МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Институт компьютерных технологий и защиты информации

(наименование института (факультета), филиала)

Кафедра Прикладной Математики и Информатики

(наименование кафедры)

09.03.04 «Программная инженерия»

(шифр и наименование направления подготовки (специальности))

Лабораторная работа № 5

По дисциплине: Операционные системы

На тему: "Работа со средой"

Выполнил: студент группы 4310

**Гауиш М.Г**.

Проверил: доц. **Урахчиский И. Н**

Казань, 2023 год

Цель работы

1) Изучение объектов WSH.

2) Изучение приемов программирования с использованием объектов WSH.

3) Получение навыков разработки сценариев.

Содержание работы

1) Изучение основных свойств и методов объектов WshShell, WshShortcut.

2) Выработка навыков использования объектов WshShell, WshShortcut.

3) Изучение и выполнение всех рассмотренных в руководстве приме-ров.

4) Разработка процедуры решения поставленной задачи.

5) Отладка и выполнение процедуры.

Краткие теоретические сведения

Объект WshShell позволяет скриптам запускать и конфигурировать другие приложения. Он также дает возможность общаться с пользователем, изменять Реестри находить папки.

Метод **Run** позволяет исполнять не только приложения Windows, но и ко-манды MS-DOS. Например, можно использовать такую команду на JScript: Shell.Run("Edit.com"); Этот оператор открывает окно редактора MS-DOS. Если добавить к команде имя файла документа, документ будет загружен в редактор. Поскольку путь не указан, Windows ищет файл программы в каталогах, заданных переменной окружения PATH.

Постановка задачи

1. Разработать процедуру вывода справки о командах MS DOS с

возможностью перенаправления справочной информации в файл.

Организовать ввод имени интересуемой команды, пути и имени

результирующего файла в процедуру с помощью параметров командной

строки при вызове процедуры.

Листинг

if(WScript.Arguments.count() == 3){ //команда, адрес и имя файла

var command = WScript.Arguments(0);

var path = WScript.Arguments(1);

var filename = WScript.Arguments(2);

var shell = WScript.CreateObject("WScript.Shell");

var strCommand = "%COMSPEC% /C help " + command + '>' + path +

filename;

WScript.Echo(strCommand); // для отладки

shell.Run(strCommand);

} else if(WScript.Arguments.count() == 1){ //(только команда)

var command = WScript.Arguments(0);

var shell = WScript.CreateObject("WScript.Shell");

shell.Run("%COMSPEC% /K help " + command);

} else { //только lab5.1.js)

var shell = WScript.CreateObject("WScript.Shell");

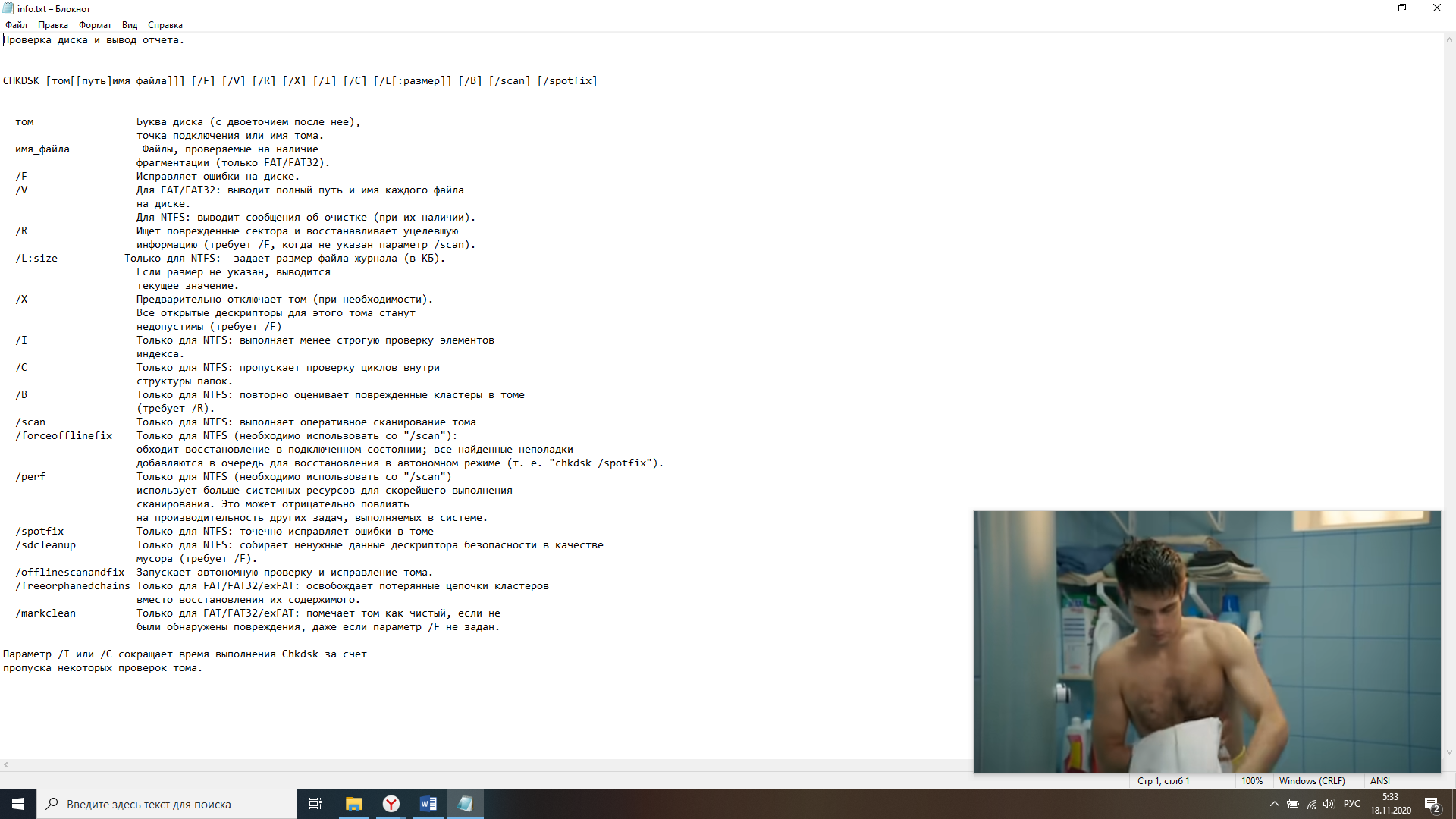
shell.Run("%COMSPEC% /K help ");

}

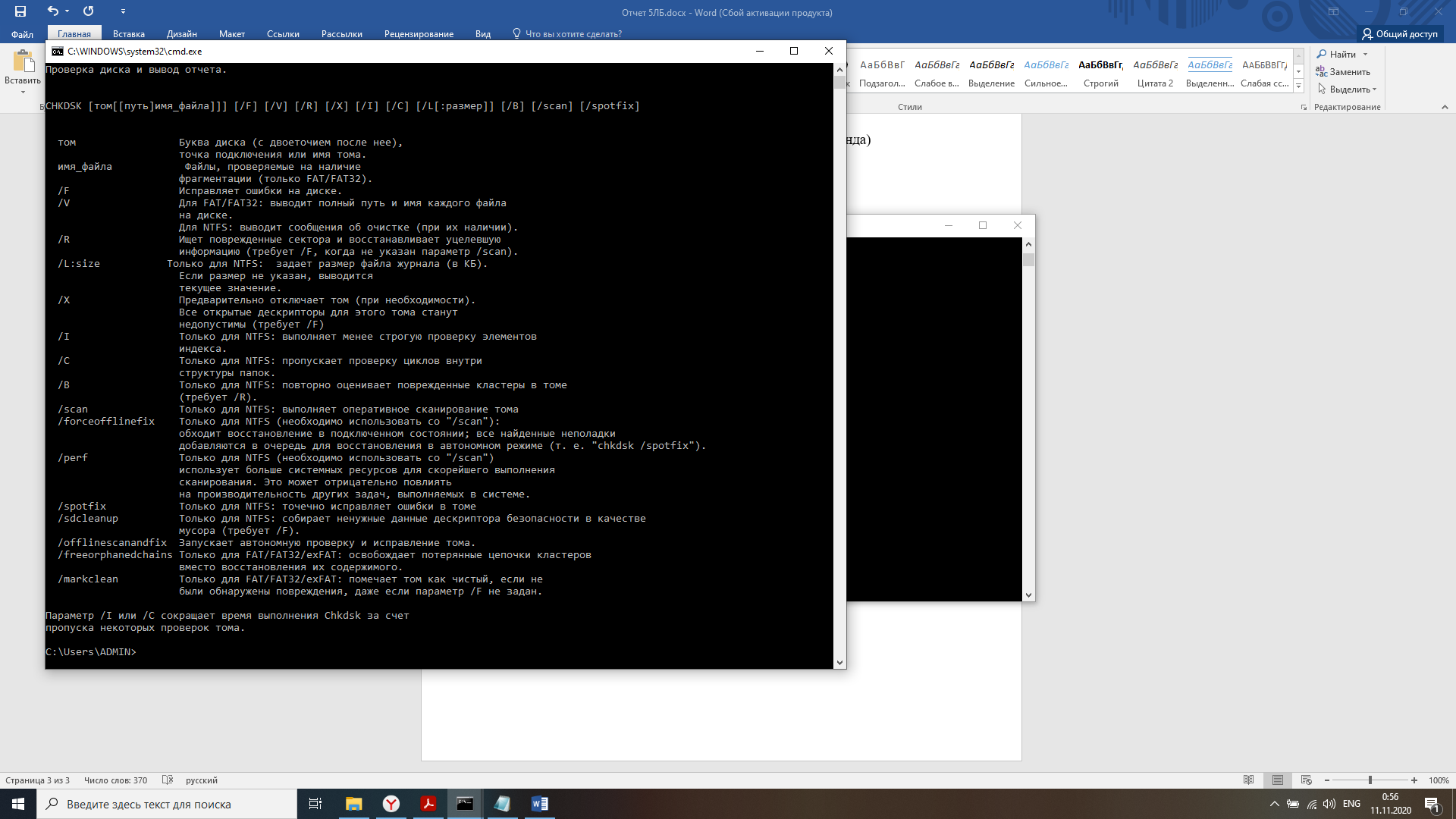
Результат работы программы

При вводе: "cscript 5LB.js CHKDSK C:\Users\ADMIN\info.txt”

CHKDSK - Проверка диска и вывод статистики.

Создается файл info.txt с содержимым:

При вводе "cscript 5LB.js CHKDSK" эта же информация выводится в консоли

При вводе cscript 5LB.js в консоль выводится список всех возможных команд

2. Разработать процедуру запуска Блокнота для создания новой или

редактирования существующей процедуры с возможностью ее последующего запуска в оконном режиме или режиме командной строки. Организовать ввод пути и имени редактируемой процедуры и режим ее запуска с помощью параметров командной строки.

Листинг

var address = WScript.Arguments(0) + '\\';

var name\_p = WScript.Arguments(1);

var mode = WScript.Arguments(2); // cmd or window

var Shell = WScript.CreateObject("WScript.Shell");

Shell.Run(("notepad.exe '" + address + name\_p + "'"), 1, true);

if (mode == "cmd")

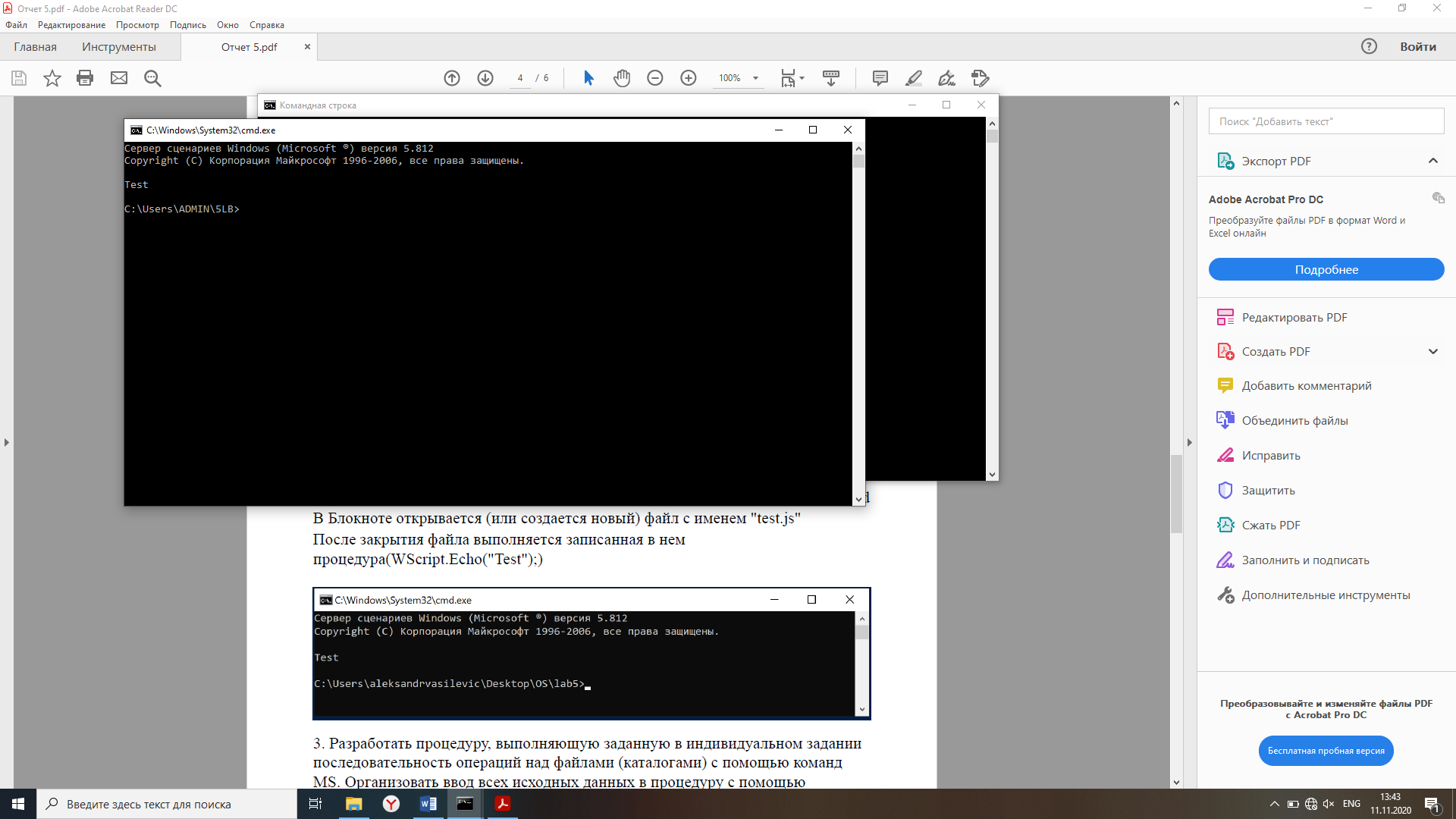
Shell.Run('cmd.exe /K cscript "' + address + name\_p + '"');

else if (mode == "window")

Shell.Run('cmd.exe /K wscript "' + address + name\_p + '" & exit');

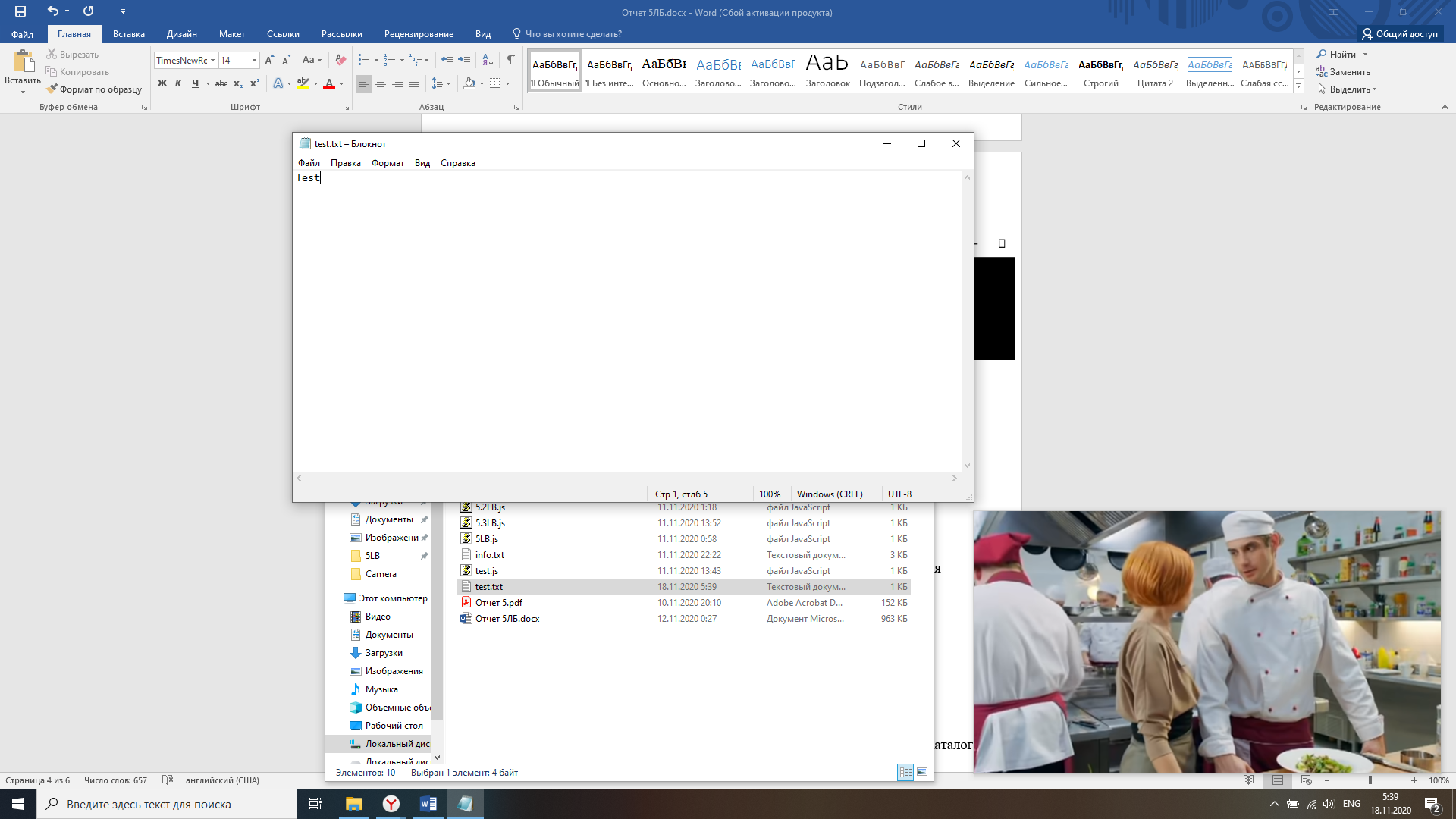
else

WScript.Echo("Ошибка выбора режима запуска процедуры");

Результат работы программы:

При вводе cscript 5.2LB.js C:\Users\ADMIN\5.2LB.js\ test.js cmd

В Блокноте открывается (или создается новый) файл с именем "test.js"

После закрытия файла выполняется записанная в нем процедура(WScript.Echo("Test");)

3. Разработать процедуру, выполняющую заданную в индивидуальном задании последовательность операций над файлами (каталогами) с помощью команд MS. Организовать ввод всех исходных данных в процедуру с помощью параметров командной строки при вызове процедуры. Предусмотреть возможность перенаправления результатов выполнения процедуры в файл.

Копирование дерева каталогов:

1. Создать исходное дерево каталогов;

2.С помощью Блокнота создать исходный файл;

3.Скопировать исходное дерево каталогов в результирующий каталог;

4.Изменить атрибуты файла в результирующем каталоге;

5.Вывести оглавление исходного и результирующего каталогов;

6.Вывести содержимое файла результирующего каталога;

Листинг программы:

var shell = WScript.CreateObject("WScript.Shell");

var oldpath = pathofwork = shell.CurrentDirectory;

var folderpath = "";

var how;

var cmd = "%COMSPEC% /K ";

var pathotchetos = shell.CurrentDirectory; // Путь к файлу(отчету)

var otchetos = "otchet.txt"; // Имя файла(отчета)

switch (WScript.Arguments.Count())

{

case 1:

{

var SCN = "SomethingNewFolder";

// Source Catalog Name Имя исходного каталога

var SFN = "SomethingNewFile.txt";

// Source File Name Имя исходного файла

break;

}

case 3:

{

var pathotchetos = WScript.Arguments(1); // Путь к файлу(отчету)

var otchetos = WScript.Arguments(2); // Имя файла(отчета)

var SCN = "SomethingNewFolder";

// Source Catalog Name Имя исходного каталога

var SFN = "SomethingNewFile.txt";

// Source File Name Имя исходного файла

break;

}

case 5:

{

var pathotchetos = WScript.Arguments(1); // Путь к файлу(отчету)

var otchetos = WScript.Arguments(2); // Имя файла(отчета)

var SCN = WScript.Arguments(3);

var SFN = WScript.Arguments(4);

break;

}

default:

WScript.Quit(WScript.Echo("\nОшибка\nВведите правильно агрументы\n" +

"Используйте: CScript " + WScript.ScriptFullName + " 'Количество уровней у дерева' \n или " +

"CScript " + WScript.ScriptFullName + " 'Количество уровней у дерева' 'Путь к файлу(отчету)' 'Имя файла(отчета)' \n или " +

"CScript " + WScript.ScriptFullName + " 'Количество уровней у дерева' 'Путь к файлу(отчету)' 'Имя файла(отчета)' " +

" 'Имя исходных каталогов' 'Имя исходного файла' "));

}

for ( i = 0; i < WScript.Arguments(0); i++)

{

shell.Run( ( cmd + " md " + pathofwork + "\\" + SCN + " & exit" ), 1, true);

WScript.Echo( "\n" + cmd + " md " + pathofwork + "\\" + SCN + " & exit" );

pathofwork += "\\" + SCN;

folderpath += "\\" + SCN;

}

WScript.Echo(pathofwork + "\n");

WScript.Echo(folderpath);

shell.Run( "notepad.exe '" + pathofwork + "\\" + SFN + "'", 1, true);

WScript.Echo( "\n" + "notepad.exe '" + pathofwork + "\\" + SFN + "'" );

shell.Run( ( cmd + " md " + oldpath + "\\result\\" + SCN + " & exit"), 1, true);

WScript.Echo( "\n" + cmd + " md " + oldpath + "\\result\\" + SCN + " & exit");

shell.Run( ( cmd + "xcopy " + oldpath + "\\" + SCN + " " + oldpath + "\\result\\" + SCN + " /S" + " & exit" ), 1, true);

WScript.Echo( "\n" + cmd + "xcopy " + oldpath + "\\" + SCN + " " + oldpath + "\\result\\" + SCN + " /S" + " & exit" );

shell.Run( ( cmd + "cd " + oldpath + "\\result" + folderpath + " & attrib +R " + SFN ), 1, true);

WScript.Echo( "\n" + cmd + "cd " + oldpath + "\\result" + folderpath + " & attrib +R " + SFN );

// Считывание с консоли

WScript.Echo("Куда вывести информацию? (1 - в командную строку | 2 - в файл | 3 - в стороку и файл");

how = WScript.StdIn.Readline();

// Исполнить команду.

switch (how)

{

case "1":

{

shell.Run(cmd + "cd " + oldpath + " & dir /s ", 1, true);

WScript.Echo( "\n" + cmd + "cd " + oldpath + " & dir /s ");

break;

}

case "2":

{

shell.Run(cmd + "chcp 1251 & cd " + oldpath + " & dir /s /b > " + pathotchetos + "\\" + otchetos, 1, true);

WScript.Echo( "\n" + cmd + "chcp 1251 & cd " + oldpath + " & dir /s /b > " + pathotchetos + "\\" + otchetos);

break;

}

case "3":

{

shell.Run(cmd + "chcp 1251 & cd " + oldpath + " & dir /s /b > " + pathotchetos + "\\" + otchetos + " & dir /s ", 1, true);

WScript.Echo( "\n" + cmd + "chcp 1251 & cd " + oldpath + " & dir /s /b > " + pathotchetos + "\\" + otchetos + " & dir /s ");

break;

}

}

shell.Run( cmd + " chcp 1251 & type " + oldpath + folderpath + "\\" + SFN , 1, true);

WScript.Echo( "\n" + cmd + " chcp 1251 & type " + oldpath + folderpath + "\\" + SFN);

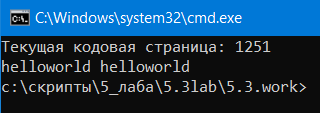
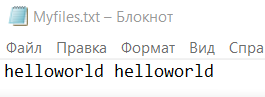
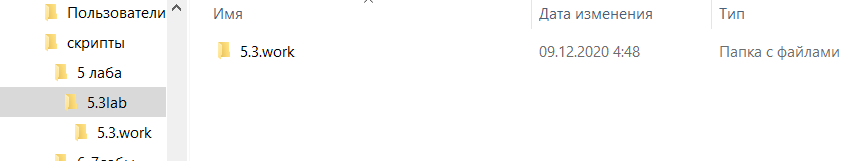
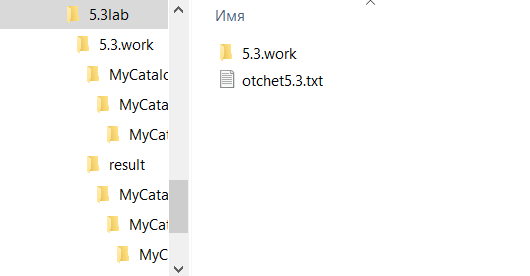
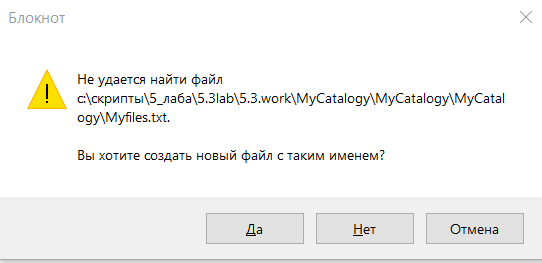
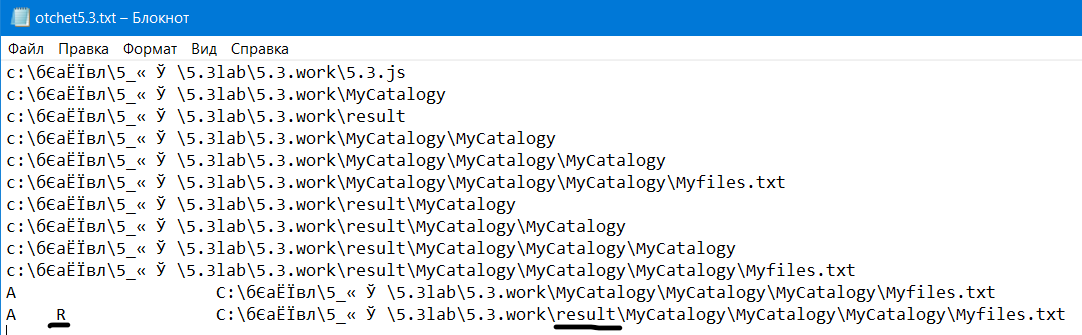
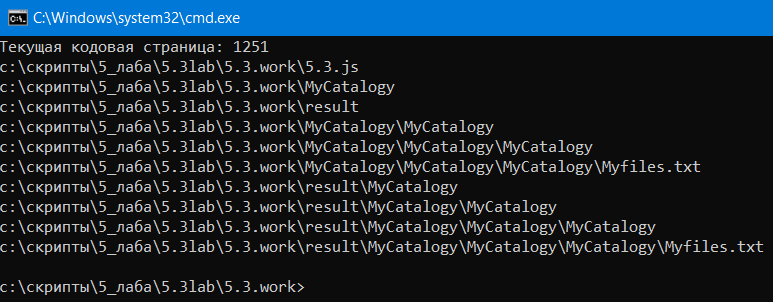
shell.Run( cmd + " chcp 1251 & attrib " + oldpath + folderpath + "\\" + SFN + " >> " + pathotchetos + "\\" + otchetos, 1, true);

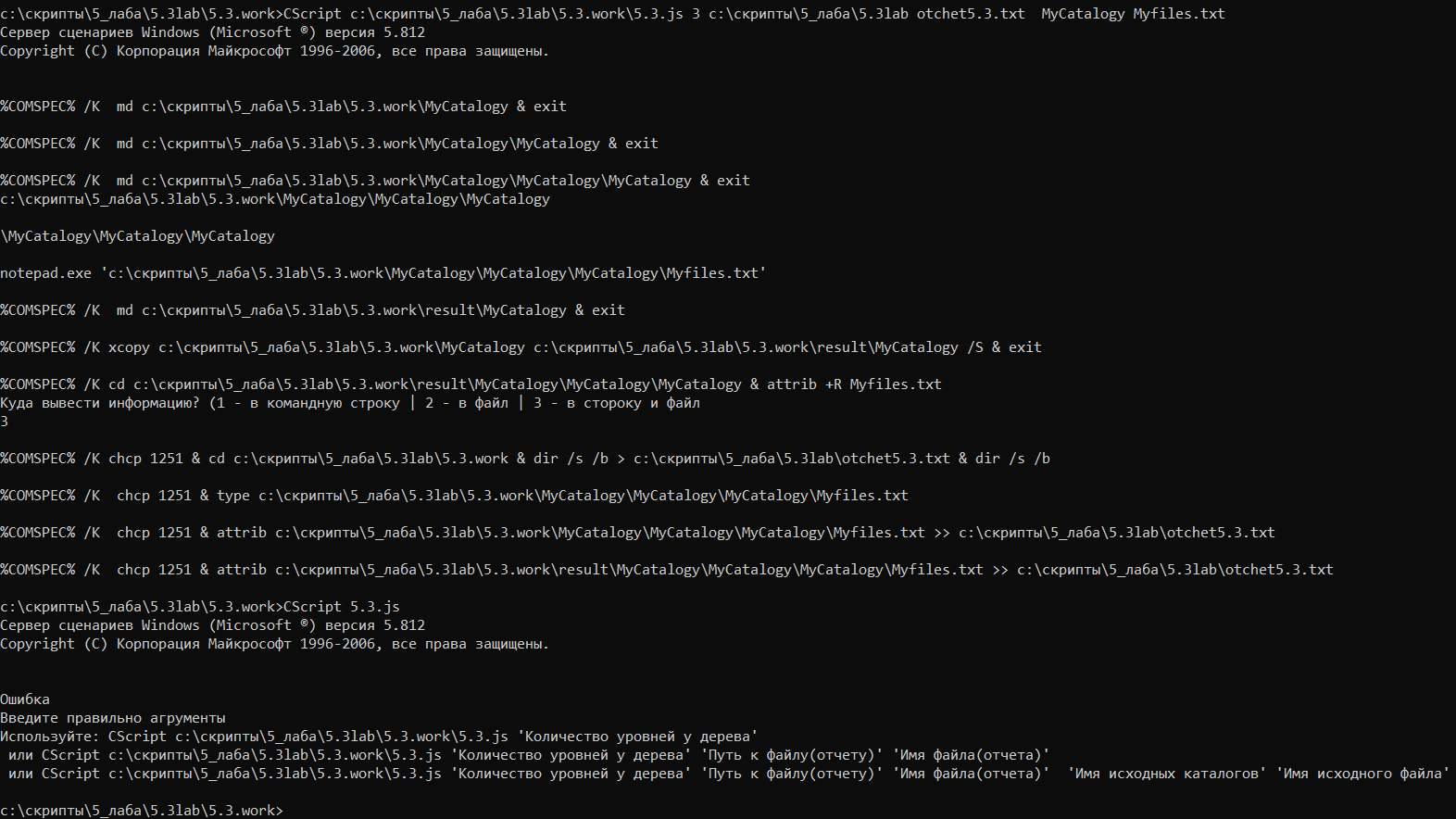
shell.Run( cmd + " chcp 1251 & attrib " + oldpath + "\\result" + folderpath + "\\" + SFN + " >> " +pathotchetos + "\\" + otchetos, 1, true);

WScript.Echo( "\n" + cmd + " chcp 1251 & attrib " + oldpath + folderpath + "\\" + SFN + " >> " + pathotchetos + "\\" + otchetos);

WScript.Echo( "\n" + cmd + " chcp 1251 & attrib " + oldpath + "\\result" + folderpath + "\\" + SFN + " >> " +pathotchetos + "\\" + otchetos);

Результаты:





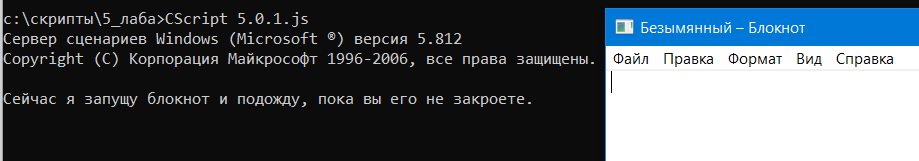
Пример № 1:

// wshExec.js – запуск процесса и ожидание его завершения

var of;

var shell=WScript.CreateObject("WScript.Shell");

WScript.Echo("Сейчас я запущу блокнот и подожду, пока вы его не закроете.");

shell.Run(shell.ExpandEnvironmentStrings("%windir%\\notepad.exe"),1, true);

Пример № 2:

// wshExec.js – запуск процесса и ожидание его завершения

var of;

var shell=WScript.CreateObject("WScript.Shell");

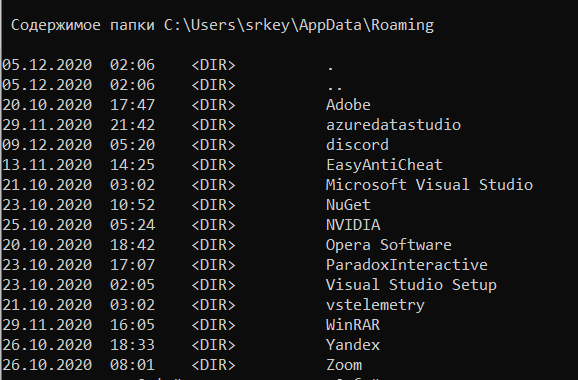
//WScript.Echo("Сейчас я запущу блокнот и подожду, пока вы его не закроете.");

//shell.Run("%windir%\\notepad.exe",1,true);

//shell.Run("cmd.exe /K Dir C:\Windows");

//shell.Run("cmd.exe /K Dir %windir%",1,true);

//shell.Run(shell.ExpandEnvironmentStrings("cmd.exe /K Dir %windir%"),1,true);

shell.Run(shell.ExpandEnvironmentStrings("%COMSPEC% /K Dir %appdata%"),1,true);

Пример № 3:

// Исполнение команды MS-DOS с помощью метода Run

var command, dos\_command, option;

var Shell = WScript.CreateObject("WScript.Shell");

// Команда вызова командного процессора

command = "%COMSPEC% /K ";

// Команда получения справки о командах MS-DOS

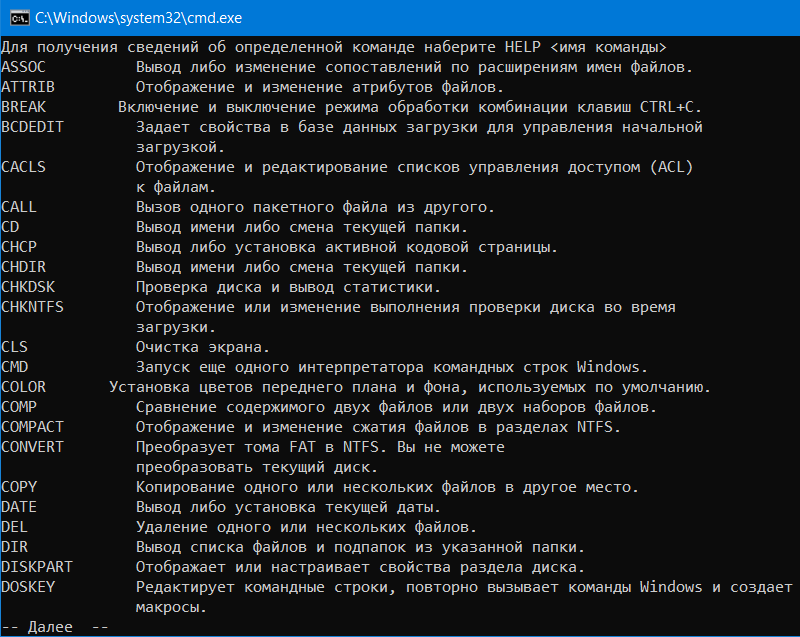
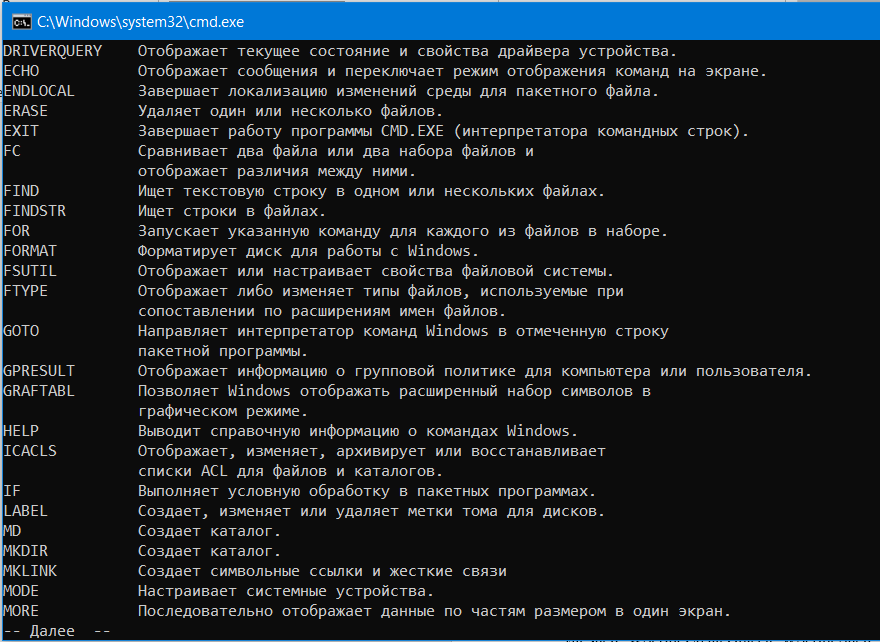
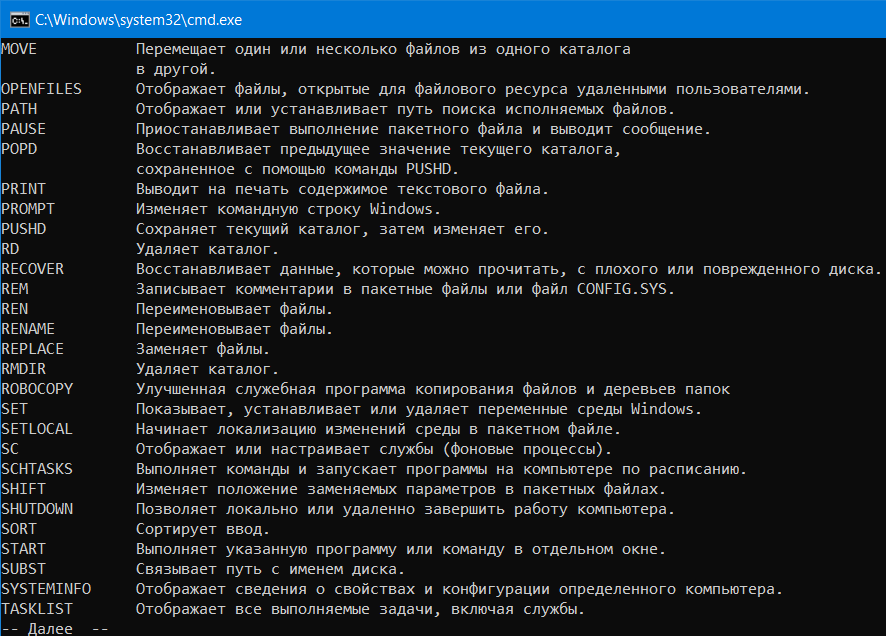
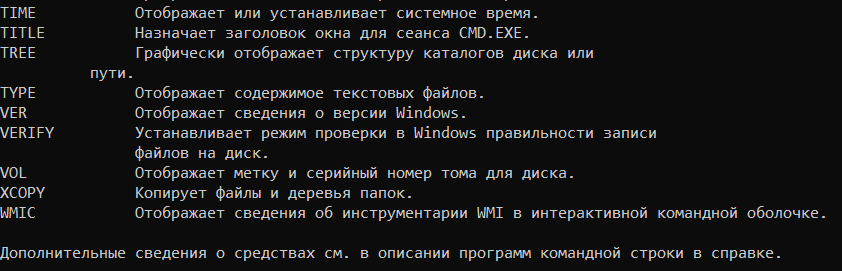
dos\_command = "help ";

// Использование постраничного вывода

option = "| more";

// Исполнить команду.

Shell.Run(command + dos\_command + option);

Пример № 4:

//wshShk.js - создание ярлыка Блокнота на Рабочем столе

var WSHShell=WScript.CreateObject("WScript.Shell");

//путь к Рабочему столу

var DesktopPath = WSHShell.SpecialFolders("Desktop");

//cоздаем ярлык на Рабочем столе

var MyShortcut=WSHShell.CreateShortcut( DesktopPath + "\\Ярлык на блокнот.lnk");

//Задаем свойства объекта-ярлыка и сохраняем их

MyShortcut.TargetPath=WSHShell.ExpandEnvironmentStrings("%windir%\\notepad.exe");

MyShortcut.WorkingDirectory=WSHShell.ExpandEnvironmentStrings("%windir%");

MyShortcut.WindowStyle=4;

MyShortcut.IconLocation=WSHShell.ExpandEnvironmentStrings("%windir%\\notepad.exe,0");

MyShortcut.Save();

WScript.Echo("Теперь на Рабочем столе есть ярлык для Notepad.");

