

Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 2

По дисциплине «Операционные системы»

На тему «Знакомство с командной оболочкой операционной системы»

Выполнил:

Студент 3 курса 9 группы

Павлович Ян Андреевич

Преподаватель: Савельева М.Г.

Минск, 2025

## **Введение**

**Цель работы:** приобрести практические навыки работы с различными командными оболочками — cmd и bash.

**Постановка задачи:**

Создать каталог OS2, запустить скрипт prep.bat, после чего выполнить ряд команд: cd, сору, хсору, move, replace, ren, fc, del, dir, md, rd.

Необходимо использовать операторы перенаправления вывода (> и >>), конвейеры (|) и фоновое выполнение (&).

Также требуется написать пять bat-файлов (bat\_01 – bat\_05) с использованием переменных, параметров, математических выражений и логических конструкций.

В ходе выполнения работы нужно проанализировать команды, их параметры и переменные окружения (особенно PATH).

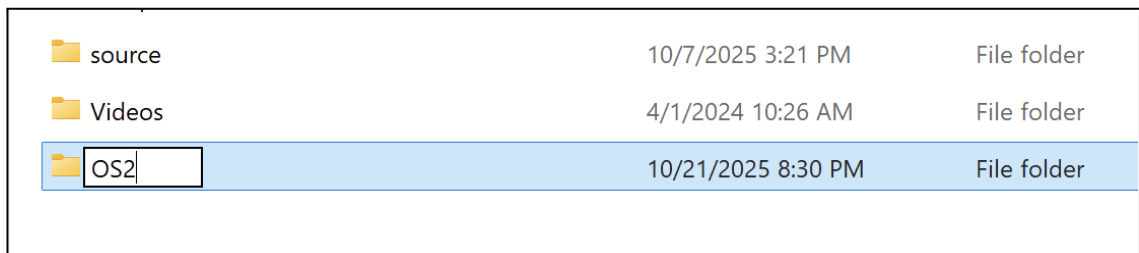
**Используемые инструменты:**

Среда виртуализации: Oracle VirtualBox, Parallels Desktop

Командные оболочки: cmd, bash

## 1 Windows

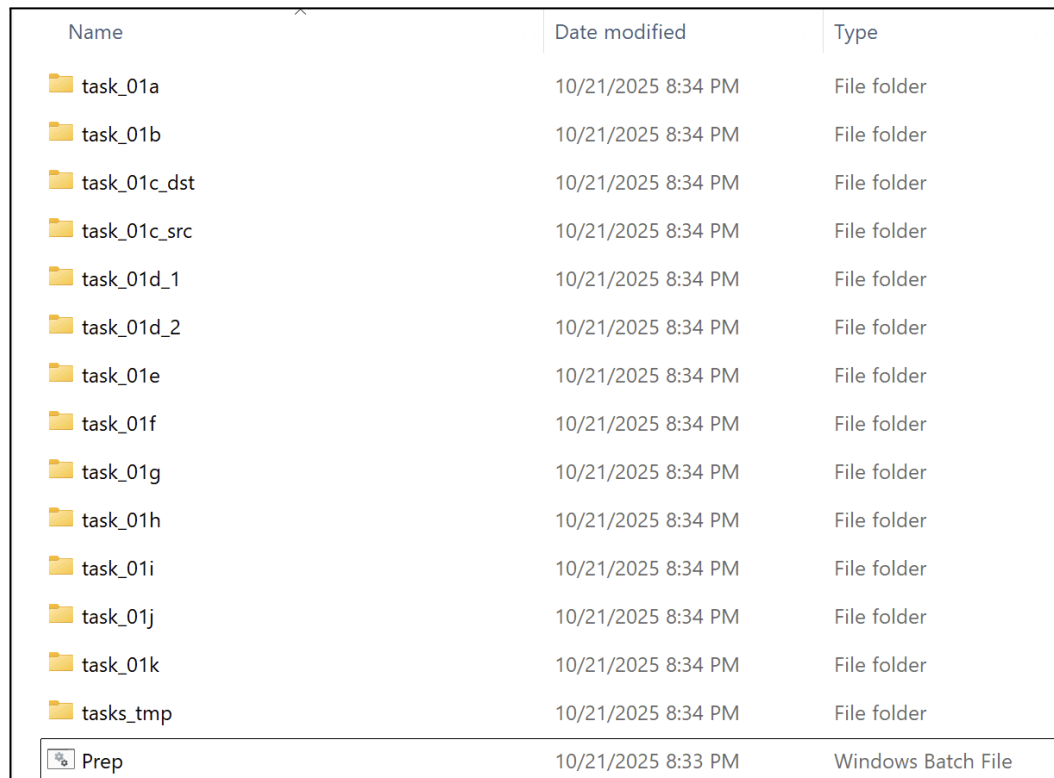
Перейдя в домашний каталог пользователя, созданного в предыдущей лабораторной работе, создаем папку OS2 для дальнейших заданий.



source	10/7/2025 3:21 PM	File folder
Videos	4/1/2024 10:26 AM	File folder
OS2	10/21/2025 8:30 PM	File folder

Рисунок 1.1 – Создание каталога

Скачали и выполнили бат-файл prep.bat, который подготовил нам среду для выполнения заданий лабораторной работы: создал все необходимые каталоги и заполнил их файлами.



Name	Date modified	Type
task_01a	10/21/2025 8:34 PM	File folder
task_01b	10/21/2025 8:34 PM	File folder
task_01c_dst	10/21/2025 8:34 PM	File folder
task_01c_src	10/21/2025 8:34 PM	File folder
task_01d_1	10/21/2025 8:34 PM	File folder
task_01d_2	10/21/2025 8:34 PM	File folder
task_01e	10/21/2025 8:34 PM	File folder
task_01f	10/21/2025 8:34 PM	File folder
task_01g	10/21/2025 8:34 PM	File folder
task_01h	10/21/2025 8:34 PM	File folder
task_01i	10/21/2025 8:34 PM	File folder
task_01j	10/21/2025 8:34 PM	File folder
task_01k	10/21/2025 8:34 PM	File folder
tasks_tmp	10/21/2025 8:34 PM	File folder
Prep	10/21/2025 8:33 PM	Windows Batch File

Рисунок 1.2 – Выполнение бат-файла

Через адресную строку проводника открываем окно командной строки.

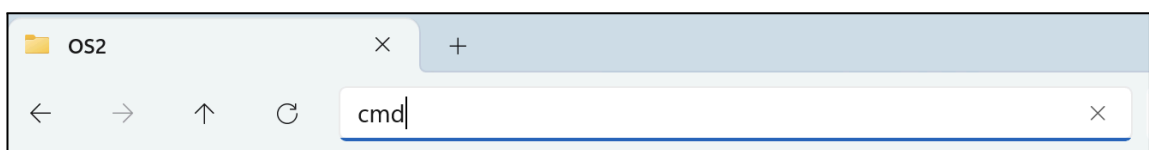


Рисунок 1.3 – Открытие cmd

Далее выполняем переходы по каталогам как с использованием абсолютных, так и относительных путей.

Переходим в каталог tasks\_01i/Русский с использованием абсолютного пути.

```
C:\Users\yanpaulovich\OS2>cd C:\Users\yanpaulovich\OS2\task_01i\Русский
C:\Users\yanpaulovich\OS2\task_01i\Русский>
```

Рисунок 1.4 – Переход в каталог по абсолютному пути

Переходим в каталог tasks\_01i/English с использованием относительного пути.

```
C:\Users\yanpaulovich\OS2\task_01i\Русский>cd ../English
C:\Users\yanpaulovich\OS2\task_01i\English>
```

Рисунок 1.5 – Переход в каталог по относительному пути

Копируем все файлы, имеющие расширение txt из каталога tasks\_tmp в каталог task\_01a.

```
C:\Users\yanpaulovich\OS2>copy tasks_tmp\*.txt task_01a
tasks_tmp\txt1.txt
tasks_tmp\txt2.txt
tasks_tmp\txt3.txt
        3 file(s) copied.
```

Рисунок 1.6 – Копирование файлов

Повторяем ту же операцию, но при этом команда запрашивает подтверждение на перезапись существующих файлов.

```
C:\Users\yanpaulovich\OS2>copy /-Y tasks_tmp\*.txt task_01a
tasks_tmp\txt1.txt
Overwrite task_01a\txt1.txt? (Yes/No/All): y
tasks_tmp\txt2.txt
Overwrite task_01a\txt2.txt? (Yes/No/All): y
tasks_tmp\txt3.txt
Overwrite task_01a\txt3.txt? (Yes/No/All): y
        3 file(s) copied.
```

Рисунок 1.7 – Копирование файлов с запросом на подтверждение

Копируем текстовые файлы из одной директории в другую, пробуем разные варианты копирования: обычное, с запросом подтверждения и с

изменением имени. Копируем файл из каталога task\_01a в каталог task\_01a, но с новым именем.

```
C:\Users\yanpaulovich\OS2>copy task_01a\txt1.txt task_01a\copy_txt1.txt
1 file(s) copied.
```

Рисунок 1.8 – Копирование файла внутри каталога под другим названием

Объединяем несколько текстовых файлов в один, а также сохраняем данные, введенные через консоль, в новый файл. Объединяем файлы txt1.txt и txt2.txt в файл merged.txt.

```
C:\Users\yanpaulovich\OS2\task_01a>copy txt1.txt+txt2.txt merged.txt
txt1.txt
txt2.txt
1 file(s) copied.
```

Рисунок 1.9 – Объединение файлов

Копируем текст, введенный в консоль в файл с именем console.txt.

```
C:\Users\yanpaulovich\OS2>copy con console.txt
Hello
^Z
1 file(s) copied.
```

Рисунок 1.10 – Копирование текста из консоли

Добавляем в конец файла txt3 информацию, вводимую с консоли.

```
C:\Users\yanpaulovich\OS2\task_01a>copy con >> txt3.txt
Hello ^Z
```

Рисунок 1.11 – Добавление информации в конец файла

Далее выполняем рекурсивное копирование каталогов с сохранением системных и скрытых атрибутов, а также с фильтрацией по атрибутам. Копируем все файлы, каталоги и подкаталоги, включая пустые, а также скрытые и системные из task\_01b в task\_01b\_dst\_1.

```

C:\Users\yanpaulovich\OS2>xcopy task_01b task_01b_dst_1 /E /H /K
Does task_01b_dst_1 specify a file name
or directory name on the target
(F = file, D = directory)? d
task_01b\nonarchive-readonly.txt
task_01b\nonarchive.txt
task_01b\readonly.txt
task_01b\system-nonarchive-readonly.txt
task_01b\system-readonly.txt
task_01b\hidden_dir\system1.txt
task_01b\hidden_dir\txt1.txt
task_01b\hidden_dir\subdir\txt1.txt
task_01b\hidden_dir\subdir\txt2.txt
task_01b\subdir\txt1.txt
task_01b\subdir\txt2.txt
11 File(s) copied

```

Рисунок 1.12 – Копирование всех файлов

/E — копирует все подкаталоги, включая пустые. /H — копирует скрытые и системные файлы. /K — сохраняет атрибуты файлов.

Перемещаем файлы, очищаем их содержимое, заменяем одни версии другими. Копируем все файлы с атрибутами «архивный» и «только для чтения» с сохранением этого атрибута для файлов-результатов из task\_01b в task\_01b\_dst\_2.

```

C:\Users\yanpaulovich\OS2>xcopy task_01b task_01b_dst_2 /A /R /K
Does task_01b_dst_2 specify a file name
or directory name on the target
(F = file, D = directory)? d
task_01b\readonly.txt
1 File(s) copied

```

Рисунок 1.13 – Копирование архивных файлов и для чтения

/A — копирует только файлы с атрибутом "архивный". /R — копирует файлы "только для чтения". /K — сохраняет атрибуты у файлов-результатов.

Перемещаем все текстовые файлы с расширением .txt из каталога task\_01c\_src в каталог task\_01c\_dst.

```

C:\Users\yanpaulovich\OS2>move task_01c_src\*.txt task_01c_dst
C:\Users\yanpaulovich\OS2\task_01c_src\txt1.txt
C:\Users\yanpaulovich\OS2\task_01c_src\txt2.txt
C:\Users\yanpaulovich\OS2\task_01c_src\txt3.txt
C:\Users\yanpaulovich\OS2\task_01c_src\txt4.txt
C:\Users\yanpaulovich\OS2\task_01c_src\txt5.txt
5 file(s) moved.

```

Рисунок 1.14 – Перемещение файлов

Заходим в каталог task\_01d\_2 и удаляем текст в файлах.

```
C:\Users\yanpaulovich\OS2>cd task_01d_2 & for %f in (*.txt) do echo. > "%f"

C:\Users\yanpaulovich\OS2\task_01d_2>echo. 1>"txt1.txt"

C:\Users\yanpaulovich\OS2\task_01d_2>echo. 1>"txt2.txt"

C:\Users\yanpaulovich\OS2\task_01d_2>echo. 1>"txt3.txt"
```

Рисунок 1.15 – Удаление текста в файлах

for %f in (\*.txt) — перебирает все .txt файлы. echo. > "%f" — перезаписывает каждый файл пустым содержимым.

Заменяем файлы в каталоге task\_01d\_2 файлами, расположенными в каталоге task\_01d\_1.

```
C:\Users\yanpaulovich\OS2>replace task_01d_1\* task_01d_2
Replacing C:\Users\yanpaulovich\OS2\task_01d_2\txt1.txt
Replacing C:\Users\yanpaulovich\OS2\task_01d_2\txt2.txt
Replacing C:\Users\yanpaulovich\OS2\task_01d_2\txt3.txt
```

Рисунок 1.16 – Замена файлов

Изменяем расширение всех файлов в каталоге task\_01e на .bmp.

```
C:\Users\yanpaulovich\OS2>ren task_01e\*.* *.bmp
```

Рисунок 1.17 – Изменение расширения

Изменяем имя файла txt1.bmp на user-9e5b2d19.Ren.

```
C:\Users\yanpaulovich\OS2\task_01e>ren txt1.bmp user_9e5b2d19.Ren
```

Рисунок 1.18 – Изменение имени файла

Сравниваем два файла в каталоге task\_01f и сохраняем результат сравнения в result.txt.

```
C:\Users\yanpaulovich\OS2\task_01f>fc txt1.txt txt2.txt > result.txt
```

Рисунок 1.19 – Сравнение файлов

Удаляем файл txt1 из каталога task\_01g, при этом запросив подтверждение на удаление.

```
C:\Users\yanpaulovich\OS2\task_01g\subdir>del /P txt1.txt
C:\Users\yanpaulovich\OS2\task_01g\subdir\txt1.txt, Delete (Y/N)? y
```

Рисунок 1.20 – Удаление файла с запросом на подтверждение

Удаляем все файлы с атрибутом «Системный» из каталога task\_01g.

```
C:\Users\yanpaulovich\OS2>for /R "task_01g" %f in (*) do (attrib "%f" | find /I "S" >nul && attrib -s -h "%f" && del "%f")
```

Рисунок 1.21 – Удаление файлов с атрибутом «Системный»

for /R "task\_01g" — рекурсивно перебирает все файлы в task\_01g и подкаталогах. attrib "%f" — проверяет атрибуты. find /i "S" — ищет системный атрибут. attrib -s -h "%f" — снимает системный и скрытый. del "%f" — удаляет файл.

Удаляем все файлы из каталога task\_01g так, чтобы остались только файлы с атрибутом «Только для чтения».

```
C:\Users\yanpaulovich\OS2>for %f in (task_01g\*) do (attrib "%f" | find /I "R" >nul || del "%f")
```

Рисунок 1.22 – Удаляем файлы

for %f in (task\_01g\\*) — перебирает все файлы в каталоге task\_01g. attrib "%f" — проверяет атрибуты файла. find /i "R" — ищет атрибут Read-only. || del "%f" — если атрибут не найден, файл удаляется.

Удаляем все оставшиеся файлы и дочерние каталоги из каталога task\_01g.

```
C:\Users\yanpaulovich\OS2>del /s /q "task_01g\*"
```

Рисунок 1.23 – Удаление оставшихся файлов

Выводим информацию о BIOS в файл и создаем пять bat-скриптов, демонстрирующих работу с переменными, параметрами и логикой.



```

C:\Users\yanpaulovich\OS2>dir C:\Windows /s /p
Volume in drive C has no label.
Volume Serial Number is D408-8F61

Directory of C:\Windows

10/21/2025  09:17 PM  <DIR>      .
10/07/2025  03:02 PM  <DIR>      appcompat
10/21/2025  09:17 PM  <DIR>      apppatch
10/21/2025  09:06 PM  <DIR>      AppReadiness
10/21/2025  08:38 AM  <DIR>      assembly
10/20/2025  03:40 PM  <DIR>      bcastdvr
10/18/2025  11:44 AM             126,976  bfsvc.exe
04/01/2024  10:34 AM  <DIR>      Boot
04/01/2024  10:26 AM  <DIR>      Branding
10/20/2025  03:40 PM  <DIR>      BrowserCore
10/21/2025  09:17 PM  <DIR>      CbsTemp
04/01/2024  10:22 AM             23,885  Core.xml
04/01/2024  10:26 AM  <DIR>      Cursors
10/07/2025  03:02 PM  <DIR>      debug
04/01/2024  10:34 AM  <DIR>      diagnostics
04/01/2024  10:34 AM  <DIR>      DiagTrack
04/01/2024  11:06 AM  <DIR>      DigitalLocker
10/08/2025  12:23 AM             2,522  DtcInstall.log
06/07/2025  06:39 PM  <DIR>      en-US
10/18/2025  11:45 AM             3,129,376  explorer.exe
04/01/2024  10:26 AM  <DIR>      GameBarPresenceWriter
04/01/2024  11:08 AM  <DIR>      Globalization
04/01/2024  11:06 AM  <DIR>      Help
10/18/2025  11:45 AM             1,081,344  HelpPane.exe
06/07/2025  06:33 PM             40,960  hh.exe
04/01/2024  10:34 AM  <DIR>      IdentityCRL
06/07/2025  06:39 PM  <DIR>      IME
10/21/2025  09:17 PM  <DIR>      ImmersiveControlPanel
06/07/2025  06:39 PM  <DIR>      InboxApps
10/21/2025  08:14 PM  <DIR>      INF
04/01/2024  10:34 AM  <DIR>      InputMethod
04/01/2024  10:26 AM  <DIR>      L2Schemas
04/01/2024  10:26 AM  <DIR>      LiveKernelReports
10/21/2025  08:19 AM  <DIR>      Logs
10/08/2025  12:21 AM             1,380  lsasetup.log
04/01/2024  10:22 AM             43,131  mib.bin

```

Рисунок 1.24 – Просмотр содержимого

/s — рекурсивно показывает все файлы и папки, включая вложенные. /p — выводит постранично, останавливаясь после каждой страницы и ожидая нажатия клавиши.

Постранично выводим все каталоги и файлы, включая вложенные подкаталоги и файлы каталога C:\Users\«пользователь» в отсортированном по алфавиту порядке с записью результата в файл user\_files.txt в каталоге task\_01h.

```

C:\Users\yanpaulovich\OS2>dir C:\Users\yanpaulovich /s /on > task_01h\user_files.txt

```

Рисунок 1.25 – Вывод содержимого по алфавиту и запись в файл

Создаем каталог task\_01j в вашем каталоге OS2.

```

C:\Users\yanpaulovich\OS2>md task_01j

```

Рисунок 1.26 – Создание каталога

Создаем иерархию каталогов task\_01j\sub\dir\full\path

```

C:\Users\yanpaulovich\OS2>md task_01j\sub\dir\full\path

```

Рисунок 1.27 – Создание иерархии

Удаляем пустой каталог \task\_01k\empty\_subdir.

```
C:\Users\yanpaulovich\OS2>rd task_01k\empty_subdir
```

Рисунок 1.28 – Удаление пустого каталога

Удаляем непустой каталог \task\_01k\subdir.

```
C:\Users\yanpaulovich\OS2>rd /S /Q task_01k\subdir
```

Рисунок 1.29 – Удаление не пустого каталога

/s — удаляет всё содержимое, включая вложенные файлы и папки. /q — тихий режим, без подтверждений.

Выполняем следующие задания с использованием различных команд, но с обязательным использованием таких операторов как перенаправление вывода (>, >>), конвейеризации (|) и соединения (&).

Возвращаемся в папку OS2.

```
C:\Users\yanpaulovich\OS2>cd C:\Users\yanpaulovich\OS2
```

Рисунок 1.30 – Возврат в папку OS2

Создаем текстовый файл со списком студентов группы и их номерами книжек, используя команду **type** и ввод с консоли.

```
C:\Users\yanpaulovich\OS2>type con > students.txt
Avr 1
Bon 2
Vol 3
Dro 4
Dub 5
Mil 6
Orl 7
Pav 8
Par 9
Pro 10
Sem 11
Hut 12
Yag 3
Ya 14
^Z
```

Рисунок 1.31 – Создание и заполнение файла

Выполняем поиск студента в файле по его фамилии.

```
C:\Users\yanpaulovich\OS2>find "Pav" students.txt
----- STUDENTS.TXT
Pav 8
```

Рисунок 1.32 – Поиск студента по фамилии

Добавляем еще студентов в конец файла.

```
C:\Users\yanpaulovich\OS2>type con >> students.txt  
Nev 15  
^Z
```

Рисунок 1.33 – Добавление студентов

Выводим список студентов в отсортированном порядке и после этого выходим из папки.

```
C:\Users\yanpaulovich\OS2>type students.txt | sort & cd ..  
Avr 1  
Bon 2  
Dro 4  
Dub 5  
Hut 12  
Mil 6  
Nev 15  
Orl 7  
Par 9  
Pav 8  
Pro 10  
Sem 11  
Vol 3  
Ya 14  
Yag 3
```

Рисунок 1.34 – Вывод списка студентов

Выводим версию BIOS через файл.

```
----- BIOS.TXT  
BIOS Version: Parallels International GmbH. 26.1.1 (572  
88), 9/29/2025  
C:\Users\yanpaulovich>
```

Рисунок 1.35 – Вывод версии BIOS

1. Создаем bat-файл bat\_01, который выводит следующую информацию, используя системные переменные ([ss64](#)): *Имя текущего пользователя, текущие дату и время, имя компьютера.*

```
C:\Users\yanpaulovich\OS2>copy con bat_01.bat
@echo off
echo name %USERNAME%
echo date %DATE%
echo time %TIME%
echo comp %COMPUTERNAME%
1 file(s) copied.

C:\Users\yanpaulovich\OS2>bat_01.bat
name yanpaulovich
date Tue 10/21/2025
time 21:56:32.45
comp YANPAULOVIC5709
```

Рисунок 1.36 – Создание bat-файла bat\_01

2. Создаем bat-файл bat\_02, который выводит следующую информацию, используя расширения параметров ([ss64](#)): *имя текущего бат-файла, время последней модификации и полный путь к текущему бат-файлу.*

```
C:\Users\yanpaulovich\OS2>copy con bat_02.bat
@echo off
echo fname: %~nx0
for %%F in (%~f0) do echo lastchange: %%~tF
echo path: %~f0
1 file(s) copied.

C:\Users\yanpaulovich\OS2>bat_02.bat
fname: bat_02.bat
lastchange: 10/21/2025 09:59 PM
path: C:\Users\yanpaulovich\OS2\bat_02.bat
```

Рисунок 1.37 – Создание bat-файла bat\_02

3. Создаем bat-файл bat\_03, который выводит следующую информацию, используя параметры ([ss64](#)): *строку со всеми переданными параметрами, а также первые 4 параметра по отдельности.*

```

C:\Users\yanpaulovich\OS2>copy con bat_03.bat
@echo off
echo All parameters: %*
echo First parameter: %1
echo Second parameter: %2
echo Third parameter: %3
echo Fourth parameter: %4
1 file(s) copied.

C:\Users\yanpaulovich\OS2>bat_03.bat 1 2 3 4
All parameters: 1 2 3 4
First parameter: 1
Second parameter: 2
Third parameter: 3
Fourth parameter: 4

```

Рисунок 1.38 – Создание bat-файла bat\_03

4. Создаем bat-файл bat\_04, который выводит следующую информацию ([ss64](#)): *результаты следующих математических выражений  $a-b$ ,  $a+b$ ,  $c/b$ ,  $c*b$ ,  $(a-b)*(b-a)$ , где  $a, b, c$  – 1, 2 и 3 параметры передаваемые бат-файлу.*

```

C:\Users\yanpaulovich\OS2>copy con bat_04.bat
@echo off
:: Usage example: bat_04 1 2 3
set /a a=%1, b=%2, c=%3

set /a sub=a-b
set /a add=a+b
set /a div=c/b
set /a mul=c*b
set /a complex=(a-b)*(b-a)

echo a - b = %sub%
echo a + b = %add%
echo c / b = %div%
echo c * b = %mul%
echo (a - b) * (b - a) = %complex%
1 file(s) copied.

C:\Users\yanpaulovich\OS2>bat_04.bat 1 2 3
a - b = -1
a + b = 3
c / b = 1
c * b = 6
(a - b) * (b - a) = -1

```

Рисунок 1.39 – Создание bat-файла bat\_04

5. Создаем bat-файл bat\_05

```

C:\Users\yanpaulovich\OS2>copy con bat_05.bat
@echo off
:: Usage: bat_05 {mode} {filename}
:: Supported modes: create, delete

if "%1"==" " (
    echo Usage: %~n0 {create|delete} filename
    exit /b
)

if "%2"==" " (
    echo File name not specified.
    exit /b
)

if /i "%1"=="create" (
    if exist "%2" (
        echo File "%2" already exists.
    ) else (
        copy nul "%2" >nul
        echo File "%2" has been created.
    )
) else if /i "%1"=="delete" (
    if exist "%2" (
        del "%2"
        echo File "%2" has been deleted.
    ) else (
        echo File "%2" not found.
    )
) else (
    echo Unknown mode. Use "create" or "delete".
)

1 file(s) copied.

```

Рисунок 1.41 – Создание bat-файла bat\_05

После создания файла проверяем, как он работает.

```

C:\Users\yanpaulovich\OS2>bat_05 create test.txt
File "test.txt" has been created.

C:\Users\yanpaulovich\OS2>bat_05 delete test.txt
File "test.txt" has been deleted.

C:\Users\yanpaulovich\OS2>bat_05.bat
'delete' is not recognized as an internal or external command,
operable program or batch file.

```

Рисунок 1.42 – Проверка использования

Таблица 1.1 – Примеры наименований команд

Команда	Происхождение	Значение
dir	directory	Показать содержимое каталога
cd	change directory	Перейти в другой каталог
md / mkdir	make directory	Создать каталог
rd / rmdir	remove directory	Удалить каталог

del	delete	Удалить файл
copy	copy	Копировать файл
type	type out	Вывести содержимое файла
echo	echo	Отобразить строку
set	set variable	Задать переменную
pause	pause execution	Приостановить выполнение
find	find string	Найти строку в файле
sort	sort lines	Отсортировать строки
systeminfo	system information	Вывести сведения о системе

Таблица 1.2 – Примеры наименований параметров

Команда	Параметр	Значение
dir	/s	subdirectories — показать вложенные каталоги
dir	/p	page — постраничный вывод
copy	/y	yes — не спрашивать подтверждение
rd	/s	subtree — удалить всё содержимое
rd	/q	quiet — без подтверждений
icacls	/grant	grant access — выдать права
icacls	/T	tree — рекурсивно по дереву
attrib	-s	system — снять системный атрибут
attrib	-h	hidden — снять скрытый атрибут
set	/a	arithmetic — режим вычислений

6. Переменные окружения — это специальные переменные, которые хранят системную информацию и доступны всем процессам в операционной системе. Они задают контекст выполнения команд, скриптов и программ.

Таблица 1.3 – Переменные окружения

Переменная	Назначение
PATH	Список каталогов, в которых система ищет исполняемые файлы
PATHEXT	Расширения файлов, считающихся исполняемыми
COMSPEC	Путь к интерпретатору командной строки (cmd.exe)
TEMP / TMP	Путь к временной папке пользователя
USERNAME	Имя текущего пользователя
USERPROFILE	Путь к домашнему каталогу пользователя
HOMEPATH	Относительный путь к домашнему каталогу
HOMEDRIVE	Диск, на котором расположен домашний каталог

LOCALAPPDATA	Путь к папке Local для хранения локальных данных приложений
NUMBER OF PROCESSORS	Количество логических процессоров
PROCESSOR_ARCHITECTURE	Архитектура процессора
PROCESSOR_IDENTIFIER	Строка с названием процессора
PROCESSOR_LEVEL	Уровень процессора
OS	Название операционной системы
LOGONSERVER	Сервер, через который выполнен вход в систему
COMPUTERNAME	Имя компьютера
USERDOMAIN	Домен пользователя
PROMPT	Настройка приглашения командной строки
ERRORLEVEL	Код возврата последней выполненной команды
CMDLINE	Путь к интерпретатору командной строки (cmd.exe)



## 2 Linux

Таблица 2.1 – Аналоги команд

Windows CMD	Linux Bash	Назначение	Отличие в стиле
dir	ls	Список файлов	ls — кратко, параметры через -
cd	cd	Переход в каталог	одинаково
md / mkdir	mkdir	Создание каталога	одинаково
rd / rmdir	rmdir	Удаление каталога	одинаково
del	rm	Удаление файла	rm — более мощная, может удалять рекурсивно
copy	cp	Копирование	cp — параметры через -r, -v
type	cat	Просмотр файла	cat — потоковый вывод
echo	echo	Вывод строки	одинаково
find	grep	Поиск по строке	grep — регулярные выражения
sort	sort	Сортировка строк	одинаково
systeminfo	uname, lscpu, ls_b_release	Информация о системе	разбито на несколько утилит

```
GNU nano 7.2
#!/bin/bash
echo "User: $USER"
echo "Date: $(date +%F)"
echo "Name: $HOSTNAME"
echo "Time: $(date +%H:%M:%S)"
```

Рисунок 2.1 – Первый скрипт

```
user-9e5b2d19@linUb:~/lab2$ ./bash_01.sh
User: user-9e5b2d19
Date: 2025-10-21
Name: linUb
Time: 19:35:00
```

Рисунок 2.2 – Запуск первого скрипта

```
GNU nano 7.2
#!/bin/bash
echo "name: $(basename "$0")"
echo "path: $(realpath "$0")"
echo "date: $(stat -c %y "$0")"
```

Рисунок 2.3 – Второй скрипт

```
user-9e5b2d19@linUb:~/lab2$ chmod +x bash_02.sh
user-9e5b2d19@linUb:~/lab2$ ./bash_02.sh
name: bash_02.sh
path: /home/user-9e5b2d19/lab2/bash_02.sh
date: 2025-10-21 19:39:36.601928219 +0000
```

Рисунок 2.4 – Запуск второго скрипта

```
GNU nano 7.2
#!/bin/bash
echo "all: $@"
echo "1: $1"
echo "2: $2"
echo "3: $3"
echo "4: $4"
```

Рисунок 2.5 – Третий скрипт

```
user-9e5b2d19@linUb:~/lab2$ ./bash_03.sh 1 2 3 4
all: 1 2 3 4
1: 1
2: 2
3: 3
4: 4
```

Рисунок 2.6 – Запуск третьего скрипта

```

GNU nano 7.2                                bash_04.sh *
#!/bin/bash
a=$1
b=$2
c=$3

echo "a - b = $((a - b))"
echo "a + b = $((a + b))"
echo "a / b = $((a / b))"
echo "a * b = $((a * b))"
echo "(a - b) * (b - a) = $(((a - b) * (b - a)))"

```

Рисунок 2.7 – Четвертый скрипт

```

user-9e5b2d19@linUb:~/lab2$ ./bash_04.sh 1 2 3
a - b = -1
a + b = 3
a / b = 0
a * b = 2
(a - b) * (b - a) = -1
user-9e5b2d19@linUb:~/lab2$

```

Рисунок 2.8 – Запуск четвертого скрипта

```

GNU nano 7.2                                bash_5.sh
#!/bin/bash
mode=$1
file=$2

if [[ -z "$mode" || -z "$file" ]]; then
echo "use: bash_05.sh create|delete file_name"
exit 1
fi

if [[ "$mode" == "create" ]]; then
if [[ -e "$file" ]]; then
echo "exists"
else
touch "$file"
echo "created"
fi
elif [[ "$mode" == "delete" ]]; then
if [[ -e "$file" ]]; then
rm "$file"
echo "deleted"
else
echo "nf"
fi
else
echo "invalid"
fi

```

Рисунок 2.9 – Пятый скрипт

```

user-9e5b2d19@linUb:~/lab2$ ./bash_5.sh create test.txt
created
user-9e5b2d19@linUb:~/lab2$ ./bash_5.sh delete test.txt
deleted

```

Рисунок 2.10 – Запуск пятого скрипта

Все bash-скрипты аналогичны тем, что были в Windows и работают по тому же принципу.

Таблица 2.2 – Переменные окружения

Переменная	Назначение
PATH	Список директорий, в которых система ищет исполняемые файлы
HOME	Домашний каталог текущего пользователя
USER	Имя текущего пользователя
LOGNAME	Имя пользователя, вошедшего в систему
SHELL	Путь к используемой командной оболочке
PWD	Текущий рабочий каталог
LANG	Язык и локаль по умолчанию
LC_ALL	Переопределение всех локалей
TERM	Тип терминала
HOSTNAME	Имя компьютера
UID	Числовой идентификатор пользователя
MAIL	Путь к почтовому ящику пользователя
DISPLAY	Адрес графического дисплея
SHLVL	Уровень вложенности оболочек
PS1	Основное приглашение командной строки, которое видно каждый раз, когда Bash ждет команду
PS2	Вторичное приглашение, которое появляется, если не завершена команда
RANDOM	Генератор случайных чисел
GROUPS	Список групп, в которых состоит пользователь
BASH_VERSION	Версия Bash
BASH_SOURCE	Путь к исполняемому скрипту
?	Код возврата последней команды

## **Заключение**

В результате выполнения лабораторной работы были получены практические навыки работы с командными оболочками cmd, PowerShell и bash. В среде Windows освоены основные команды для управления файлами и каталогами, перенаправления потоков и создания пакетных bat-файлов с переменными и логическими конструкциями. Исследованы правила именования команд, их параметров и структура переменных окружения.

В рамках работы в Linux проведено сравнение команд Windows и их аналогов в bash, выявлены различия в подходах к именованию и использованию параметров. Полученные знания применены на практике при написании эквивалентных bash-скриптов. Также проведен анализ переменных окружения Linux и проведено их сравнение с реализацией в Windows.

Работа продемонстрировала ключевые сходства и различия в работе с командной строкой двух операционных систем, закрепив понимание их основ и принципов автоматизации задач.