

# Лабораторная работа №11

Курс “Операционные Системы”

---

Лихтенштейн А.А., НКАбд-03-22

15 апреля 2023

Российский университет дружбы народов, Москва, Россия

- Лихтенштейн Алина Алексеевна
- студент группы НКАбд-03-22
- кафедры Компьютерные и информационные науки
- Российский университет дружбы народов
- 1132229533@pfur.ru

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд архивации необходимо узнать, изучив справку.
2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.

3. Написать командный файл — аналог команды `ls` (без использования самой этой команды и команды `dir`). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.
4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (`.txt`, `.doc`, `.jpg`, `.pdf` и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

## Выполнение лабораторной работы

---

## Исходный код скрипта №1



aaliechtenstein@aaliechtenstein:~ — vim script1.sh

```
#!/bin/bash

# определяем имя текущего скрипта
script_name="script1.sh"

# создаем директорию backup, если ее еще нет
mkdir -p "$HOME/backup"

# создаем имя файла архива
backup_filename="$HOME/backup/${script_name%.*}.zip"

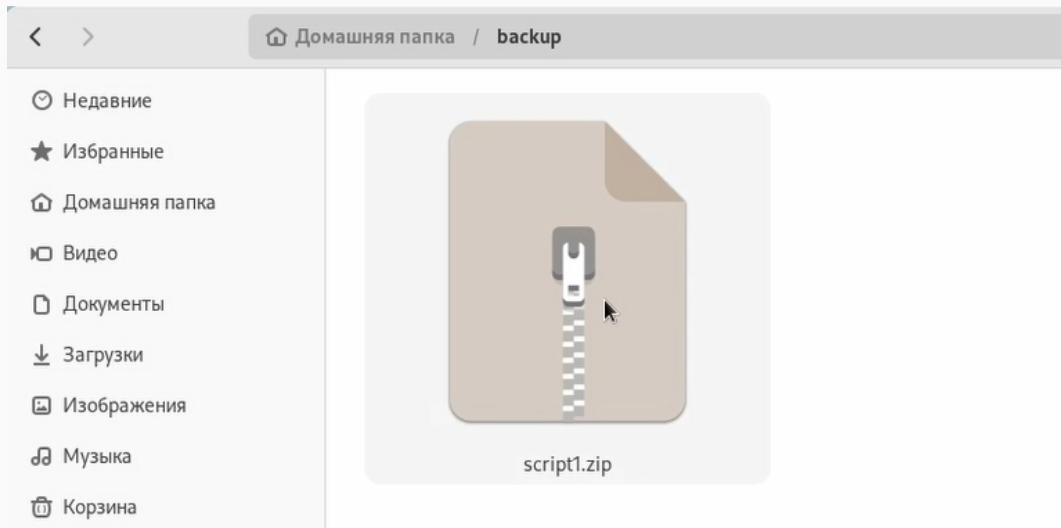
# создаем резервную копию текущего скрипта в архив zip
zip -r "$backup_filename" "$0"

# выводим сообщение об успешном завершении
echo "Резервная копия $script_name сохранена в $backup_filename"
```

Этот скрипт создает резервную копию самого себя (то есть файла, в котором содержится исходный код скрипта) в формате zip и сохраняет ее в директории “backup” в домашней директории пользователя. Сначала скрипт определяет имя текущего скрипта с помощью команды `basename`, затем создает директорию “backup” с помощью команды `mkdir -p`. Затем скрипт формирует имя файла архива с помощью переменной `backup_filename`, используя имя текущего скрипта и расширение “.zip”. Далее скрипт создает резервную копию самого себя с помощью команды `zip`, которой передается имя файла архива и имя текущего скрипта. В конце скрипт выводит сообщение об успешном завершении операции.



## Результат выполнения скрипта №1





```
#!/bin/sh

# проверяем, были ли переданы аргументы
if [ $# -eq 0 ]
then
    echo "Аргументы командной строки не были переданы."
    exit 1
fi

# перебираем все переданные аргументы и выводим их значения
for arg in "$@"
do
    echo "Аргумент: $arg"
done

exit 0
```

Этот скрипт обрабатывает аргументы командной строки и последовательно выводит их значения. Если аргументы не были переданы, то выводится сообщение об ошибке, и скрипт завершается с кодом 1. Если аргументы были переданы, то они перебираются в цикле, и для каждого аргумента выводится его значение с помощью команды `echo`. Завершение скрипта происходит с кодом 0.

## Результат выполнения скрипта №2

```
aaliechtenstein@aaliechtenstein:~  
[aaliechtenstein@aaliechtenstein ~]$ vim script2.sh  
[aaliechtenstein@aaliechtenstein ~]$ bash script2.sh 1 2 3 4 5 6 7 8 9 10 11  
Аргумент: 1  
Аргумент: 2  
Аргумент: 3  
Аргумент: 4  
Аргумент: 5  
Аргумент: 6  
Аргумент: 7  
Аргумент: 8  
Аргумент: 9  
Аргумент: 10  
Аргумент: 11  
[aaliechtenstein@aaliechtenstein ~]$
```

## Исходный код скрипта №3

```
aaliechtenstein@aaliechtenstein:~ — vim script3.sh

# Проверяем, существует ли каталог
if [ ! -d "$1" ]; then
    echo "$1 is not a directory"
    exit 1
fi

# Выводим список файлов в каталоге
for file in "$1"/*
do
    # Выводим имя файла
    echo -n "$file "

    # Выводим права доступа к файлу
    if [ -r "$file" ]; then
        echo -n "r"
    else
        echo -n "-"
    fi

    if [ -w "$file" ]; then
        echo -n "w"
    else
        echo -n "-"
    fi

    if [ -x "$file" ]; then
        echo "x"
    else
        echo "-"
    fi
done
```

Сначала проверяется, был ли передан путь к каталогу в качестве аргумента командной строки, используя проверку на пустоту переменной "\$1". Если переменная пуста, то выводится сообщение об использовании скрипта и завершается его выполнение с кодом ошибки 1.

Затем проверяется, существует ли указанный каталог, используя проверку с помощью команды "[ ! -d "\$1" ]". Если каталог не существует, то выводится сообщение об ошибке и скрипт завершается с кодом ошибки 1.

Далее используется цикл “for” для перебора файлов в заданном каталоге. Для каждого файла выводится его имя с помощью команды “echo -n”, которая не переводит строку на новую строку. Затем для каждого файла проверяется наличие прав доступа на чтение, запись и выполнение с помощью команд “if [ -r "\$file" ]”, “if [ -w "\$file" ]” и “if [ -x "\$file" ]”. Если права доступа есть, то выводится соответствующий символ (r, w или x), если прав доступа нет, то выводится дефис (-).

Итоговый вывод командного файла будет представлять собой список файлов в заданном каталоге с указанием прав доступа к каждому файлу.

## Результат выполнения скрипта №3

```
aaliechtenstein@aaliechtenstein:~  
[aaliechtenstein@aaliechtenstein ~]$ vim script3.sh  
[aaliechtenstein@aaliechtenstein ~]$ bash script3.sh /home/aaliechtenstein/  
/home/aaliechtenstein//backup rwx  
/home/aaliechtenstein//bin rwx  
/home/aaliechtenstein//lab.09.sh rw-  
/home/aaliechtenstein//lab.09.sh~ rw-  
/home/aaliechtenstein//script1.sh rw-  
/home/aaliechtenstein//script2.sh rw-  
/home/aaliechtenstein//script3.sh rw-  
/home/aaliechtenstein//work rwx  
/home/aaliechtenstein//Видео rwx  
/home/aaliechtenstein//Документы rwx  
/home/aaliechtenstein//Загрузки rwx  
/home/aaliechtenstein//Изображения rwx  
/home/aaliechtenstein//Музыка rwx  
/home/aaliechtenstein//Общедоступные rwx  
/home/aaliechtenstein//Рабочий стол rwx  
/home/aaliechtenstein//Шаблоны rwx  
[aaliechtenstein@aaliechtenstein ~]$
```



```
#!/bin/bash

# Проверяем, что передано два аргумента командной строки
if [ $# -ne 2 ]; then
    echo "Usage: $0 <directory_path> <file_extension>"
    exit 1
fi

# Проверяем, что первый аргумент - директория, существует и доступна для чтения
if [ ! -d "$1" ] || [ ! -r "$1" ]; then
    echo "$1 is not a valid directory or is not readable"
    exit 1
fi

# Считаем количество файлов с заданным расширением
count=$(find "$1" -maxdepth 1 -type f -name ".*$2" | wc -l)

echo "There are $count files with extension .$2 in directory $1"
```

Данный скрипт предназначен для поиска файлов с заданным расширением в указанной директории. Скрипт принимает два аргумента командной строки: путь к директории и расширение файла. С помощью команды `find` скрипт ищет все файлы с указанным расширением в указанной директории и ее поддиректориях. Количество найденных файлов определяется с помощью команды `wc`. Наконец, скрипт выводит сообщение с количеством найденных файлов. Если переданы неверные аргументы командной строки, скрипт выводит сообщение об использовании и завершается с ошибкой.

## Результат выполнения скрипта №4

```
aaliechtenstein@aaliechtenstein:~  
[aaliechtenstein@aaliechtenstein ~]$ vim script4.sh  
[aaliechtenstein@aaliechtenstein ~]$ bash script4.sh /home/aaliechtenstein/ txt  
There are 0 files with extension .txt in directory /home/aaliechtenstein/  
[aaliechtenstein@aaliechtenstein ~]$ touch a.txt b.txt c.txt  
[aaliechtenstein@aaliechtenstein ~]$ bash script4.sh /home/aaliechtenstein/ txt  
There are 3 files with extension .txt in directory /home/aaliechtenstein/  
[aaliechtenstein@aaliechtenstein ~]$ touch 1.pdf 2.pdf  
[aaliechtenstein@aaliechtenstein ~]$ bash script4.sh /home/aaliechtenstein/ pdf  
There are 2 files with extension .pdf in directory /home/aaliechtenstein/  
[aaliechtenstein@aaliechtenstein ~]$
```

## Результаты

---

Были получены навыки написания небольших командных файлов в оболочке ОС UNIX/Linux.