

**Министерство науки и высшего образования
Российской Федерации**

**Федеральное государственное автономное
образовательное учреждение**

высшего образования

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»**

(МОСКОВСКИЙ ПОЛИТЕХ)

Лабораторная работа 1

**По теме: «Программа на bash для сбора и вывода
информации о системе»**

По дисциплине:

«Администрирование серверов»

Выполнил

Студент 3 курса

Киркова Кристина С.

Гр 231-321

Проверил

Гневшев А.Ю.

Москва 2025

1. Цель работы

Получить практические навыки работы с командной оболочкой `bash` и утилитами для получения информации о системе, а также освоить базовые алгоритмические конструкции скриптов.

2. Постановка задачи

Написать скрипт на `bash`, который собирает информацию о системе и выводит её в интерактивном (меню) и автоматическом режимах. При запуске без параметров — показывать меню на русском языке; пользователь выбирает пункт по номеру; после вывода каждой информации показывается «Для продолжения нажмите Enter», затем меню показывается снова. Также предусмотреть опцию командной строки `--tofile <filename>`, которая выводит всю собираемую информацию в указанный файл. Блоки информации должны разделяться строкой из 40 символов `=`. В отчёт прикрепить файл `output.txt` с выводом всей собираемой информации.

3. Краткое описание функционала программы

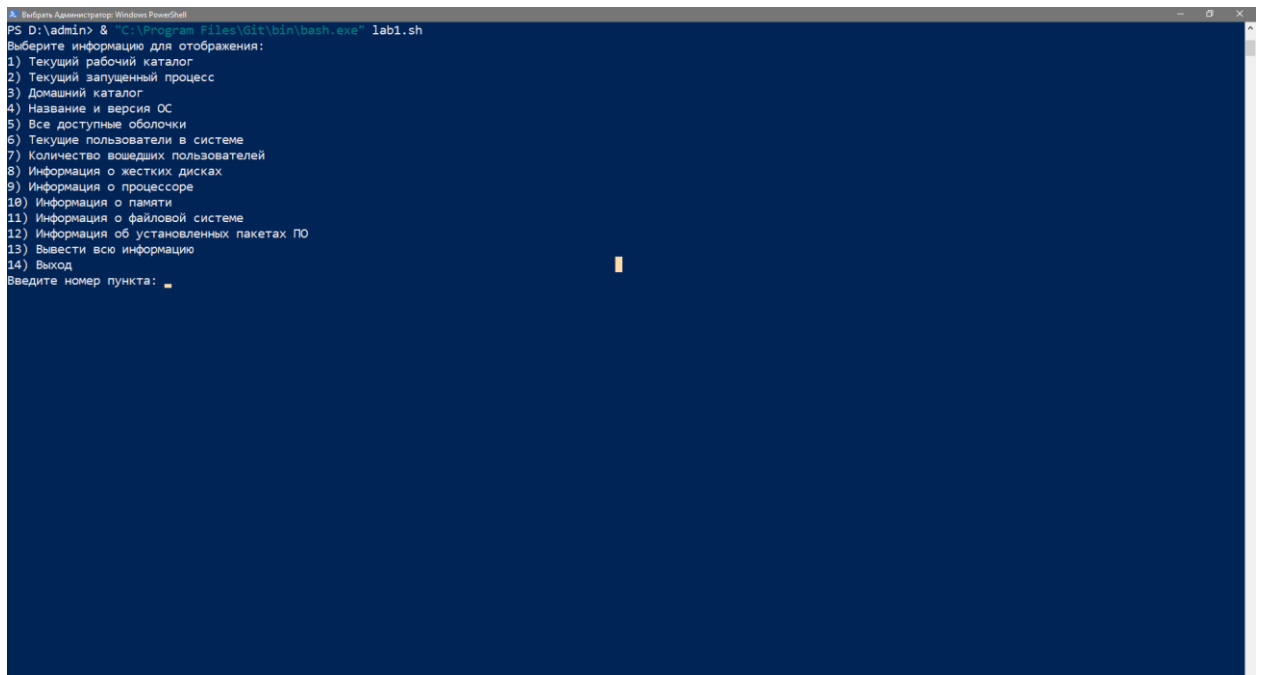
Скрипт собирает и выводит (каждый пункт — отдельная опция меню):

1. Текущий рабочий каталог
2. Текущий запущенный процесс (PID скрипта)
3. Домашний каталог
4. Название и версия ОС
5. Все доступные оболочки (содержимое `/etc/shells`)
6. Текущие пользователи, вошедшие в систему (`who`)
7. Количество вошедших пользователей
8. Информация о жестких дисках (`df -h`)
9. Информация о процессоре (`lscpu`)
10. Информация о памяти (`free -h`)
11. Информация о файловой системе (`mount`)
12. Информация об установленных пакетах ПО (для Debian/Ubuntu — `dpkg -l`)
13. Пункт «Вывести всю информацию» (сохранённый формат с разделителями `=`)
14. Выход

Также реализована опция `--tofile <имя_файла>`: при запуске скрипта с этой опцией весь сбор сведений записывается в указанный файл и программа завершает работу.

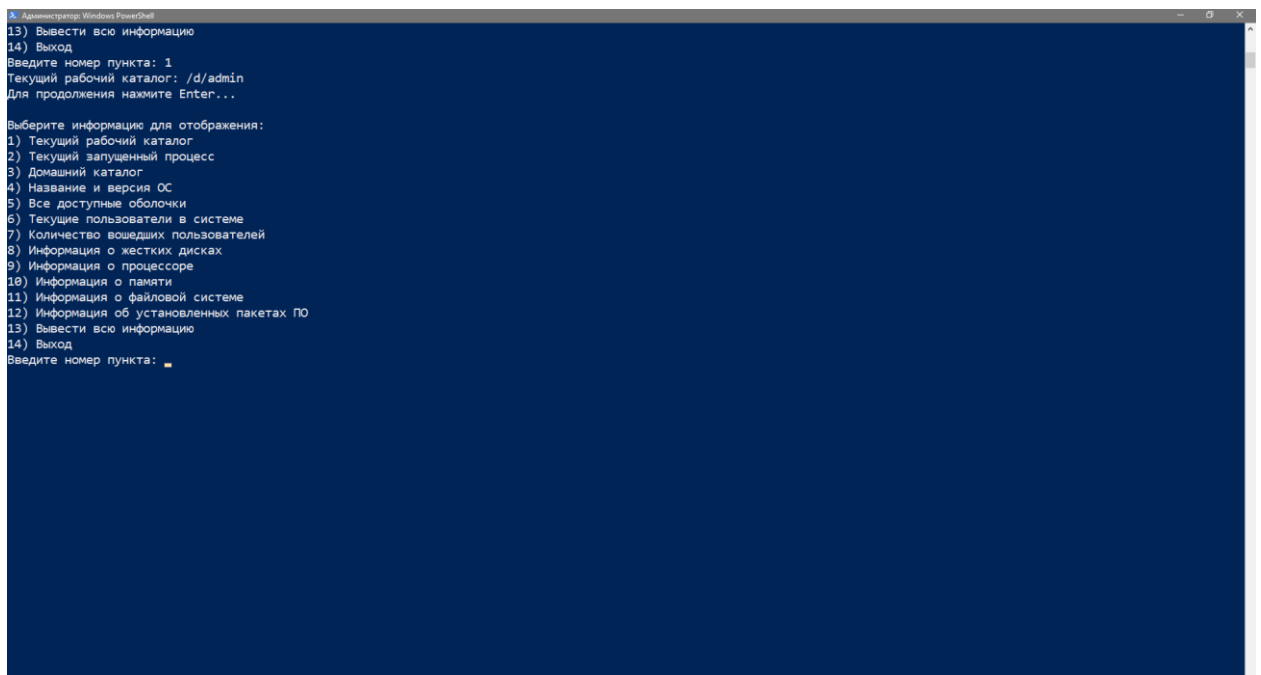
4. Скриншоты интерфейса программы (инструкция)

- Скриншот запуска меню (скрипт запущен без параметров, видно список пунктов меню).



```
Выборы Администратор: Windows PowerShell
PS D:\admin> & "C:\Program Files\Git\bin\bash.exe" lab1.sh
Выберите информацию для отображения:
1) Текущий рабочий каталог
2) Текущий запущенный процесс
3) Домашний каталог
4) Название и версия ОС
5) Все доступные оболочки
6) Текущие пользователи в системе
7) Количество вошедших пользователей
8) Информация о жестких дисках
9) Информация о процессоре
10) Информация о памяти
11) Информация о файловой системе
12) Информация об установленных пакетах ПО
13) Вывести всю информацию
14) Выход
Введите номер пункта: █
```

- Скриншот результата выбора одного из пунктов (например, пункт 1 — pwd).



```
13) Вывести всю информацию
14) Выход
Введите номер пункта: 1
Текущий рабочий каталог: /d/admin
Для продолжения нажмите Enter...

Выберите информацию для отображения:
1) Текущий рабочий каталог
2) Текущий запущенный процесс
3) Домашний каталог
4) Название и версия ОС
5) Все доступные оболочки
6) Текущие пользователи в системе
7) Количество вошедших пользователей
8) Информация о жестких дисках
9) Информация о процессоре
10) Информация о памяти
11) Информация о файловой системе
12) Информация об установленных пакетах ПО
13) Вывести всю информацию
14) Выход
Введите номер пункта: █
```

- Скриншот результата пункта «13) Вывести всю информацию» или содержимого output.txt.


```

echo "Использование: $0 [опции]"
echo "Опции:"
echo "  --tofile <файл> Сохранить информацию в указанный файл"
echo "  -h             Показать справку"
exit 0
}

display_menu() {
echo "Выберите информацию для отображения:"
echo "1) Текущий рабочий каталог"
echo "2) Текущий запущенный процесс"
echo "3) Домашний каталог"
echo "4) Название и версия ОС"
echo "5) Все доступные оболочки"
echo "6) Текущие пользователи в системе"
echo "7) Количество вошедших пользователей"
echo "8) Информация о жестких дисках"
echo "9) Информация о процессоре"
echo "10) Информация о памяти"
echo "11) Информация о файловой системе"
echo "12) Информация об установленных пакетах ПО"
echo "13) Вывести всю информацию"
echo "14) Выход"
}

collect_info() {
echo "===== "
echo "Текущий рабочий каталог: $(pwd)"
echo "===== "
echo "Текущий запущенный процесс: $$"
echo "===== "
echo "Домашний каталог: $HOME"
echo "===== "
echo "Название и версия ОС: $(uname -a)"
echo "===== "
echo "Доступные оболочки:"
cat /etc/shells
echo "===== "
echo "Текущие пользователи:"
who
echo "===== "
echo "Количество вошедших пользователей: $(who | wc -l)"
echo "===== "
echo "Информация о жестких дисках:"
df -h

```

```

echo "=====
echo "Информация о процессоре:"
lscpu
echo "=====
echo "Информация о памяти:"
free -h
echo "=====
echo "Информация о файловой системе:"
mount | column -t
echo "=====
echo "Информация об установленных пакетах ПО:"
dpkg -l 2>/dev/null
echo "=====
}

if [[ $1 == "--tofile" && -n $2 ]]; then
    collect_info > "$2"
    echo "Информация сохранена в файл $2"
    exit 0
fi

if [[ $1 == "-h" ]]; then
    show_help
fi

while true; do
    display_menu
    read -p "Введите номер пункта: " choice
    case $choice in
        1) echo "Текущий рабочий каталог: $(pwd)" ;;
        2) echo "Текущий запущенный процесс: $$" ;;
        3) echo "Домашний каталог: $HOME" ;;
        4) echo "Название и версия ОС: $(uname -a)" ;;
        5) echo "Доступные оболочки:"; cat /etc/shells ;;
        6) echo "Текущие пользователи:"; who ;;
        7) echo "Количество вошедших пользователей: $(who | wc -l)" ;;
        8) echo "Информация о жестких дисках:"; df -h ;;
        9) echo "Информация о процессоре:"; lscpu ;;
        10) echo "Информация о памяти:"; free -h ;;
        11) echo "Информация о файловой системе:"; mount | column -t ;;
        12) echo "Информация об установленных пакетах ПО:"; dpkg -l
2>/dev/null ;;
        13) collect_info ;;
        14) echo "Выход..."; exit 0 ;;
        *) echo "Некорректный ввод, попробуйте снова." ;;
    esac
done

```

```
esac  
echo "Для продолжения нажмите Enter..."  
read  
done
```