

Цель работы: Знакомство с языком интерпретатора командной строки ОС MS Windows и командными файлами.

Ход работы:

1. Ознакомиться с теоретическим материалом.
2. Выполнить задания.
3. Ответить на контрольные вопросы.

Задание:

1. Вывод на экран имен всех файлов с указанным расширением, находящихся в каталоге, имя которого задается при запуске командного файла первым параметром. Расширение файлов задается вторым параметром.

2. Среди введенных с клавиатуры целых чисел (использовать SET /P) найти наибольшее и наименьшее. Признаком конца ввода – знак «-».

3. В заданном каталоге и его подкаталогах найти общее количество подкаталогов. На экран вывести только требуемый результат.

4. В каталогах, имена которых заданы первым и вторым параметрами командного файла, найти и вывести на экран имена файлов (расширения могут быть любые), присутствующие, как в первом, так и во втором каталоге. Следует использовать только один оператор FOR.

5. Вычисление и вывод на экран значения факториала целого числа, задаваемого при запуске КФ. Предусмотреть проверку заданного значения и при задании отрицательного значения или значения, превышающего максимально возможную величину, выводить соответствующие сообщения. Для проверки правильности вычислений использовать калькулятор.

6. Просмотр содержимого каталога, указанного первым параметром КФ. Необходимо: 1) создать подкаталоги с именами EXE, TXT, CMD, DOC и OTHER. 2) В каждый подкаталог скопировать файлы с соответствующими расширениями. 3) Пустые подкаталоги удалить.

7. В каталоге, указанном первым параметром КФ, (и его подкаталогах) найти файлы наибольшего и наименьшего размеров. Вывести имена файлов, их размеры и даты создания.

Описание выполнения работы

1. Вывод на экран имен всех файлов с указанным расширением, находящихся в каталоге, имя которого задается при запуске командного файла первым параметром. Расширение файлов задается вторым параметром.

Создадим файл Task1.bat. Напишем код, который будет выполнять данную задачу.



Рисунок 2.1 – Листинг первой задачи

Вызовем программу с помощью командной строки, причём первым параметром укажем нужный нам каталог, а вторым – расширение искомых файлов. Результат выполнения кода, представленного на рисунке 2.1, изображён на рисунке 2.2.

```
PS D:\GH\university\term6\OS\lab2> .\Task1.bat ..\lab2 bat
Task1.bat
Task5.bat
task6.bat
task7.bat
Том в устройстве D имеет метку Local D
Серийный номер тома: 2D7E-8820

Содержимое папки D:\GH\university\term6\OS\lab2

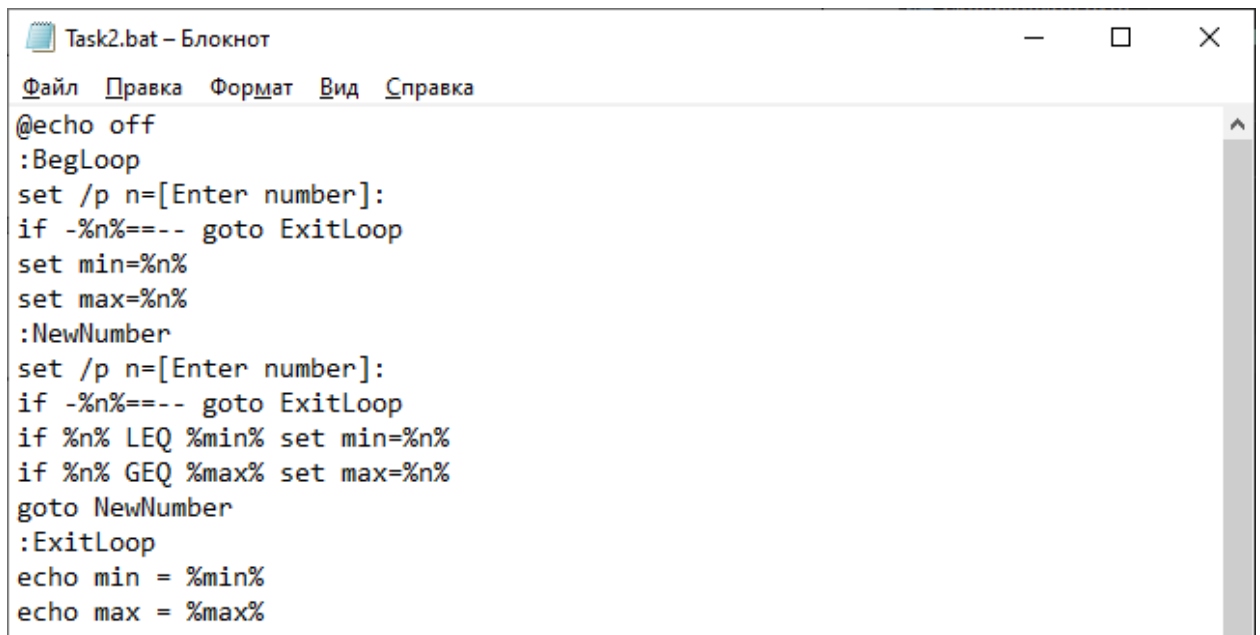
20.03.2023  21:00    <DIR>          .
20.03.2023  21:00    <DIR>          ..
20.03.2023  20:59             336 376 otchet2.docx
20.03.2023  20:55             56 Task1.bat
20.03.2023  21:33             334 Task5.bat
12.03.2023  23:45    <DIR>          task6
12.03.2023  23:45             412 task6.bat
19.03.2023  20:25    <DIR>          task7
19.03.2023  20:40             599 task7.bat
              5 файлов             337 777 байт
              4 папок   158 756 016 128 байт свободно
```

Рисунок 2.2 – Выполнение первой программы

2. Среди введенных с клавиатуры целых чисел (использовать SET /P) найти наибольшее и наименьшее. Признаком конца ввода – знак «-».

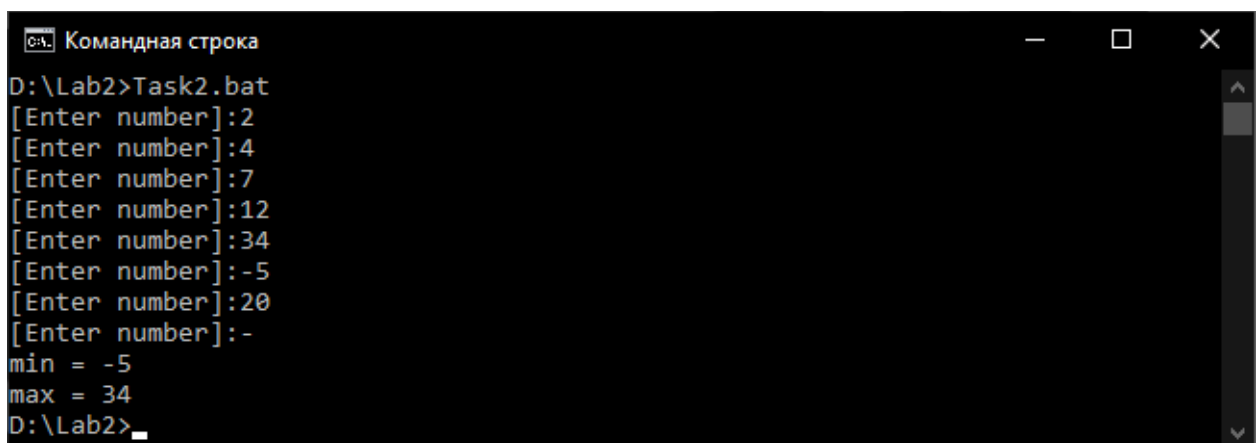
Код программы нахождения наибольшего и наименьшего целых чисел представлен на рисунке 2.3.

Результат работы программы представлен на рисунке 2.4.



```
@echo off
:BeginLoop
set /p n=[Enter number]:
if -%n%==-- goto ExitLoop
set min=%n%
set max=%n%
:NewNumber
set /p n=[Enter number]:
if -%n%==-- goto ExitLoop
if %n% LEQ %min% set min=%n%
if %n% GEQ %max% set max=%n%
goto NewNumber
:ExitLoop
echo min = %min%
echo max = %max%
```

Рисунок 2.3 – Листинг второй задачи



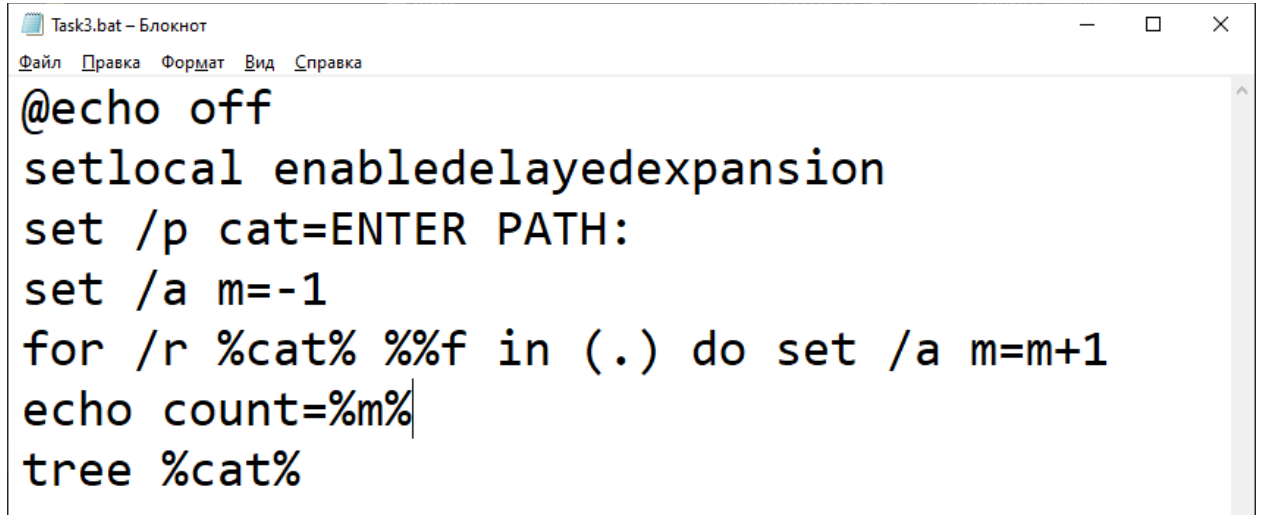
```
D:\Lab2>Task2.bat
[Enter number]:2
[Enter number]:4
[Enter number]:7
[Enter number]:12
[Enter number]:34
[Enter number]:-5
[Enter number]:20
[Enter number]:-
min = -5
max = 34
D:\Lab2>
```

Рисунок 2.4 – Выполнение второй программы

3. В заданном каталоге и его подкаталогах найти общее количество подкаталогов. На экран вывести только требуемый результат.

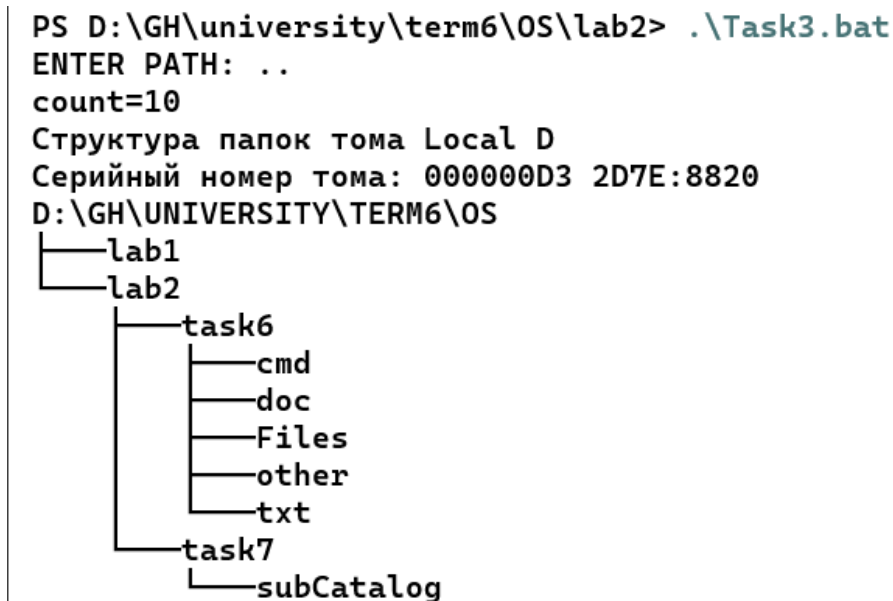
Листинг программы, выполняющей данное задание, представлен на рисунке 2.5.

Результат выполнения это кода изображен на рисунке 2.6.



```
Task3.bat – Блокнот
Файл  Правка  Формат  Вид  Справка
@echo off
setlocal enabledelayedexpansion
set /p cat=ENTER PATH:
set /a m=-1
for /r %cat% %%f in (.) do set /a m=m+1
echo count=%m%
tree %cat%
```

Рисунок 2.5 – Код третьей задачи

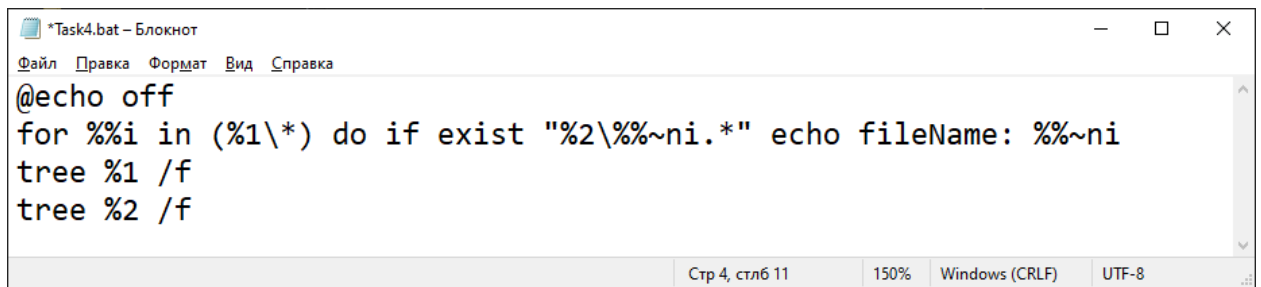


```
PS D:\GH\university\term6\OS\lab2> .\Task3.bat
ENTER PATH: ..
count=10
Структура папок тома Local D
Серийный номер тома: 000000D3 2D7E:8820
D:\GH\UNIVERSITY\TERM6\OS
├──lab1
├──lab2
│   ├──task6
│   │   ├──cmd
│   │   ├──doc
│   │   ├──Files
│   │   ├──other
│   │   └──txt
│   ├──task7
│   └──subCatalog
```

Рисунок 2.6 – Выполнение третьей программы

4. В каталогах, имена которых заданы первым и вторым параметрами командного файла, найти и вывести на экран имена файлов (расширения могут быть любые), присутствующие как в первом, так и во втором каталоге.

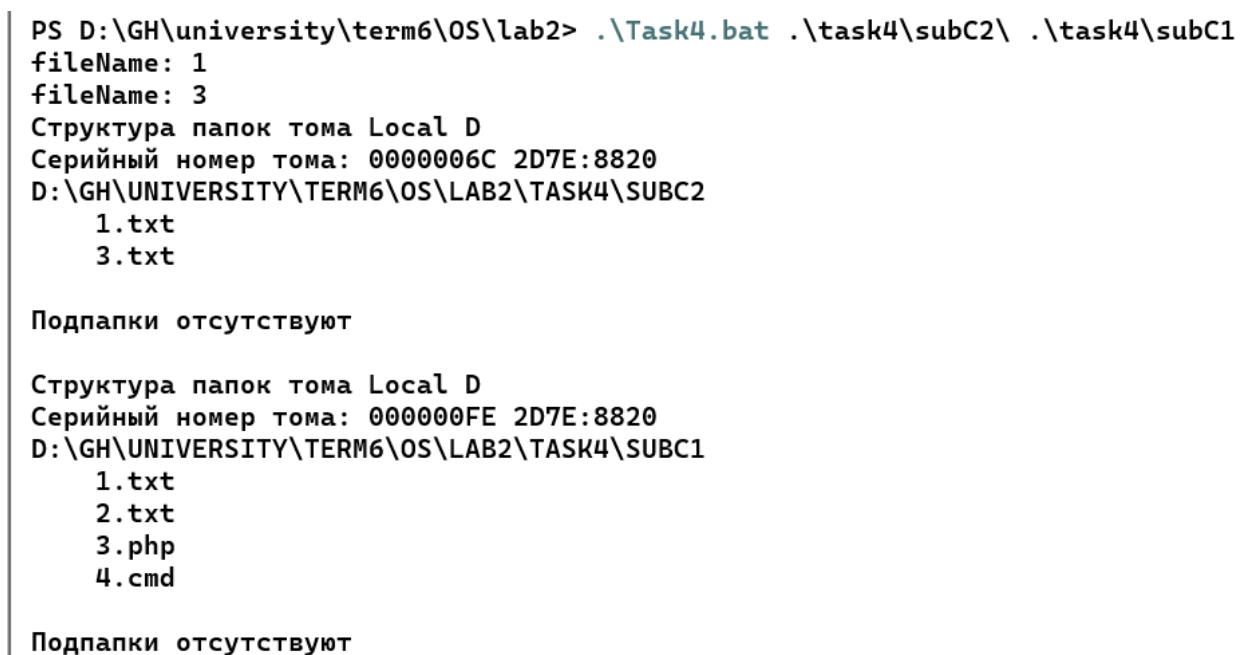
Следует использовать только один оператор FOR. Листинг программы, выполняющей данное задание, представлен на рисунке 2.7.



```
*Task4.bat – Блокнот
Файл  Правка  Формат  Вид  Справка
@echo off
for %%i in (%1\*) do if exist "%2\%%~ni.*" echo fileName: %%~ni
tree %1 /f
tree %2 /f
Стр 4, столб 11    150%    Windows (CRLF)    UTF-8
```

Рисунок 2.7 – Листинг четвёртой задачи

Результат выполнения этого кода изображен на рисунке 2.8.



```
PS D:\GH\university\term6\OS\lab2> .\Task4.bat .\task4\subC2\ .\task4\subC1
fileName: 1
fileName: 3
Структура папок тома Local D
Серийный номер тома: 0000006C 2D7E:8820
D:\GH\UNIVERSITY\TERM6\OS\LAB2\TASK4\SUBC2
    1.txt
    3.txt

Подпапки отсутствуют

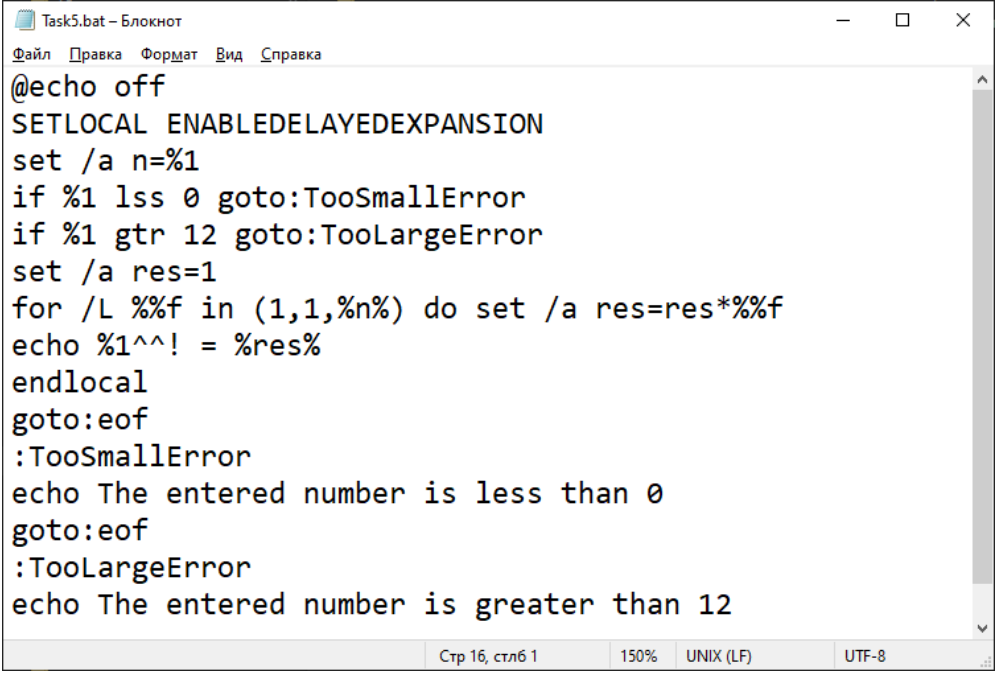
Структура папок тома Local D
Серийный номер тома: 000000FE 2D7E:8820
D:\GH\UNIVERSITY\TERM6\OS\LAB2\TASK4\SUBC1
    1.txt
    2.txt
    3.php
    4.cmd

Подпапки отсутствуют
```

Рисунок 2.8 – Выполнение четвёртой программы

5. Вычисление и вывод на экран значения факториала целого числа, задаваемого при запуске КФ. Предусмотреть проверку заданного значения и при задании отрицательного значения или значения, превышающего максимально возможную величину равную тридцати, выводить соответствующие сообщения. Для проверки правильности вычислений использовать калькулятор.

Листинг программы, выполняющей данное задание, представлен на рисунке 2.9:



```
Task5.bat - Блокнот
Файл  Правка  Формат  Вид  Справка
@echo off
SETLOCAL ENABLEDELAYEDEXPANSION
set /a n=%1
if %1 lss 0 goto:TooSmallError
if %1 gtr 12 goto:TooLargeError
set /a res=1
for /L %%f in (1,1,%n%) do set /a res=res*%%f
echo %1^^! = %res%
endlocal
goto:eof
:TooSmallError
echo The entered number is less than 0
goto:eof
:TooLargeError
echo The entered number is greater than 12
Стр 16, столб 1    150%    UNIX (LF)    UTF-8
```

Рисунок 2.9 – Листинг пятой задачи

```
PS D:\GH\university\term6\OS\lab2> .\Task5.bat -3
The entered number is less than 0
PS D:\GH\university\term6\OS\lab2> .\Task5.bat 0
0! = 1
PS D:\GH\university\term6\OS\lab2> .\Task5.bat 1
1! = 1
PS D:\GH\university\term6\OS\lab2> .\Task5.bat 6
6! = 720
PS D:\GH\university\term6\OS\lab2> .\Task5.bat 13
The entered number is greater than 12
PS D:\GH\university\term6\OS\lab2> |
```

Рисунок 2.10 – Выполнение пятой программы

6. Просмотр содержимого каталога, указанного первым параметром КФ. Необходимо: 1. создать подкаталоги с именами EXE, TXT, CMD, DOC и

OTHER. 2. В каждый подкаталог скопировать файлы с соответствующими расширениями. 3. Пустые подкаталоги удалить.

Листинг программы, выполняющей задание 6, представлен на рисунке 2.11.



```
task6.bat – Блокнот
Файл  Правка  Формат  Вид  Справка
@echo off

md "%~1\exe" "%~1\doc" "%~1\cmd" "%~1\txt" "%~1\other"

for %%f in (%~1\Files\*.*) do (
    if "%%~xf"==".exe" (
        copy %%f "%~1\exe"
    ) else if "%%~xf"==".doc" (
        copy %%f "%~1\doc"
    ) else if "%%~xf"==".cmd" (
        copy %%f "%~1\cmd"
    ) else if "%%~xf"==".txt" (
        copy %%f "%~1\txt"
    ) else copy %%f "%~1\other"
)

FOR /F delims^= %%A IN ('DIR/AD/B/S^|SORT/R') DO RD "%%A"
cls
dir %1
```

Рисунок 2.11 – Код для выполнения 6 задания

Результат запуска КФ с параметром task6 изображен на рисунке 2.12. В папке-источника файлов намеренно не было файлов с расширением .exe, поэтому папки с именем exe не должно быть в результате выполнения КФ.

```

Том в устройстве D имеет метку Local D
Серийный номер тома: 2D7E-8820

Содержимое папки D:\GH\university\term6\OS\lab2\task6

12.03.2023  23:45    <DIR>        .
12.03.2023  23:45    <DIR>        ..
12.03.2023  23:37    <DIR>        cmd
12.03.2023  23:37    <DIR>        doc
12.03.2023  22:10    <DIR>        Files
12.03.2023  23:37    <DIR>        other
12.03.2023  23:37    <DIR>        txt
              0 файлов              0 байт
              7 папок  164 296 847 360 байт свободно

D:\GH\university\term6\OS\lab2>task6.bat task6_

```

Рисунок 2.12 – Результат запуска КФ для 6 задания

7. В каталоге, указанном первым параметром КФ, (и его подкаталогах) найти файлы наибольшего и наименьшего размеров. Вывести имена файлов, их размеры и даты создания.

Листинг программы, выполняющей задание 7, представлен на рисунке 2.13.


```

@echo off
cls
set maxsize=0
setlocal enabledelayedexpansion
for /f "tokens=* delims=" %%a in ('dir /a-d/b/s "%~1"') do (
    if %%~za gtr !maxsize! (
        set "name=%%~nxa"
        set "maxsize=%%~za"
        set "create=%%~ta"
    )
)
echo File max size
echo. %name% ^| %maxsize% ^| %create%
echo.
set minsize=%maxsize%
for /f "tokens=* delims=" %%a in ('dir /a-d/b/s "%~1"') do (
    if %%~za lss !minsize! (
        set "name=%%~nxa"
        set "minsize=%%~za"
        set "create=%%~ta"
    )
)
echo File min size
echo. %name% ^| %minsize% ^| %create%
pause>nul
exit|

```

Рисунок 2.13 – Листинг КФ для выполнения задания 7.

Результат запуска КФ с параметром task7 изображен на рисунке 2.14.

```

File max size
test (2).php | 895776 | 19.03.2023 20:24

File min size
request.jpg | 863 | 17.03.2023 19:41

```

Рисунок 2.14 – Результат запуска КФ для 7 задания.

Контрольные вопросы

1. Вывод сообщений и дублирование команд. По умолчанию команды пакетного файла перед исполнением дублируются на экране, что иногда излишне. С помощью команды ECHO OFF можно отключить дублирование команд, идущих после нее. Включить дублирование можно аналогично командой: ECHO ON.

2. Использование параметров командной строки.

При запуске пакетных файлов в командной строке можно указывать произвольное число параметров, значения которых можно использовать внутри файла. Это позволяет, например, применять один и тот же командный файл для выполнения команд с различными параметрами. Для доступа из командного файла к параметрам командной строки применяются символы %0, %1, ..., %9 или %*. При этом вместо %0 подставляется имя выполняемого пакетного файла, вместо %1, %2, ..., %9 – значения первых девяти параметров командной строки соответственно, а вместо %* – все аргументы.

3. Переменные среды, получение и изменение их значений. Внутри командных файлов можно использовать так называемые переменные среды (или переменными окружения), каждая из которых хранится в оперативной памяти, имеет свое уникальное имя, а ее значением является строка. Стандартные переменные среды автоматически инициализируются в процессе загрузки операционной системы. Такими переменными являются:

- WINDIR, которая определяет расположение каталога Windows,
- TEMP, которая определяет путь к каталогу для хранения временных файлов Windows
- PATH, в которой хранится системный путь (путь поиска), то есть список каталогов, в которых система должна искать выполняемые файлы или файлы совместного доступа (например, динамические библиотеки).

Кроме того, в командных файлах с помощью команды SET можно объявлять собственные переменные среды.

4. Операции со строковыми и числовыми переменными.

Со строковыми переменными можно выполнять операции конкатенации:

SET A=Раз

SET B=Два

SET C=%A%%B%

Можно выделить подстроку с помощью следующей конструкции: %имя_переменной:~n1,n2%, где число n1 определяет смещение (количество пропускаемых символов) от начала (если n1 положительно) или от конца (если n1 отрицательно) соответствующей переменной среды, а число n2 – количество выделяемых символов (если n2 положительно) или количество последних символов в переменной, которые не войдут в выделяемую подстроку (если n2 отрицательно). Если указан только один отрицательный параметр -n, то будут извлечены последние n символов.

Можно выполнять процедуру замены подстрок с помощью конструкции:

%имя_переменной:s1=s2% (в результате будет возвращена строка, в которой каждое вхождение подстроки s1 в соответствующей переменной заменено на s2).

При включенной расширенной обработке команд (этот режим в Windows используется по умолчанию) имеется возможность рассматривать значения переменных среды как числа и производить с ними арифметические вычисления (используются только целые числа). Для этого используется команда SET с ключом /A. В команде SET с ключом /A могут использоваться операции - (вычитание), * (умножение), / (деление нацело), % (остаток от деления). При использовании знака % в качестве знака операции в командных файлах он должен быть записан дважды.

5. Проверка существования заданного файла и наличия переменной среды.

Проверка существования заданного файла осуществляется с помощью следующей команды:

IF [NOT] EXIST файл команда1 [ELSE команда2]

Условие считается истинным, если указанный файл существует.

Аналогично файлам команда IF позволяет проверить наличие в системе определенной переменной среды:

IF DEFINED переменная команда1 [ELSE команда2]

Здесь условие DEFINED применяется подобно условию EXISTS, но принимает в качестве аргумента имя переменной среды и возвращает истинное значение, если эта переменная определена.

6. Выполнение заданной команды для всех элементов указанного множества.

Для выполнения какой-либо заданной команды для всех элементов указанного множества используется следующий вид цикла FOR:

FOR %%переменная IN (множество)

DO команда [параметры]

7. Выполнение заданной команды для всех подходящих имен файлов.

Для выполнения какой-либо заданной команды для всех подходящих имен файлов в качестве множества в цикл FOR передаются искомые имена или шаблон.

8. Выполнение заданной команды для всех подходящих имен каталогов.

Для выполнения какой-либо заданной команды для всех подходящих имен каталогов используется цикл FOR с ключом /D:

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

9. Выполнение заданной команды для определенного каталога, а также всех его подкаталогов.

Для выполнения какой-либо заданной команды для определенного каталога, а также всех его подкаталогов используется цикл FOR с ключом /R:

FOR /R [[диск:]путь] %переменная IN (набор)

DO команда [параметры]

10. Получение последовательности чисел с заданными началом, концом и шагом приращения.

Получить последовательность чисел с заданными началом, концом и шагом приращения позволяет цикл FOR с ключом /L:

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

11. Чтение и обработка строк из текстового файла.

Чтение и обработка строк из текстового файла осуществляется с помощью цикла FOR с ключом /F:

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

Здесь параметр набор содержит имена одного или нескольких файлов, которые по очереди открываются, читаются и обрабатываются. Обработка состоит в чтении файла, разбиении его на отдельные строки текста и выделении из каждой строки заданного числа подстрок. Затем найденная подстрока используется в качестве значения переменной при выполнении основного тела цикла (заданной команды).

12. Команда Findstr. Назначение. Ключи. Использование регулярных выражений в команде. Задание и использование класса цифр и класса букв через диапазон.

Назначение команды - поиск строк в текстовых файлах.

FINDSTR [/B] [/E] [/L] [/R] [/S] [/I] [/X] [/V] [/N] [/M] [/O] [/P] [/T:файл] [/C:строка] [/Ю:файл] [/D:список_папок] [/A:цвета] [/OFF[LINE]] строки [[диск:][путь]имя_файла[...]]

/L-Поиск строк дословно.

/R-Поиск строк как регулярных выражений.

/S-Поиск файлов в текущей папке и всех ее подпапках.

/I-Определяет, что поиск будет вестись без учета регистра.

/X-Печатает строки, которые совпадают точно.

/V-Печатает строки, не содержащие совпадений с искомыми.

/N-Печатает номер строки, в которой найдено совпадение, и ее содержимое.

/M-Печатает только имя файла, в которой найдено совпадение.

/O-Печатает найденные строки через пустую строку.

/P-Пропускает строки, содержащие непечатаемые символы.

/T:файл-Читает список файлов из заданного файла

/C:строка-Использует заданную строку как искомую фразу поиска.

Также с этой командой могут использоваться и регулярные выражения:

. – Любой символ.

* – Повтор: ноль или более вхождений предыдущего символа или класса.

^ – Позиция в строке: начало строки.

\$ – Позиция в строке: конец строки.

[класс] – Класс символов: любой единичный символ из множества.

[x-y] – Диапазон: любые символы из указанного диапазона

\x – Служебный символ: символьное обозначение служебного символа

x.

\<xyz – Позиция в слове: в начале слова.

xyz\> – Позиция в слове: в конце слова.

13. Операторы перехода и вызова.

Командный файл может содержать метки и команды GOTO перехода к этим меткам:

GOTO метка

Любая строка, начинающаяся с двоеточия :, воспринимается при обработке командного файла как метка. Имя метки задается набором символов, следующих за двоеточием до первого пробела или конца строки.

Для перехода к метке внутри текущего командного файла кроме команды

GOTO можно использовать и команду CALL:

CALL :метка аргументы

При вызове такой команды создается новый контекст текущего пакетного файла с заданными аргументами, и управление передается 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 строки CALL, а второй выход завершает выполнение пакетного файла.

14. Какое минимальное количество строк (включая @echo off) должен иметь командный файл, выводящий на экран минимальное значения двух числовых аргументов?

Командный файл, выводящий на экран минимальное значения двух числовых аргументов, должен иметь, по крайней мере, 2 строчки (включая @echo off):

```
@ECHO OFF
```

```
if %1 GTR %2 (echo %1 ) else (echo %2)
```

15. Какое минимальное количество строк (включая @echo off) должен иметь командный файл, выводящий на экран минимальное значения трех числовых аргументов?

Командный файл, выводящий на экран минимальное значения двух числовых аргументов, должен иметь, по крайней мере, 2 строчки (включая @echo off):

```
@ECHO OFF
```

```
if %1 GTR %2 (if %1 GTR %3 (echo %1) else (echo %3)) else (if %2 GTR %3 (echo %2) else (echo
```