

1. Лабораторная работа № 1

Исследование возможностей командного интерпретатора ОС Windows

1.1. Цель работы

Ознакомиться с возможностями командного интерпретатора ОС Windows. Изучить основные команды интерпретатора ОС Windows. Приобрести практические навыки написания сценариев интерпретатора ОС Windows (bat-файлов).

1.2. Постановка задачи

Изучить способы создания файлов и папок, изменения имени и расширения файлов, копирование и перемещение файлов средствами командного интерпретатора ОС Windows. Написать сценарий оболочки ОС Windows.

1.3. Ход выполнения работы

1.3.1. Текст сценария

```
@echo off
setlocal EnableDelayedExpansion
set log_file=%~N0.log
set input_file_1=%1%
set input_file_2=%2%
set output_file_name=%3%
echo ---=== Start of a new session ===--- 1>>%log_file%
set EXT 1>nul 2>&1
if errorlevel 1 (
    set EXT=.doc 1>nul 2>&1
    echo [WARNING] The EXT variable is not set. Setting the default
value. 1>>%log_file%
)
if not exist "%log_file%" ( echo qwe 1>"%log_file%" )
if not exist "%input_file_1%" (
    echo [ERROR] Invalid first argument.
    echo [ERROR] Invalid first argument. 1>>%log_file%
    goto :EOF
)
if not exist "%input_file_2%" (
    echo [ERROR] Invalid second argument.
    echo [ERROR] Invalid second argument. 1>>%log_file%
    goto :EOF
)
if /I "%output_file_name%" == "" (
    echo [WARNING] Missed third argument. 1>>%log_file%
    :loop1
    set /P answer="Use the name of the first file as the name of the
third [Y/n]? "
    if /I "!answer!" == "y" set output_file_name=%~N1
    if /I "!answer!" == "yes" set output_file_name=%~N1
    if /I "!answer!" == "" set output_file_name=%~N1
    if /I "!answer!" == "n" set output_file_name=%~N2
    if /I "!answer!" == "no" set output_file_name=%~N2
    if /I "!output_file_name!" == "" goto :loop1
    echo * Used filename: !output_file_name!
```

```

    echo * Final file name: !output_file_name!%EXT%
    echo [DEBUG] Used filename: !output_file_name! 1>>%log_file%
    echo [DEBUG] Final file name: !output_file_name!%EXT% 1>>%log_file%
)
echo Merge "%input_file_1%" and "%input_file_2%" and copy to
"%output_file_name%!EXT!".
copy "%input_file_1%" + "%input_file_2%" "%output_file_name%!EXT!" 1>nul
2>&1
if not errorlevel 1 (
    echo * Success
    echo [DEBUG] Successful file creation. 1>>%log_file%
)

if errorlevel 1 (
    echo * Error
    echo [ERROR] Failed file creation. 1>>%log_file%
)
endlocal

```

1.3.2. Выполнение заданий

Выполнение поставленных задач продемонстрировано на Рисунках А.1 – А.10 в Приложении А.

На Рисунке А.1 показано создание директории «FILOZO.IS2» с помощью команды «mkdir». Результат выполнения показан с помощью «dir».

На Рисунке А.2 показан создание трёх директорий с помощью команд «mkdir» и «for..in». Создано три директории. Результат выполнения показан с помощью «dir».

На Рисунке А.3 показан процесс создания текстового файла с заданным содержимым. Выполняется с помощью команды «сору» путём копирования введённых символов из стандартного потока ввода в текстовый файл. Содержимое созданного файла показано с помощью команды «type».

На Рисунке А.4 показано копирование текстового файла из одной директории в другую с помощью команды «сору». Результат выполнения показан с помощью «dir».

На Рисунке А.5 продемонстрировано изменение имени файла с помощью команды «move». Результат выполнения показан с помощью «dir».

Для объединения содержимого двух файлов и сохранения результата объединения в третий файл была использована команда «сору» с использованием ключевого слова «+», что и продемонстрировано на Рисунке А.6. Результат выполнения показан с помощью «dir».

На Рисунке А.7 показано перемещение файла на уровень выше, в родительскую директорию, с помощью команды «move». Результат выполнения показан с помощью «dir».

На Рисунке А.8 показан один из способов вывода содержимого файла – команда «type».

На Рисунке А.9 показан способ удаления файла с помощью команды «del». Результат выполнения показан с помощью «dir».

На Рисунке А.10 показано выполнение сценария командной строки. В первом случае сценарий был запущен без аргументов, вследствие чего сценарий сообщил

об ошибке и аварийно завершил работу, Во втором случае сценарий был запущен с двумя аргументами – путями к первому и второму файлам. В процессе выполнения сценарий потребовал выбрать имя итогового файла. Все сообщения, ошибки и неточности сохранены в log-файле.

Выводы

При выполнении данной лабораторной работы были получены навыки работы с командной строкой Windows, её командами: `copy`, `move`, циклом `for..in` –, перенаправлением потоков вывода и ошибок. Также были получены навыки создания сценариев командной строки. Полученные навыки помогут более эффективно работать с командной строкой, а также создание сценариев позволит автоматизировать некоторые задачи.