Операционные системы

Отчет по лабораторной работе № 3

Выполнил: студент группы 4311

Наумов Д.О.

Цель работы

1. Изучение объектов WSH.
2. Изучение приемов программирования с использованием объектов WSH.
3. Получение навыков разработки сценариев.

Примеры

Пример 1

**Листинг программы:**

// 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 ";

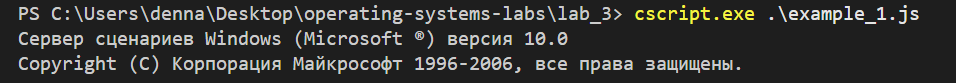
// Использование постраничного вывода

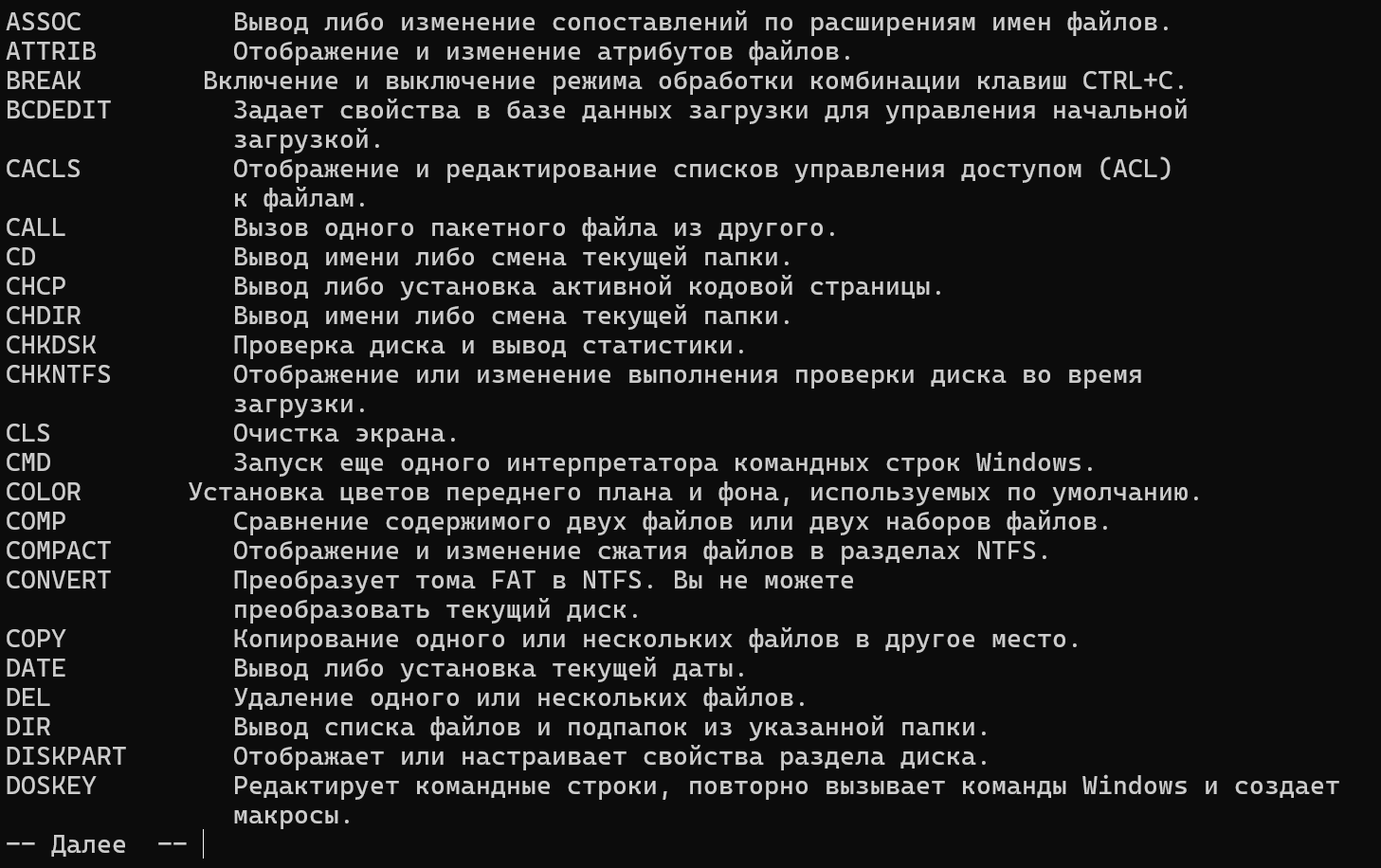
option = "| more"

// Исполнить команду.

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

**Пример работы программы:**

****

****

Пример 2

**Листинг программы:**

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 наблокнот.lnk");

//Задаем свойства объекта-ярлыка и сохраняем их

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

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

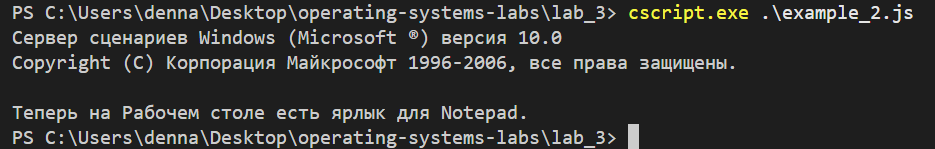
MyShortcut.WindowStyle=4;

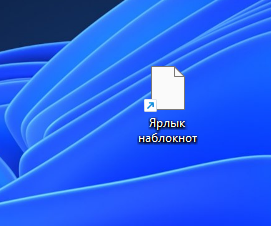
MyShortcut.IconLocation=WSHShell.ExpandEnvironmentStrings("%windir%\\notepaexe,0");

MyShortcut.Save();

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

**Пример работы программы:**

****

****

Пример 3

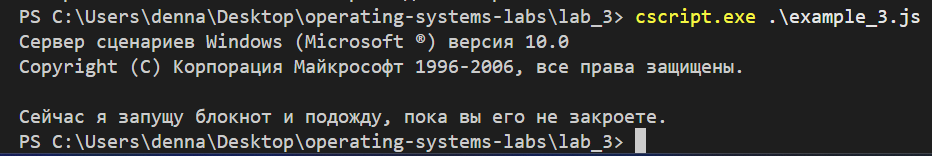
**Листинг программы:**

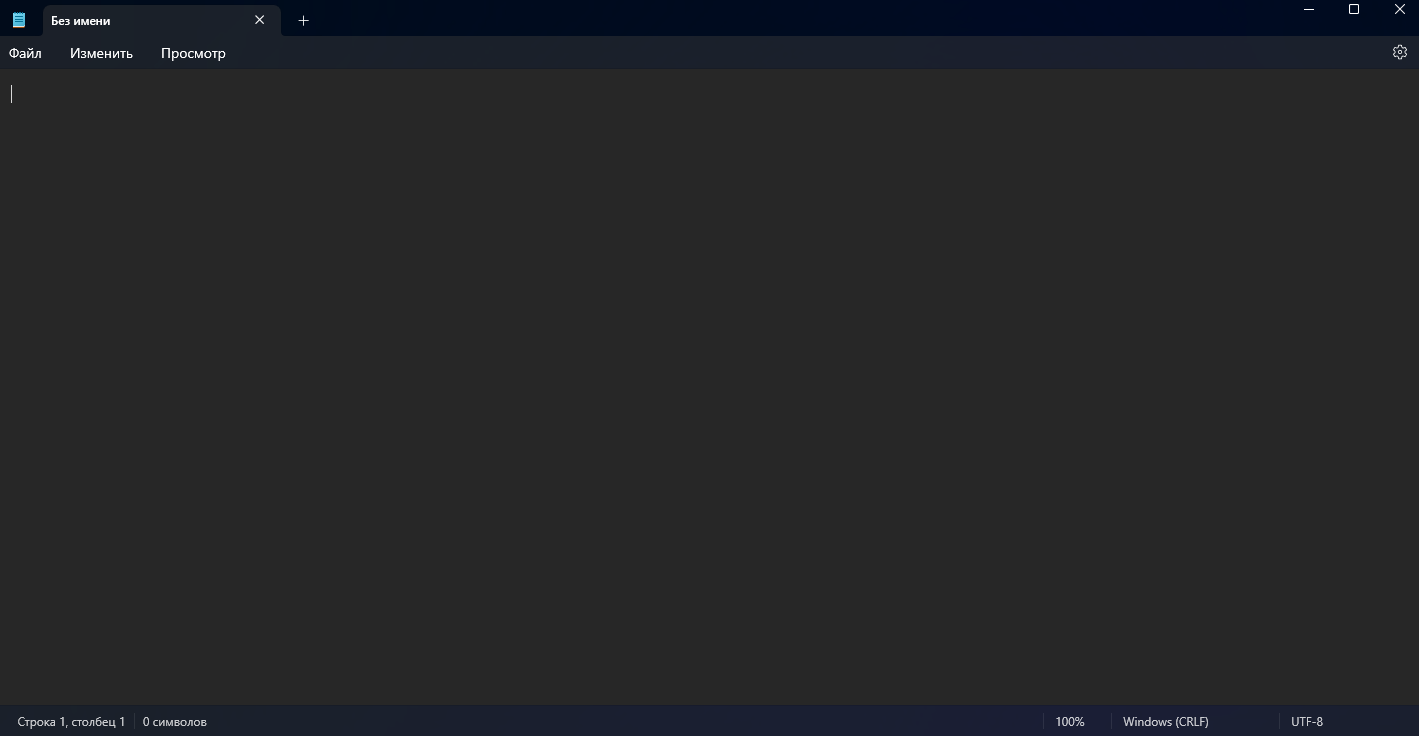
var shell = WScript.CreateObject("WScript.Shell");

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

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

**Пример работы программы:**

****

****

Задания

Задание 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 программы:**

if (WScript.Arguments.count() == 3) {

    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);

    shell.Run(strCommand);

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

    var command = WScript.Arguments(0);

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

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

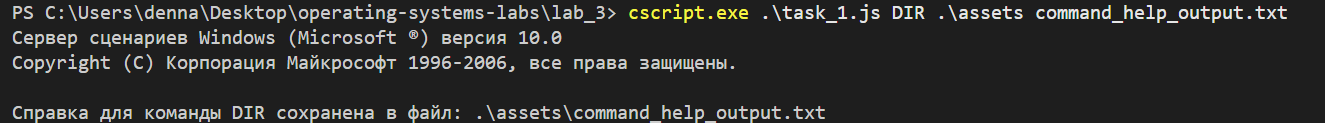
} else {

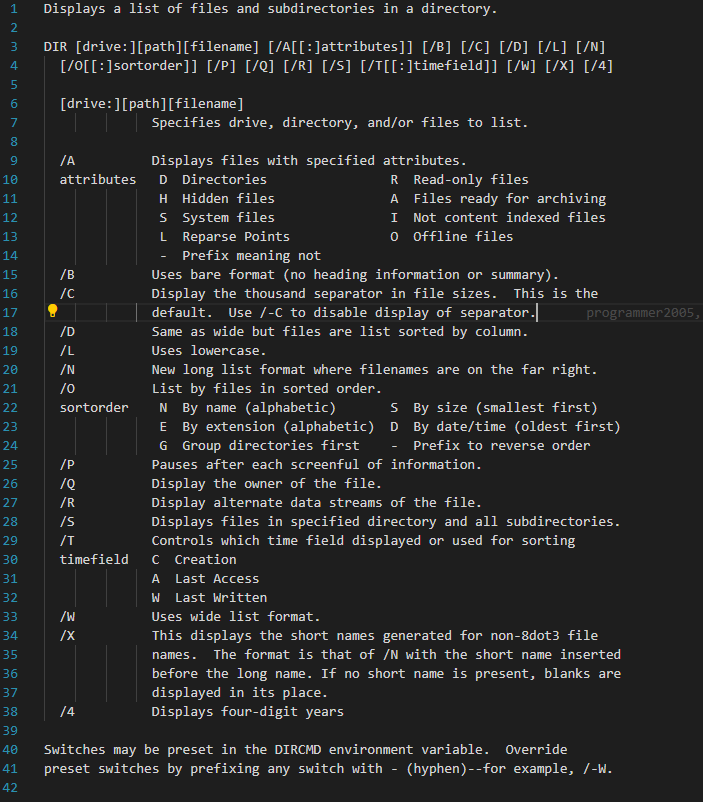
    var shell = WScript.CreateObject("WScript.Shell");

    shell.Run("%COMSPEC% /K help ");

}

**Пример работы программы:**





Задание 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 программы:**

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 + ".txt", 1, true);

if (mode == "cmd")

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

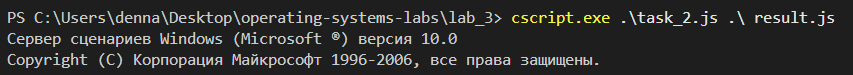
else if (mode == "window")

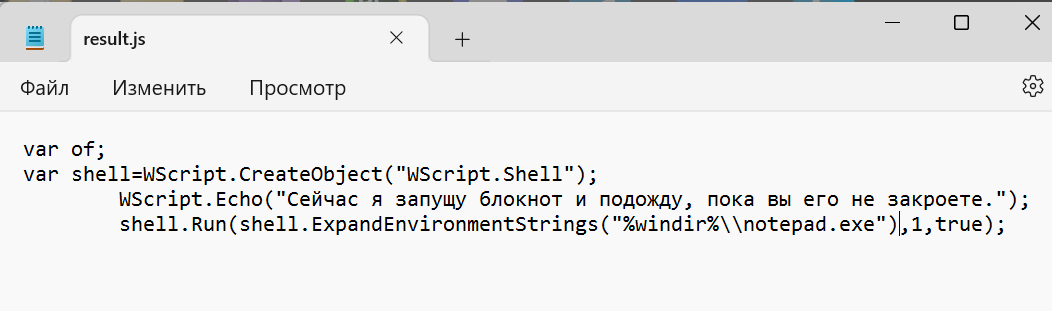
    Shell.Run("cmd.exe /K wscript " + address + name\_p + ".txt & exit");

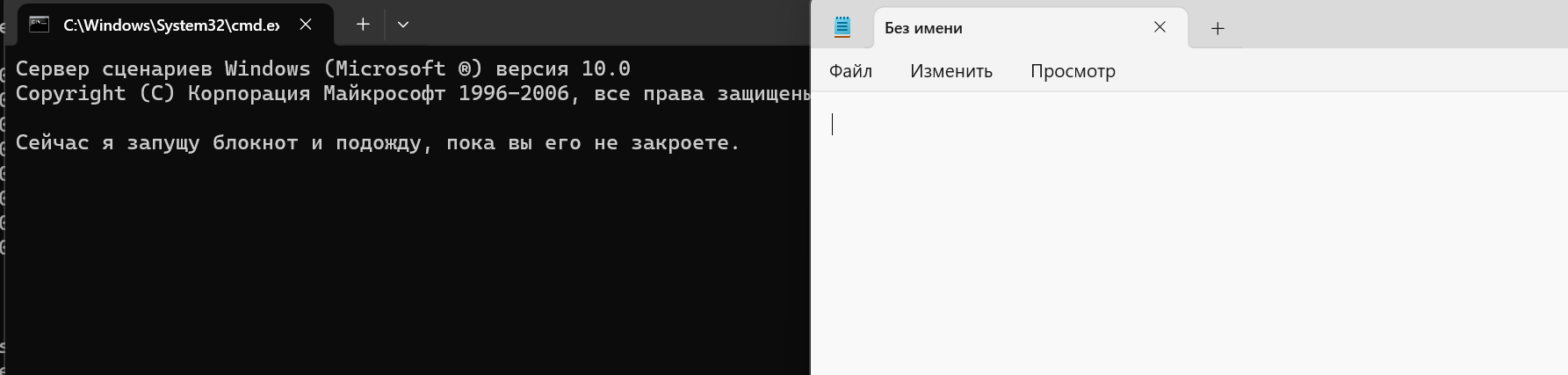
else

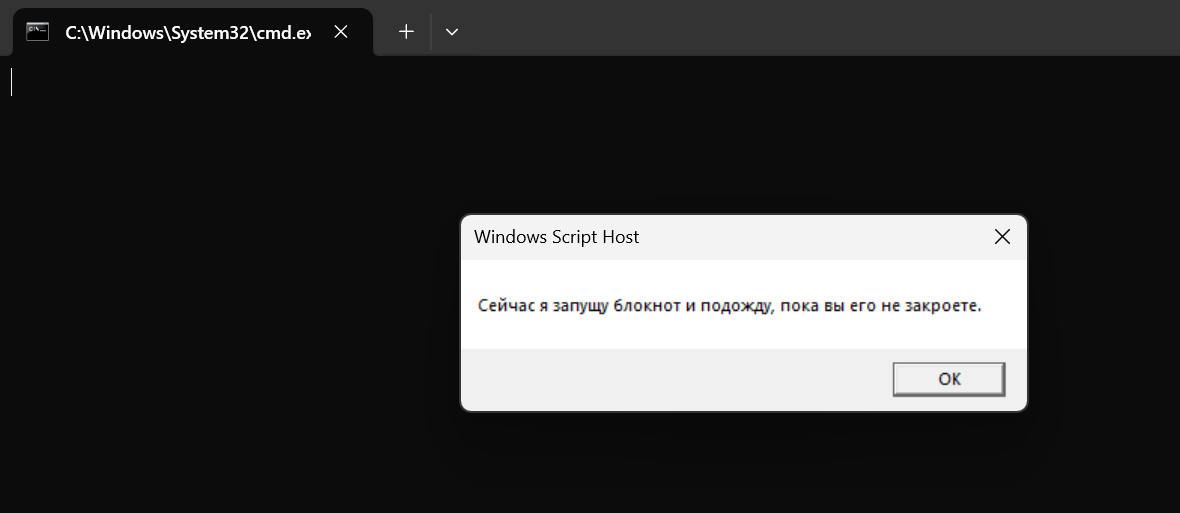
    WScript.Echo("Ошибка выбора режима запуска процедуры!");

**Пример работы программы:**









Задание 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.

3. Переименование каталогов:

3.1. Создать исходный каталог;

3.2. С помощью Блокнота создать исходный файл;

3.3. Переименовать исходный каталог в результирующий каталог;

3.4. Изменить атрибуты файла в результирующем каталоге;

3.5. Вывести оглавление исходного и результирующего каталогов;

3.6. Вывести содержимое файла результирующего каталога;

**Листинг программы:**

// Проверка количества аргументов

if (WScript.Arguments.Length < 3) {

    WScript.Echo("Error: Not enough arguments!");

    WScript.Echo("Usage: cscript " + WScript.ScriptName + " source\_directory file\_name target\_directory [output\_file]");

    WScript.Echo("Example: cscript script.js C:\\source test.txt C:\\result output.log");

    WScript.Echo("Example without redirection: cscript script.js C:\\source test.txt C:\\result");

    WScript.Quit(1);

}

// Получение аргументов командной строки

var fromDir = WScript.Arguments(0);      // исходный каталог

var filename = WScript.Arguments(1);     // имя исходного файла

var toDir = WScript.Arguments(2);        // результирующий каталог

var outputFile = "";                     // файл для перенаправления вывода (опционально)

if (WScript.Arguments.Length >= 4) {

    outputFile = WScript.Arguments(3);

}

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

// 3.1. Создать исходный каталог

WScript.Echo("Creating source directory: " + fromDir);

shell.Run('cmd /c mkdir "' + fromDir + '"', 0, true);

// 3.2. С помощью Блокнота создать исходный файл

WScript.Echo("Creating source file with Notepad...");

shell.Run('cmd /c notepad.exe "' + fromDir + '\\' + filename + '"', 1, true);

// 3.3. Переименовать исходный каталог в результирующий каталог

WScript.Echo("Renaming directory from " + fromDir + " to " + toDir);

shell.Run('cmd /c ren "' + fromDir + '" "' + toDir + '"', 0, true);

// 3.4. Изменить атрибуты файла в результирующем каталоге (устанавливаем атрибут "только для чтения")

WScript.Echo("Setting read-only attribute for file: " + toDir + "\\" + filename);

shell.Run('cmd /c attrib +R "' + toDir + '\\' + filename + '"', 0, true);

// Функция для выполнения команд с перенаправлением вывода

function executeCommand(command, description) {

    if (outputFile != "") {

        shell.Run('cmd /c ' + command + ' >> "' + outputFile + '"', 0, true);

    } else {

        if (description != "") {

            WScript.Echo(description);

        }

        var execObj = shell.Exec('cmd /c ' + command);

        var output = "";

        while (!execObj.StdOut.AtEndOfStream) {

            output += execObj.StdOut.ReadLine() + "\n";

        }

        while (!execObj.StdErr.AtEndOfStream) {

            output += "\n[STDERR]\n" + execObj.StdErr.ReadLine() + "\n";

        }

        WScript.Echo(output);

    }

}

// 3.5. Вывести оглавление результирующего каталога

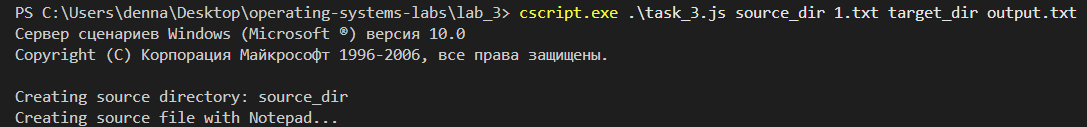
executeCommand('dir "' + toDir + '"', "Contents of target directory:");

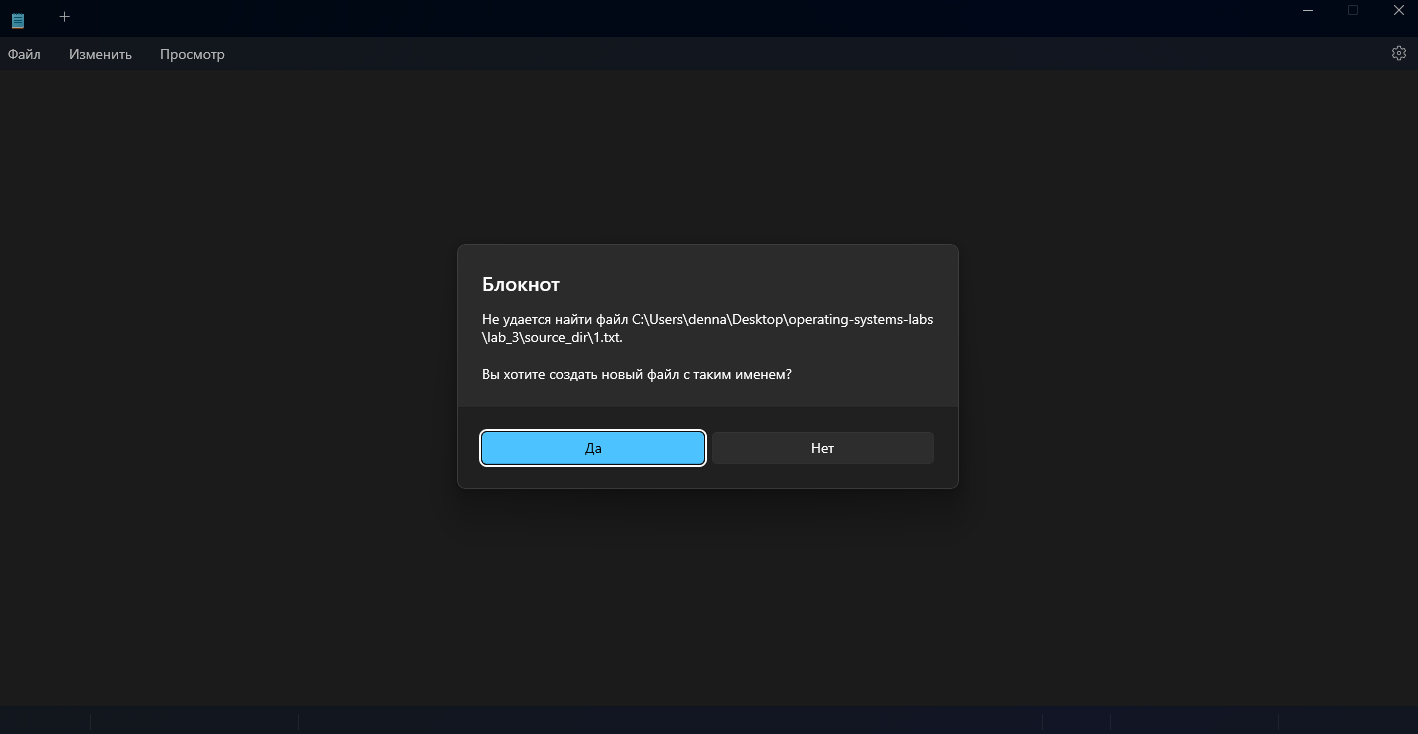
// 3.6. Вывести содержимое файла результирующего каталога

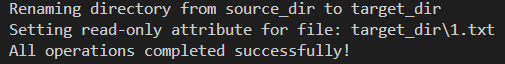
executeCommand('type "' + toDir + '\\' + filename + '"', "Contents of file in target directory:");

WScript.Echo("All operations completed successfully!");

**Пример работы программы:**

****

****

****

