

# Программирование в командном процессоре ОС UNIX. Командные файлы

Лабораторная работа No 10.

---

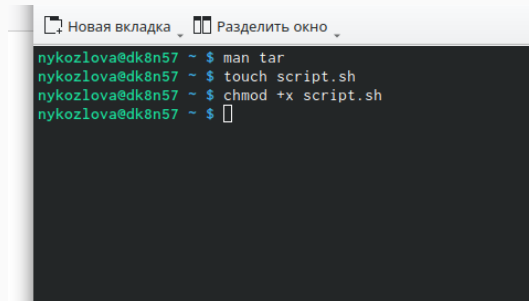
Козлова Нонна

Российский университет дружбы народов, Москва, Россия

НБИбд-04-22

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

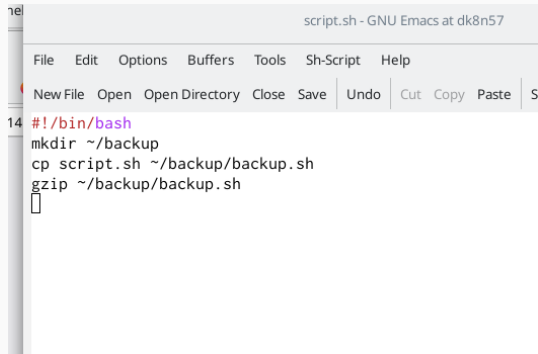
### 1. Создаю файл script и делаю его исполняемым

A terminal window with a dark background and light green text. The window has a title bar with two buttons: 'Новая вкладка' (New tab) and 'Разделить окно' (Split window). The terminal shows a series of commands and their outputs. The prompt is 'nykozlova@dk8n57 ~ \$'. The commands are 'man tar', 'touch script.sh', and 'chmod +x script.sh'. The output of 'man tar' is visible, showing the manual page for the tar command. The prompt returns after each command.

```
nykozlova@dk8n57 ~ $ man tar
nykozlova@dk8n57 ~ $ touch script.sh
nykozlova@dk8n57 ~ $ chmod +x script.sh
nykozlova@dk8n57 ~ $
```

Рис. 1: Пользуюсь командами touch и chmod

2. Пишу скрипт, который при запуске будет делать резервную копию самого себя в другую директорию backup в моем домашнем каталоге.

A screenshot of an Emacs editor window titled "script.sh - GNU Emacs at dk8n57". The window has a menu bar with "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". Below the menu bar is a toolbar with buttons for "New File", "Open", "Open Directory", "Close", "Save", "Undo", "Cut", "Copy", "Paste", and "S". The main text area shows a shell script starting with a shebang line, followed by three lines of code to create a backup directory, copy the script, and compress it. A cursor is visible at the end of the third line.

```
14 #!/bin/bash
mkdir ~/backup
cp script.sh ~/backup/backup.sh
gzip ~/backup/backup.sh
```

### 3. Запускаю файл

```
nykozlova@dk8n57 ~ $  
nykozlova@dk8n57 ~ $ ./script.sh  
nykozlova@dk8n57 ~ $
```

#### 4. Проверяю, что он сработал

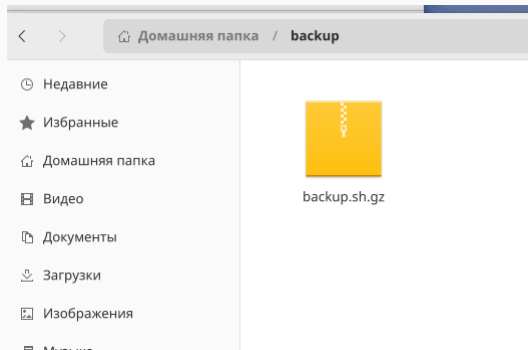
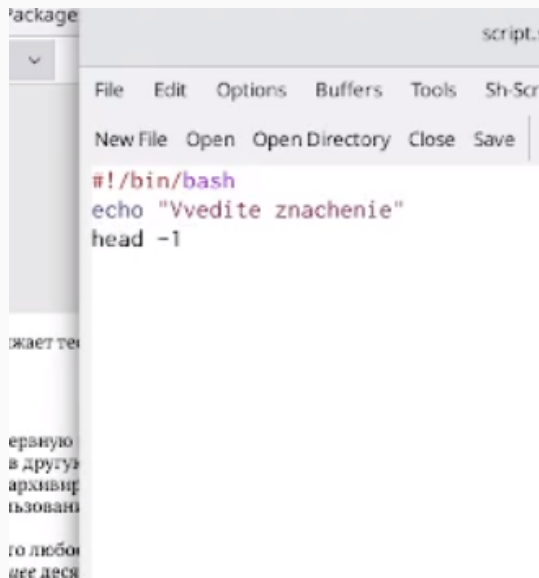


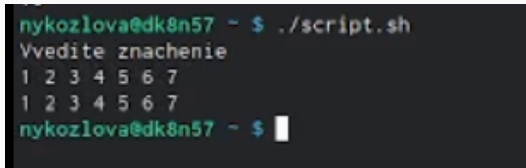
Рис. 2: работает

5. Пишу скрипт, который может последовательно распечатывать значения всех переданных аргументов



```
Package script.sh
File Edit Options Buffers Tools Sh-Scr
New File Open Open Directory Close Save
#!/bin/bash
echo "Vvedite znachenie"
head -1
```

## 6. Проверяю работу

A terminal window with a dark background. The prompt is 'nykozlova@dk8n57 ~ \$'. The user has entered './script.sh'. The output is 'Vvedite znachenie' followed by two lines of '1 2 3 4 5 6 7'. The prompt is now 'nykozlova@dk8n57 ~ \$' with a cursor.

```
nykozlova@dk8n57 ~ $ ./script.sh
Vvedite znachenie
1 2 3 4 5 6 7
1 2 3 4 5 6 7
nykozlova@dk8n57 ~ $
```

Рис. 3: работает



## 7. Пишу командный файл — аналог команды ls

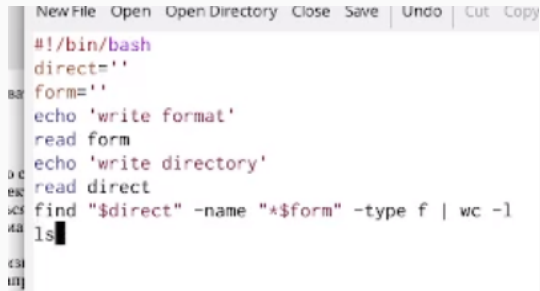
```
script.sh - GNU Emacs at dk8n57
File Edit Options Buffers Tools Sh-Script Help
New File Open Open Directory Close Save Undo Cut Copy Paste
#!/bin/bash
for A in *
do if test -d $A
then echo $A: is a directory
else echo -n $A: is a file and
if test -w $A
then echo writeable
elif test -r $A
then echo readable
else echo neither readable not writeable
fi
fi
done
```

## 8. Проверяю его работу

```
nykozlova@dk8n57 ~ $ ./script.sh
backup: is a directory
bin: is a directory
blog: is a directory
GNUstep: is a directory
KozlovaNonnaBI.github.io: is a directory
public: is a directory
public_html: is a directory
script.sh: is a file and writeable
script.sh-: is a file and writeable
tmp: is a directory
work: is a directory
Видео: is a directory
Документы: is a directory
Загрузки: is a directory
Изображения: is a directory
Музыка: is a directory
Общедоступные: is a directory
./script.sh: строка 3: test: Рабочий: ожидается бинарный оператор
Рабочий стол: is a file and ./script.sh: строка 6: test: Рабочий: ожидается бинарный оператор
./script.sh: строка 8: test: команда не найдена
neither readable nor writeable
Шаблоны: is a directory
nykozlova@dk8n57 ~ $
```

Рис. 4: работает

9. Пишу командный файл, который получает в качестве аргумента формат файла и вычисляет количество таких файлов в указанной директории.

A screenshot of a text editor window. The title bar at the top contains the text "NewFile Open OpenDirectory Close Save Undo Cut Copy". The editor area displays a shell script with the following lines: 

```
#!/bin/bash
direct=''
form=''
echo 'write format'
read form
echo 'write directory'
read direct
find "$direct" -name "$form" -type f | wc -l
ls
```

 The script prompts the user to enter a format and a directory, then counts the number of files in that directory matching the format. The cursor is positioned at the end of the last line, after "ls".

```
NewFile Open OpenDirectory Close Save Undo Cut Copy
#!/bin/bash
direct=''
form=''
echo 'write format'
read form
echo 'write directory'
read direct
find "$direct" -name "$form" -type f | wc -l
ls
```

## 10. Проверяю его работу.

```
root@kali:~/kali# cd /root/.ssh
root@kali:~/kali/.ssh# ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
root@kali:~/kali/.ssh# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAK...
root@kali:~/kali/.ssh#
```

В ходе лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать небольшие командные файлы.