

# Работа с unix

## Лабораторная работа 12

Смольняков Данил Евгеньевич

Архитектура компьютеров и ОС

### Цель работы

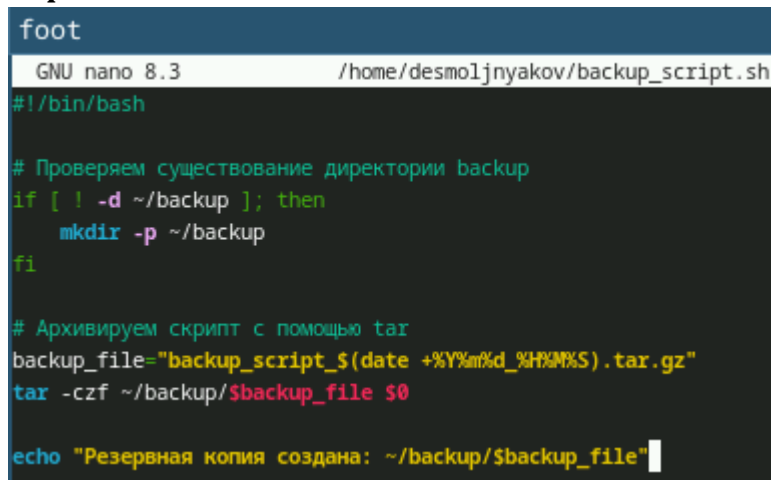
#### Цель

- Изучить основы программирования в оболочке bash.
  - Научиться создавать и выполнять командные файлы.
- 

### Скриншоты и описание действий

#### 1. Создание скрипта резервного копирования

- **Действие:** Создаем скрипт для автоматического бэкапа:
- **Скриншот:**



```
foot
GNU nano 8.3 /home/desmoljnyakov/backup_script.sh
#!/bin/bash

# Проверяем существование директории backup
if [ ! -d ~/backup ]; then
    mkdir -p ~/backup
fi

# Архивируем скрипт с помощью tar
backup_file="backup_script_$(date +%Y%m%d_%H%M%S).tar.gz"
tar -czf ~/backup/$backup_file $0

echo "Резервная копия создана: ~/backup/$backup_file"
```

Создание скрипта резервного копирования

---

#### 2. Обработка аргументов командной строки

- **Действие:** Пишем обработчик аргументов командной строки
- **Скриншот:**

```

foot
GNU nano 8.3 /home/desmoljnyakov/args_processor.sh
echo "Всего аргументов: $#"
```

```

echo "Список аргументов:"

i=1
for arg in "$@"; do
    echo "Аргумент $i: $arg"
    ((i++))
done

```

Обработка аргументов командной строки

---

### 3. Реализация аналога команды ls

- **Действие:** Разрабатываем упрощенный аналог команды ls
- **Скриншот:**

```

foot
GNU nano 8.3 /home/desmoljnyakov/my_ls.sh Изменё
#!/bin/bash

dir=${1:-.} # Используем текущий каталог, если аргумент не указан

echo "Содержимое каталога $dir:"
echo "-----"

for file in "$dir"/*; do
    if [ -d "$file" ]; then
        type="Каталог"
    elif [ -f "$file" ]; then
        type="Файл"
    else
        type="Другое"
    fi

    permissions=""
    if [ -r "$file" ]; then permissions="${permissions}r"; else permissions="${perm
    if [ -w "$file" ]; then permissions="${permissions}w"; else permissions="${perm
    if [ -x "$file" ]; then permissions="${permissions}x"; else permissions="${perm

    echo "Type: ${file##*/} | Права: $permissions"
done

```

Реализация аналога команды ls

---

### 4. Подсчет файлов по расширению

- **Действие:** Создаем счетчик файлов по расширению
- **Скриншот:**

```
foot
GNU nano 8.3 /home/desmoljnyakov/count_files.sh
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Использование: $0 <расширение> <директория>"
    exit 1
fi

extension=$1
directory=$2

if [ ! -d "$directory" ]; then
    echo "Ошибка: директория $directory не существует"
    exit 1
fi

count=$(find "$directory" -type f -name ".*$extension" | wc -l)

echo "Количество файлов с расширением .$extension в $directory: $count"
```

Подсчет файлов по расширению

---

## 5. Отображение созданных скриптов

- **Действие:** Убеждаемся в корректном создании всех скриптов
- **Скриншот:**

```
-rw-r--r-- 1 desmoljnyakov desmoljnyakov 164 мая  3 22:33 /home/desmoljnyakov/args_processor.sh
-rw-r--r-- 1 desmoljnyakov desmoljnyakov 368 мая  3 22:36 /home/desmoljnyakov/backup_script.sh
-rw-r--r-- 1 desmoljnyakov desmoljnyakov 460 мая  3 22:35 /home/desmoljnyakov/count_files.sh
-rw-r--r-- 1 desmoljnyakov desmoljnyakov  98 апр 26 23:13 /home/desmoljnyakov/lab07.sh
-rw-r--r-- 1 desmoljnyakov desmoljnyakov 783 мая  3 22:34 /home/desmoljnyakov/my_ls.sh
[desmoljnyakov@desmoljnyakov ~]$
```

Отображение созданных скриптов

---

## Контрольные вопросы

### Ответы

1. Командная оболочка - интерпретатор команд ОС (bash, zsh, ksh), отличаются функционалом и синтаксисом.
2. POSIX - стандарт совместимости UNIX-систем, обеспечивающий переносимость.
3. Переменные - var=значение, массивы - arr=(знач1 знач2).
4. let - арифметические вычисления, read - чтение ввода пользователя.
5. Арифметические операции - + - \* / % \*\* и битовые операции.
6. (( )) - конструкция для арифметических вычислений и сравнений.

- 7.Стандартные переменные - PATH, HOME, USER, PS1, PWD.
- 8.Метасимволы - спецсимволы (\* ? > < |), имеющие особое значение.
- 9.Экранирование - через , ' ' или " " (частичное).
- 10.Создание - текстовый файл с shebang (!/bin/bash), запуск - chmod +x и ./файл.
- 11.Функции - func\_name() { команды } или function func\_name { команды }.
- 12.Проверка - test -f (файл), test -d (каталог) или [ -f файл ].
- 13.set - управление параметрами оболочки, typeset - объявление переменных, unset - удаление.
- 14.Передача параметров - через аргументы командной строки (\$1, 2, ...n).
- 15.Специальные переменные:\$# - количество, \$\* - все, \$? - код, \$\$ - PID, \$0 - имя

## Заключение

### Выводы

- Освоены основы bash: переменные, циклы, условия.
- Созданы скрипты для автоматизации задач в Linux.