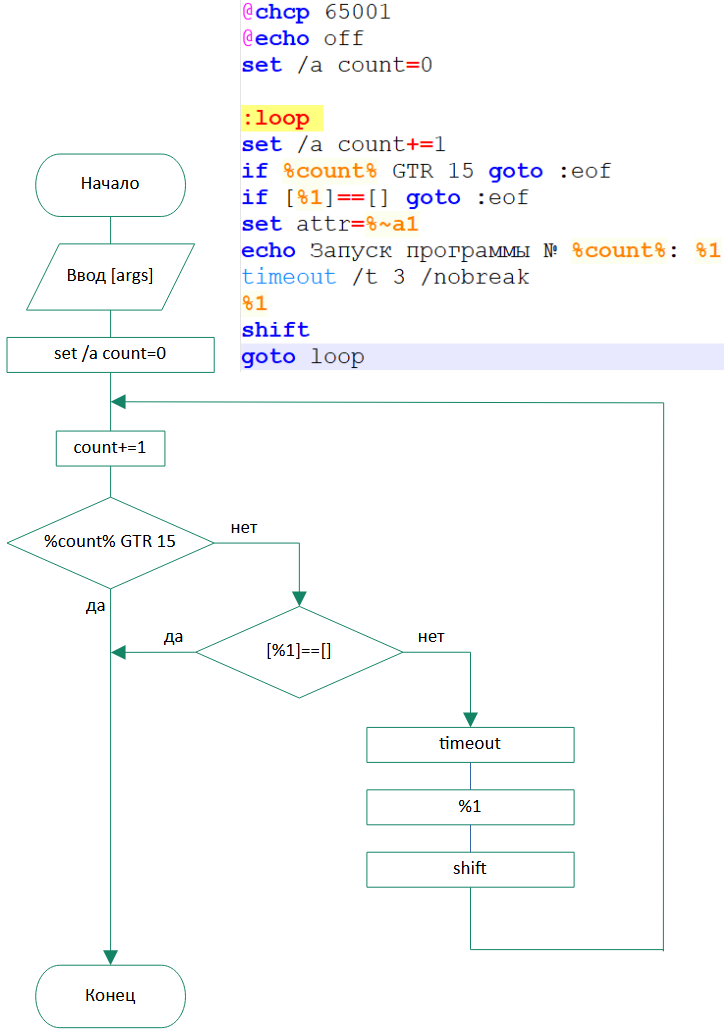
Вариант 1

Создать командный файл P11.bat, выводящий на экран содержимое директорий, перечисленных в командной строке при запуске P11 на выполнение. Если в командной строке перечислено более пяти директорий, для вывода использовать пять первых.  

Тест выполняется скриптом P11\_test.bat.

Вариант 2

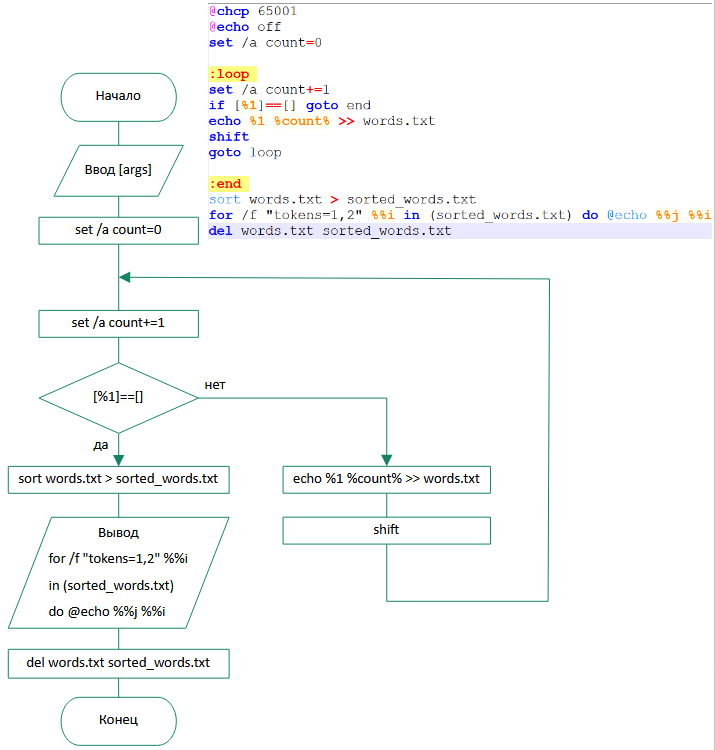
Создать командный файл P21.bat, запускающий на выполнение программы, перечисленные в командной строке (в количестве от 1-й до 15-й). На консоль выводить перед запуском каждой программы ее имя и делать приостановку.



Тест выполняется скриптом P21\_test.bat.

Вариант 3

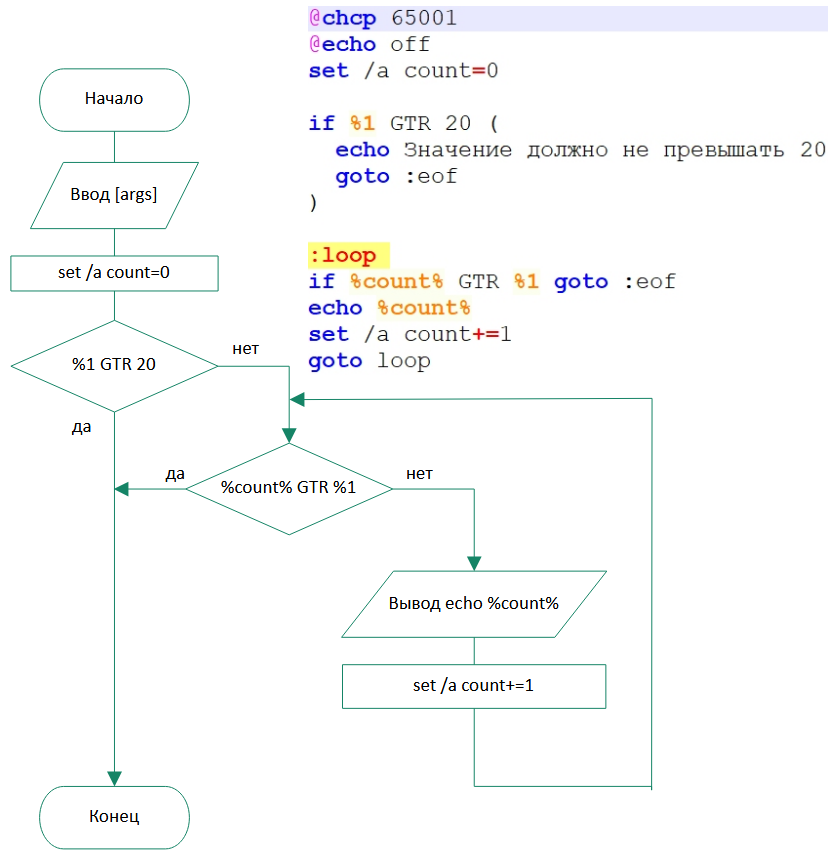
Создать командный файл P3.bat, выводящий на экран построчно слова из командной строки запуска P3 на выполнение. Слова вывести в алфавитном порядке. Каждому слову должен предшествовать порядковый номер, который формируется в пакетном файле.



Тест выполняется скриптом P3\_test.bat.

Вариант 4

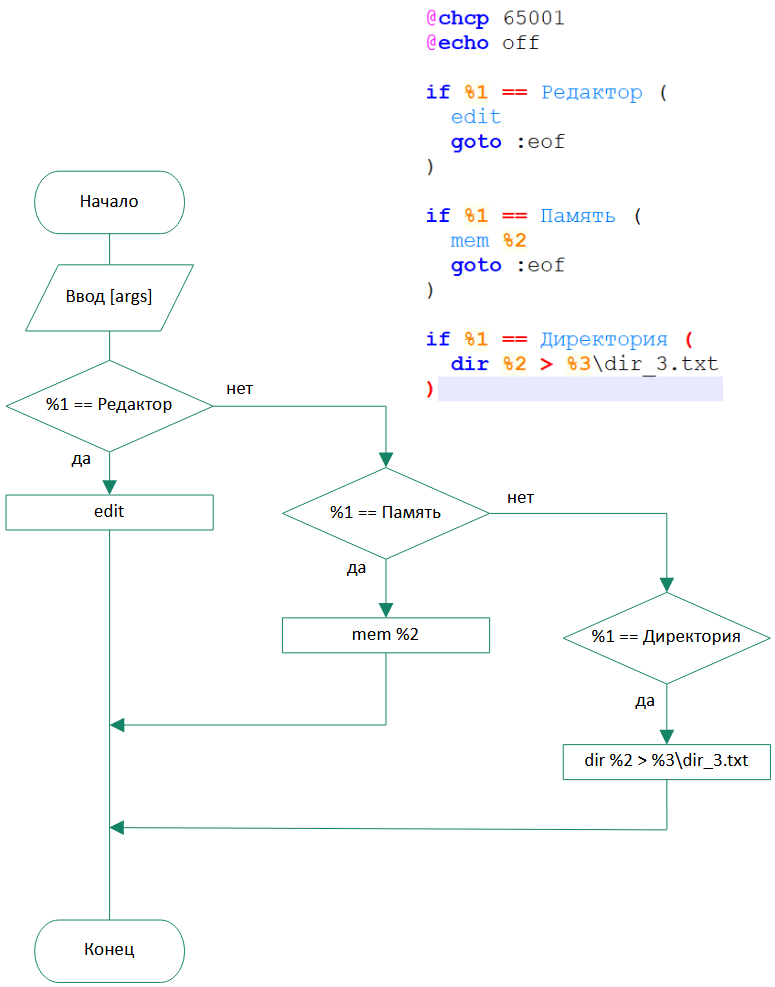
Создать командный файл P4.bat, выводящий на экран построчно порядковые номера в количестве, задаваемом параметром в командной строке запуска P4 на выполнение и не превышающем значения 20. Контролировать корректность параметра.



Тест выполняется скриптом P4\_test.bat.

Вариант 5

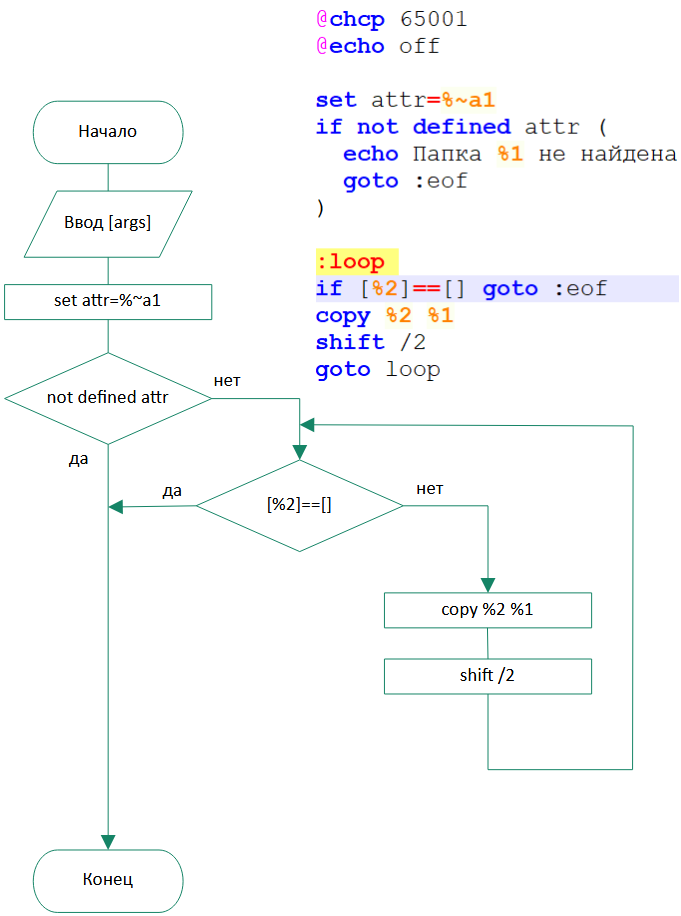
Создать командный файл P5.bat. В качестве первого параметра при запуске этого пакетного файла предполагается ввод одного из слов: Редактор(1), Память(2), Директория(3). В случае 1 необходимо запустить программу Edit. В случае 2 следует запустить программу Mem с ключом, заданным во втором параметре, а в случае 3 – обеспечить запись содержимого директории, заданной вторым параметром, в файл dir\_3.txt, местонахождение которого определяется третьим параметром.



Тест выполняется скриптом P5\_test.bat.

Вариант 6

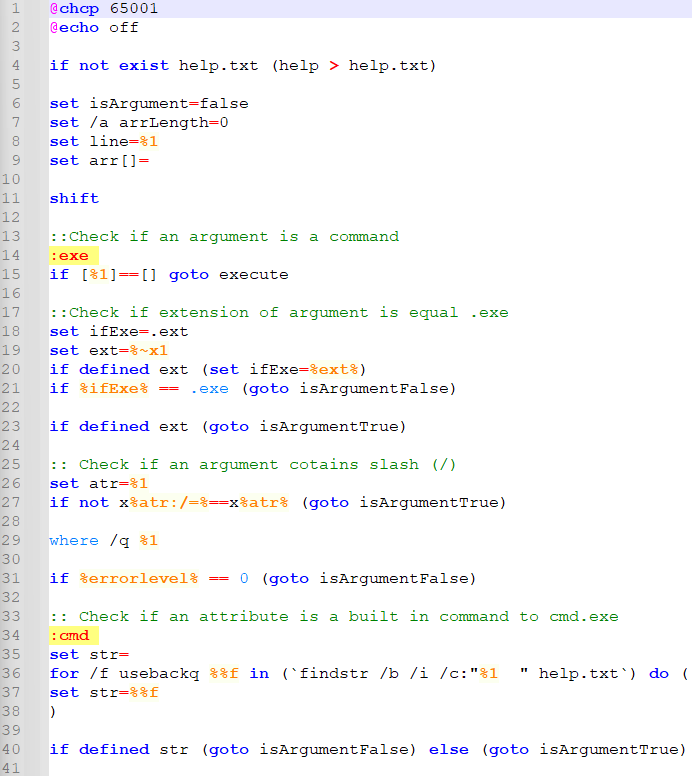
Создать командный файл P61, который будет копировать указанные файлы в заданный каталог. Каталог и перечень любого количества файлов задаются в качестве параметров.

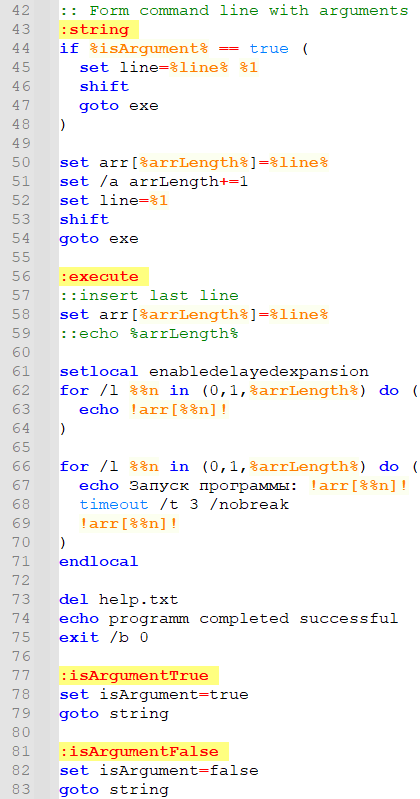


Тест выполняется скриптом P61\_test.bat.

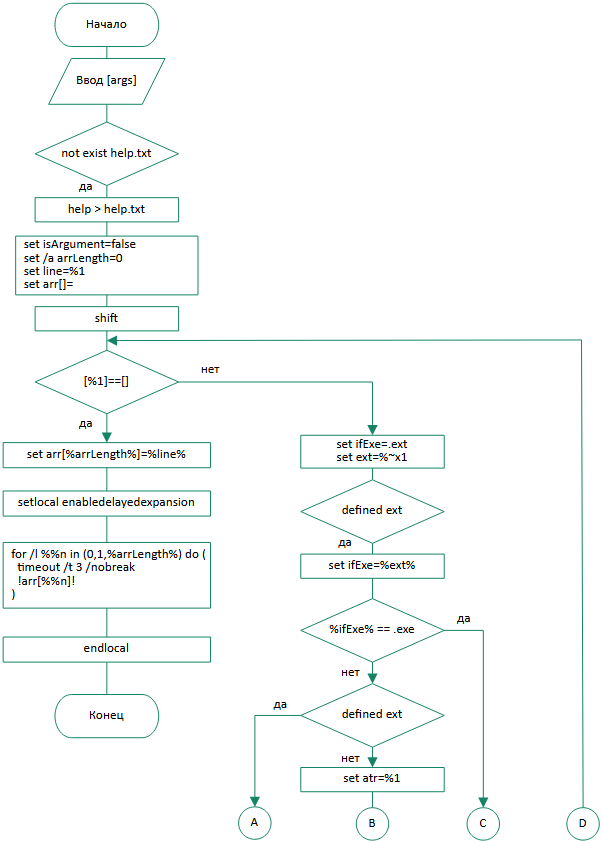
Вариант 7

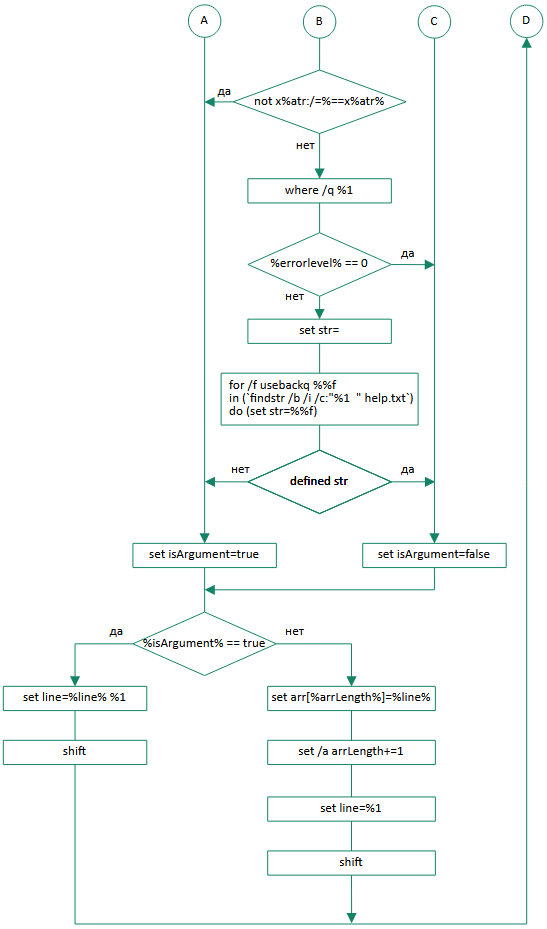
Создать командный файл P71, осуществляющий поочередный (с приостановкой) запуск программ, перечисленных в командной строке запуска этого файла на выполнение. Обеспечить возможность передачи параметров запускаемым программам.





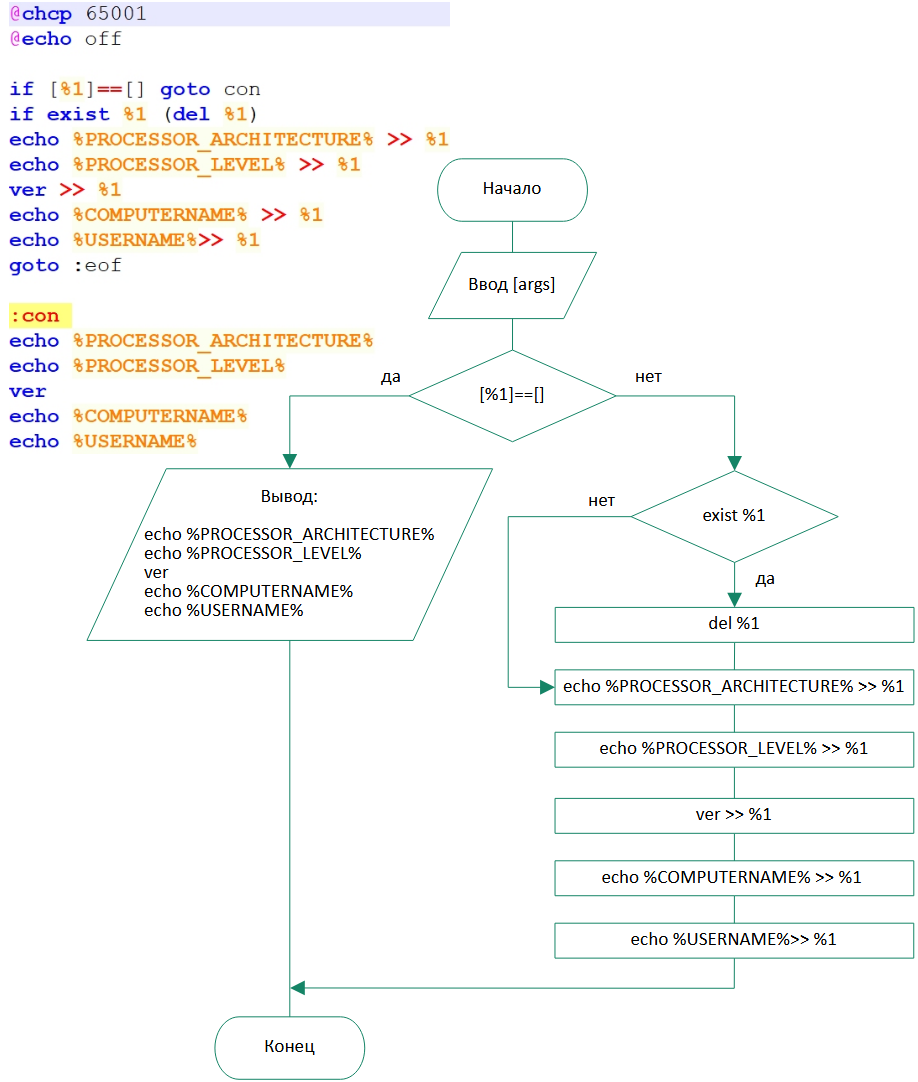
Тест выполняется скриптом P71\_test.bat.





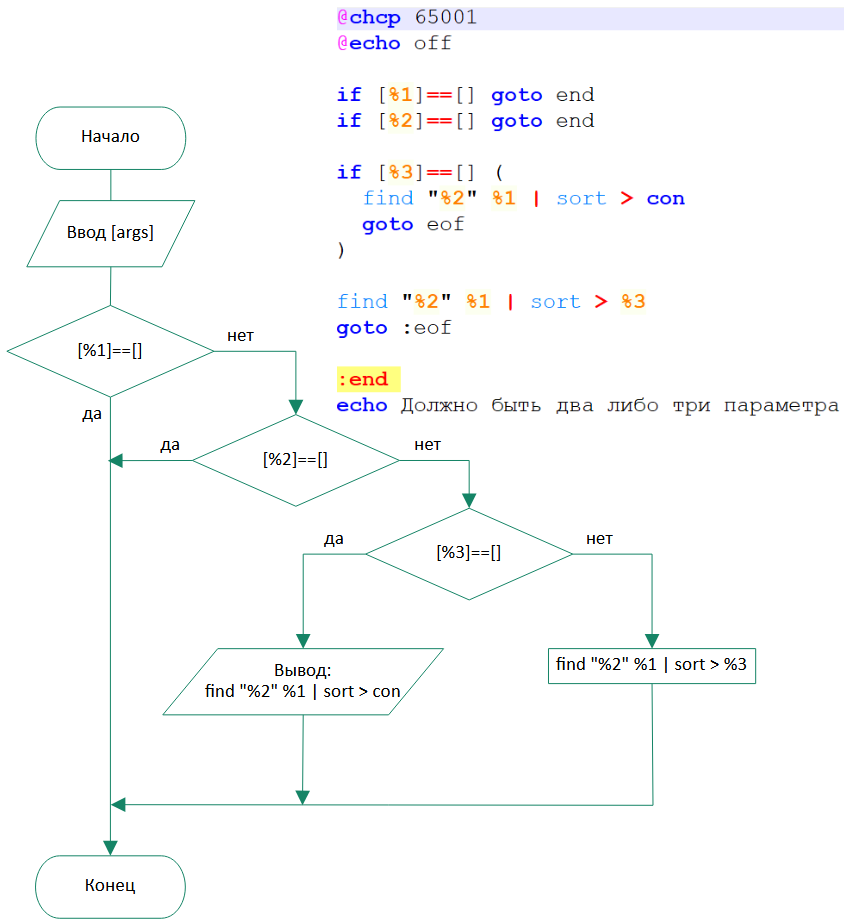
Вариант 8

Создать командный файл P81, обеспечивающий вывод информации из блока окружения: тип и номер процессора, версия системы, имя компьютера, имя пользователя. Текст вывести в файл, задаваемый в качестве параметра. Если файл не задан, то вывести текст на экран.



Вариант 9

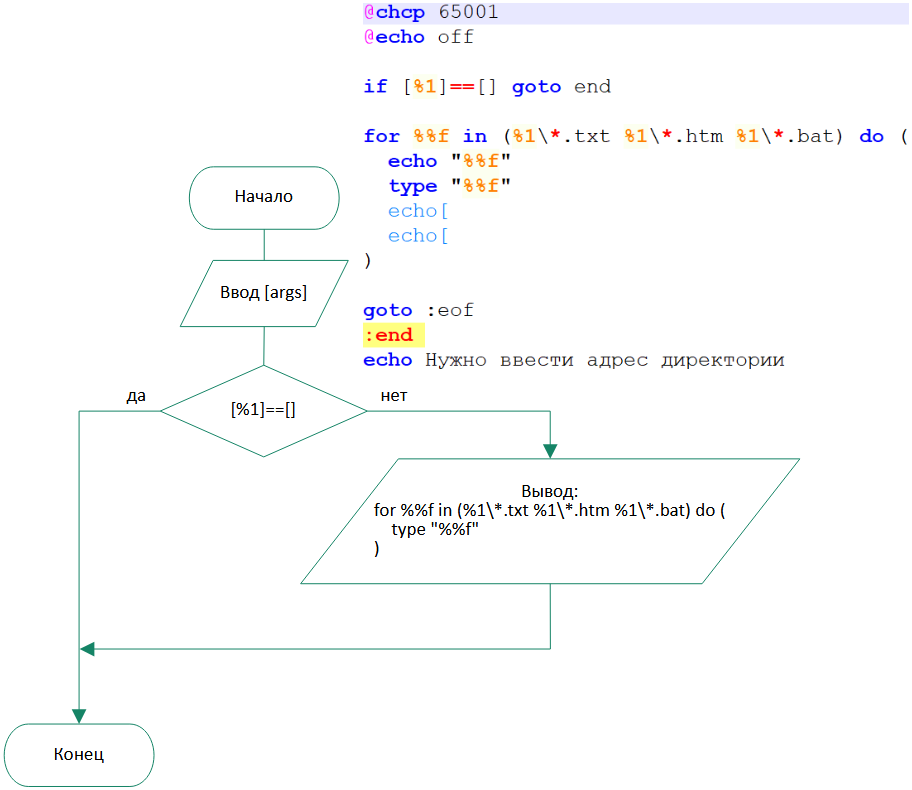
Создать пакетный файл P91, обеспечивающий поиск в указанном первым параметром файле строк, содержащих заданное вторым параметром слово. Найденные строки записать в алфавитном порядке в файл, заданный третьим параметром. Если третий параметр не задан, вывести текст на экран.



Тест выполняется скриптом P91\_test.bat.

Вариант 10

Создать пакетный файл P10, обеспечивающий вывод на экран содержимого всех файлов с расширениями (.txt), (.htm), (.bat) из директории, задаваемой параметром в командной строке.



Выводы

* изучены возможности автоматизации управления операционной системой с помощью командных файлов;
* получены навыки создания командных файлов и применения специфичных для них команд консольного режима на практике.