

# Программирование в командном процессоре. Командные файлы

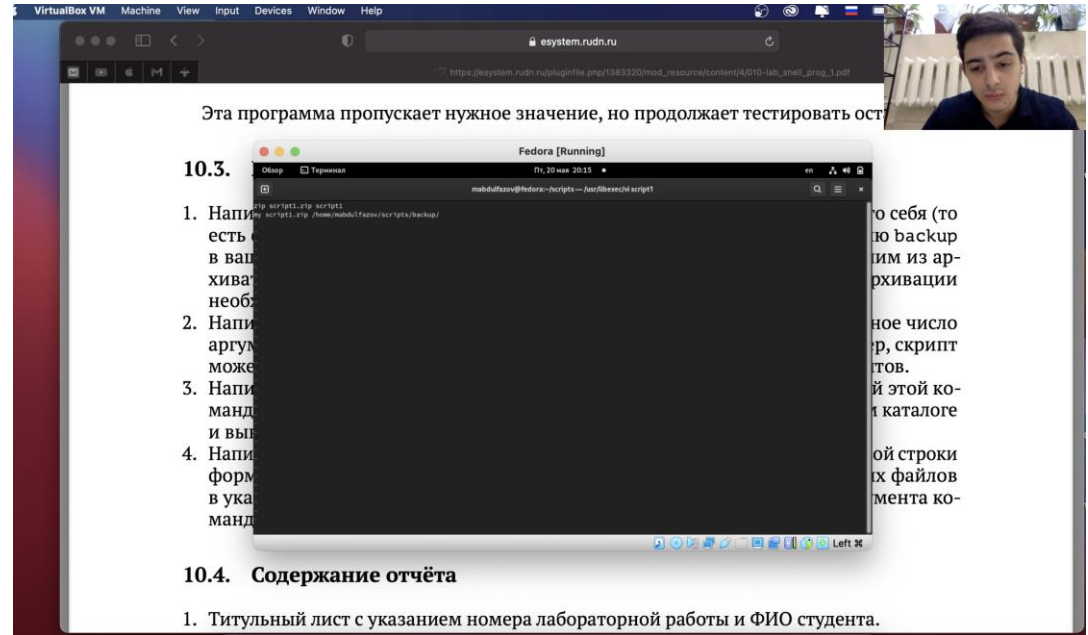
Мансур А. о. Абдулфазов Группа НФИбд-01-21

# Цель работы

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

## Выполнение лабораторной работы

### 1. Скрипт 1 (рис. 1)



The screenshot shows a VirtualBox VM window titled "VirtualBox VM". Inside, a presentation slide is visible with the URL "esystem.rudn.ru" in the browser address bar. The slide content includes the text "Эта программа пропускает нужное значение, но продолжает тестировать ост" and a list of tasks under the heading "10.3.". A terminal window titled "Fedora [Running]" is overlaid on the slide, showing a command prompt with the user "mahul@mahul:~/scripts" and the command "cat script1.sh". The terminal output is mostly blacked out. To the right of the terminal, there is a small video feed of a person.

Эта программа пропускает нужное значение, но продолжает тестировать ост

### 10.3.

1. Напи...  
есть с...  
в вап...  
хиват...  
необх...
2. Напи...  
аргум...  
може...
3. Напи...  
манд...  
и выл...
4. Напи...  
форм...  
в ука...  
манд...

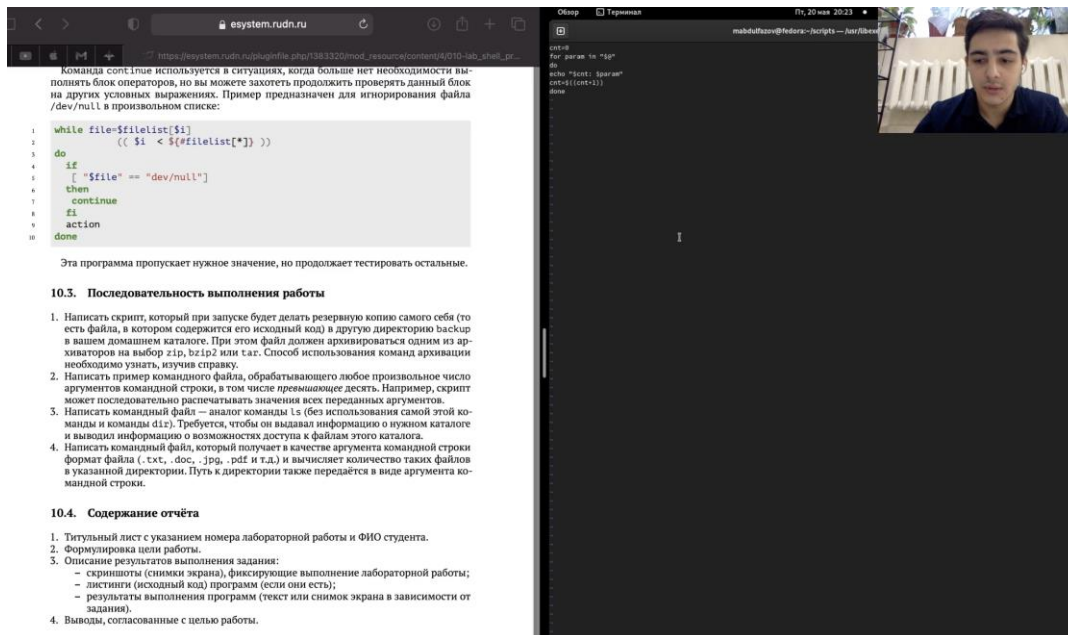
то себя (то  
ю backup  
им из ар-  
рхивации  
ное число  
р, скрипт  
тов.  
й этой ко-  
1 каталоге  
ой строки  
х файлов  
мента ко-

### 10.4. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.

## Скрипт 1

## 2. Скрипт 2 (рис. 2)



Команда `continue` используется в ситуациях, когда больше нет необходимости выполнять блок операторов, но вы можете захотеть продолжить проверять данный блок на других условиях выражения. Пример предназначен для игнорирования файла `/dev/null` в произвольном списке:

```
1 while file=${filelist[$i]}
2   (( ${i} < ${#filelist[*]} ))
3 do
4   if [ "$file" == "/dev/null" ]
5   then
6     continue
7   fi
8   action
9
10 done
```

Эта программа пропускает нужное значение, но продолжает тестировать остальные.

### 10.3. Последовательность выполнения работы

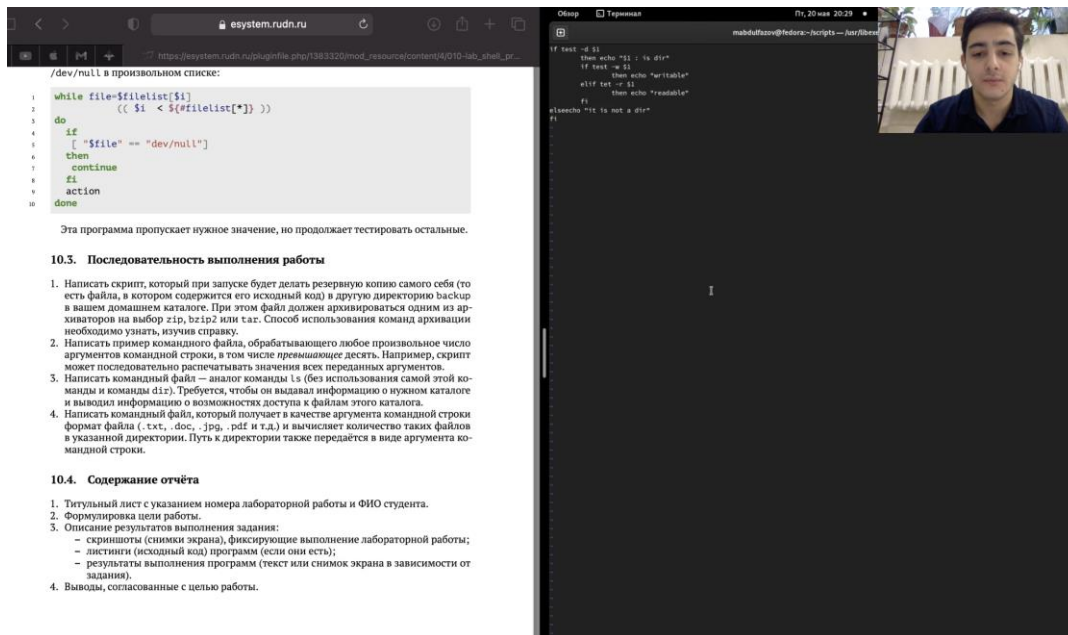
1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию `backup` в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор `zip`, `bzip2` или `tar`. Способ использования 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 рассчитывать 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.

### 10.4. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка цели работы.
3. Описание результатов выполнения задания:
  - скриншоты (снимки экрана), фиксирующие выполнение лабораторной работы;
  - листинги (исходный код) программ (если они есть);
  - результаты выполнения программ (текст или снимок экрана в зависимости от задания).
4. Выводы, сопоставленные с целью работы.

## Скрипт 2

### 3. Скрипт 3 (рис. 3)



The image shows a web browser window on the left and a terminal window on the right. The browser window displays a script on the website `esystem.rudn.ru`. The script is a shell script that iterates over a list of files and checks if they are in the `dev/null` directory. The terminal window shows the execution of the script, with output indicating that the script is skipping a value and continuing to test the remaining files.

```
while file=$(filelist[$i])
do
  if [ "$file" == "dev/null" ]
  then
    continue
  fi
  action
done
```

Эта программа пропускает нужное значение, но продолжает тестировать остальные.

#### 10.3. Последовательность выполнения работы

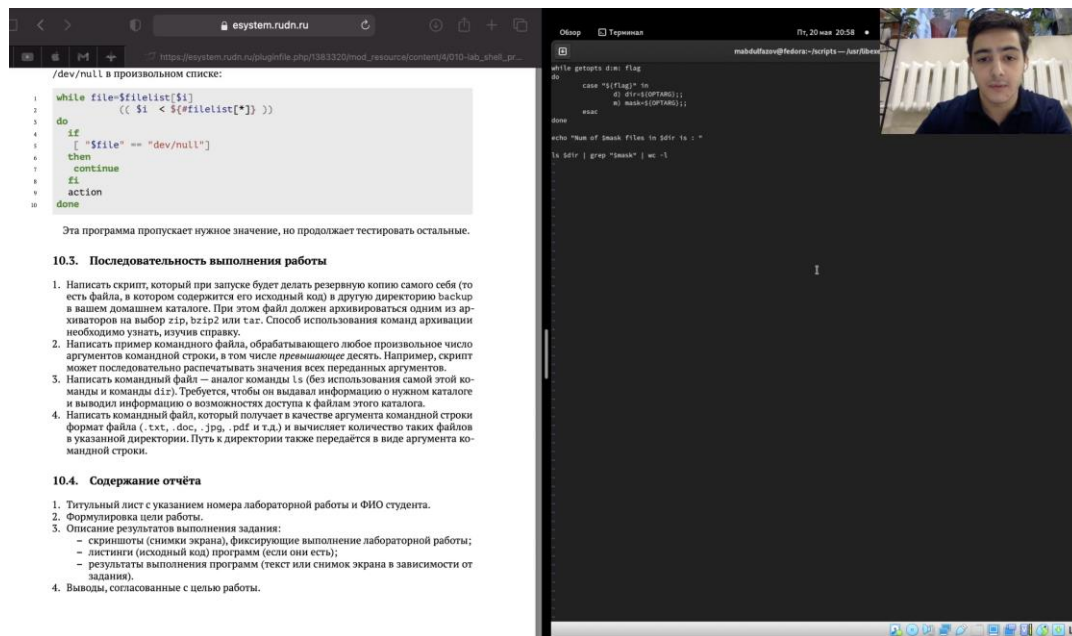
1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bz2 или cat. Способ использования команд 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 рассчитывать 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.

#### 10.4. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка цели работы.
3. Описание результатов выполнения задания:
  - скриншоты (снимки экрана), фиксирующие выполнение лабораторной работы;
  - листинги (исходный код) программ (если они есть);
  - результаты выполнения программ (текст или снимок экрана в зависимости от задания).
4. Выводы, согласованные с целью работы.

## Скрипт 1

#### 4. Скрипт 4 (рис. 4)



## Скрипт 4

# Вывод

Научился писать небольшие командные файлы.