

# Презентация по лабораторной работе №10

---

Подъярова Ксения Витальевна

Российский Университет Дружбы Народов

## Цель работы

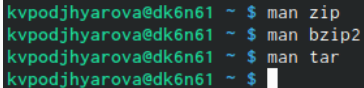
---

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

## Выполнение лабораторной работы

---

- 1) Для начала я изучила команды архивации, используя команды `man bzip2`, `man zip`, `man tar`. (рис. 1)



```
kvpodjhyarova@dk6n61 ~ $ man zip
kvpodjhyarova@dk6n61 ~ $ man bzip2
kvpodjhyarova@dk6n61 ~ $ man tar
kvpodjhyarova@dk6n61 ~ $
```

Figure 1: Команды архивации

Синтаксис команды zip для архивации файла: zip [опции] [имя файла.zip] [файлы ил папки, которые будем архивировать]. Синтаксис команды zip для разархивации файла: unzip [опции] [файл архива.zip] [файлы] -x [исключить] -d [папка] (рис. 2)

```

ZIP(1L)                                ZIP(1L)
NAME
    Zip - package and compress (archive) files

SYNOPSIS
    zip [-aABcdDdGffghjklLmqRStuvVwXyzI#s] [--longoption ...] [-b path] [-n suffixes] [-t date] [-tt date] [zipfile
    [file ...]] [-x list]

    zipcloak (see separate man page)
    zipnote (see separate man page)
    zipsplit (see separate man page)

    Note: Command line processing in zip has been changed to support long options and handle all options and arguments
    more consistently. Some old command lines that depend on command line inconsistencies may no longer work.

DESCRIPTION
    zip is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows 9x/NT/XP, Minix, Atari, Macin-
    tosh, Amiga, and Acorn RISC OS. It is analogous to a combination of the Unix commands tar(1) and compress(1) and
    is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems).

```

Figure 2: ZIP

Синтаксис bzip2 для архивации файла: bzip2 [опции][имена файлов].  
Синтаксис команды bzip2 для разархивации файла:  
bunzip2[опции][архивы.bz2] (рис. 3)

```
bzip2(1)                                General Commands Manual                                bzip2(1)

NAME
  bzip2, bunzip2 - a block-sorting file compressor, v1.0.8
  bzipcat - decompresses files to stdout
  bzip2recover - recovers data from damaged bzip2 files

SYNOPSIS
  bzip2 [ -cdfkqstzVL123456789 ] [ filenames ... ]
  bunzip2 [ -fkvsVL ] [ filenames ... ]
  bzipcat [ -s ] [ filenames ... ]
  bzip2recover filename

DESCRIPTION
  bzip2 compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.
  Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors,
  and approaches the performance of the PPM family of statistical compressors.
```

Figure 3: bzip2

Синтаксис команды tar для архивации: tar [опции][архив.tar][файлы для архивации]. Синтаксис команды для разархивации: tar [опции][архив.tar] (рис. 4)

```
TAR(1)                                GNU TAR Manual                                TAR(1)
NAME
    tar - an archiving utility

SYNOPSIS
    Traditional usage
    tar (A|c|d|r|t|u|x)[CnSkUNOMPpMlajJzZhP]RvwO [ARG...]

    UNIX-style usage
    tar -A [OPTIONS] ARCHIVE ARCHIVE
    tar -c [-f ARCHIVE] [OPTIONS] [FILE...]
    tar -d [-f ARCHIVE] [OPTIONS] [FILE...]
```

Figure 4: tar



- 2) Создала файл, в котором в будущем буду писать первый скрипт, и открыла его в редакторе emacs, используя клавиши Ctrl-x Ctrl-f
- 3) Написала скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в моем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор ( я выбрала при написании bzip2 (рис. 5)

```
#!/bin/bash
name="backup.sh"
mkdir ~/backup
bzip2 -k ${name}
mv ${name}.bz2 ~/backup/
echo "Выполнено"
```

#В переменную name сохраняем файл со скриптом  
#Создаем каталог  
#Архивируем скрипт  
#Перемещаем архивированный скрипт в каталог

Figure 5: Скрипт 1

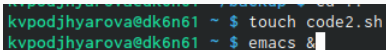
- 4) Проверила работу скрипта (команда `./backup.sh`), предварительно добавив для него права на выполнение (`chmod +x *.sh`). Проверила, появился ли каталог `backup/`, перейдя в него (команда `cd backup/`), посмотрела его содержимое (`ls`) и просмотрела содержимое архива (`bunzip2 -backup.sh.bz2`). Скрипт работает корректно (рис. 6)

```
kvpodjhyarova@dk6n61 ~ $ touch backup.sh
kvpodjhyarova@dk6n61 ~ $ emacs &
[1] 37881
kvpodjhyarova@dk6n61 ~ $ ./backup.sh
bash: ./backup.sh: Отказано в доступе
kvpodjhyarova@dk6n61 ~ $ chmod +x *.sh
kvpodjhyarova@dk6n61 ~ $ ./backup.sh
Выполнено
kvpodjhyarova@dk6n61 ~ $ cd backup/
kvpodjhyarova@dk6n61 ~/backup $ ls
backup.sh.bz2
kvpodjhyarova@dk6n61 ~/backup $ bunzip2 -c backup.sh.bz2
#!/bin/bash

name='backup.sh'
mkdir ~/backup          #В переменную name сохраняем файл со скриптом
bzip2 -k ${name}         #Создаем каталог
mv ${name}.bz2 ~/backup/ #Архивируем скрипт
echo "Выполнено"         #Перемещаем архивированный скрипт в каталог
kvpodjhyarova@dk6n61 ~/backup $
```

Figure 6: Проверка работы скрипта

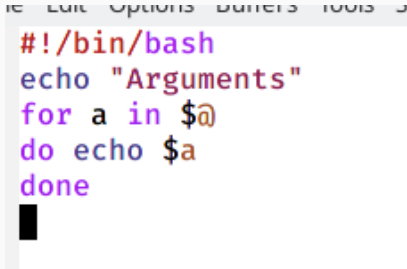
2. 1) Создала файл, в котором буду писать второй скрипт, и открыла его в редакторе emacs, используя клавиши Ctrl-x Ctrl-f (рис. 7)

A terminal window with a dark background and green text. The prompt is 'kvpodjhyarova@dk6n61 ~ \$'. The first command is 'touch code2.sh' and the second is 'emacs &'.

```
kvpodjhyarova@dk6n61 ~ $ touch code2.sh  
kvpodjhyarova@dk6n61 ~ $ emacs &
```

Figure 7: Создание файла

- 2) Написала пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее 10. Например, скрипт может последовательно распечатать значения всех переданных аргументов (рис. 8)

A screenshot of a terminal window with a light gray background. At the top, there is a menu bar with the following items: 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Shell'. Below the menu bar, the script content is displayed in a monospaced font with syntax highlighting. The script starts with a shebang line, followed by an echo statement, a for loop that iterates over all command-line arguments, and an echo statement inside the loop. The script ends with a 'done' statement. A black cursor is visible on the line following the 'done' statement.

```
#!/bin/bash
echo "Arguments"
for a in $@
do echo $a
done
```

Figure 8: Скрипт 2

- 3) Проверила работу написанного скрипта, предварительно добавив для него право на выполнение. Вводила аргументы количество которых меньше 10(рис. 9) и больше 10(рис. 10). Скрипт работает корректно

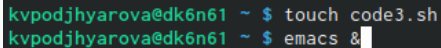
```
kupodjhyarova@kde1 ~$ chmod +x *.sh
kupodjhyarova@kde1 ~$ ls
-          backup.sh  conf.txt  lab       my.os     reports  Видео  Общедоступные
abc1      backup.sh-  feathers  lab07.sh  os-intro  ski_places  Документы  "Рабочий стол"
Architecture_PC  bin        file.txt  lab07.sh-  play      text.txt  Загрузки  Шаблоны
australia  code2.sh   fun       may       public    tap       Изображения
backup     code2.sh-  games    monthly   public_html  work      Музыка
kupodjhyarova@kde1 ~$ ./code2.sh 0 1 2 3 4
Arguments
0
1
2
3
4
```

Figure 9: Проверка работы скрипта

```
kupodjhyarova@kde1 ~$ ./code2.sh 1 2 3 0 4 5 6 7 8 9 10 11 12
Arguments
1
2
3
0
4
5
6
7
8
9
10
11
12
```

Figure 10: Больше 10 аргументов

3. 1) Создала файл, в котором буду писать третий скрипт, и открыла его в редакторе emacs (рис. 11)

A terminal window with a dark background and green text. It shows two commands being executed: 'touch code3.sh' and 'emacs &'. The prompt is 'kvpodjhyarova@dk6n61 ~ \$'.

```
kvpodjhyarova@dk6n61 ~ $ touch code3.sh
kvpodjhyarova@dk6n61 ~ $ emacs &
```

Figure 11: Создание файла

- 2) Написалаа командный файл - аналог команды ls без использования этой самой команды и команды dir). Он должен выдавать информацию о нужном каталоге и выводить информацию о возможностях доступа к файлам этого каталога (рис. 12)

```
#!/bin/bash
a="$1"
for i in ${a}/*
do
    echo "$i"
    if test -f $i
    then echo "Обычный файл"
    fi

    if test -d $i
    then echo "Каталог"
    fi

    if test -r $i
    then echo "Чтение разрешено"
    fi

    if test -w $i
    then echo "Запись разрешена"
    fi

    if test -x $i
    then echo "Выполнение разрешено"
    fi
done
```

Figure 12: Скрипт 3

- 3) Далее проверила работу скрипта, предварительно добавив для него право на выполнение. Скрипт работает корректно (рис. 13)

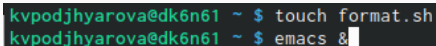
```
kvpodjhyarova@dk6n61 ~ $ touch code3.sh
kvpodjhyarova@dk6n61 ~ $ emacs &
[3] 40059
kvpodjhyarova@dk6n61 ~ $ chmod +x *.sh
kvpodjhyarova@dk6n61 ~ $ ls
-          backup.sh  code3.sh  fun      may      public      tmp      Изображения
Architecture_PC  backup.sh~  code3.sh~  games   monthly  public.html  work      Музыка
australia        bin         conf.txt  lab     my_os    reports     Видео     Общедоступные
backup           code2.sh   feathers  lab07.sh  os-intro  ski_places  Документы 'Рабочий стол'
code2.sh~        file.txt   lab07.sh~  play     text.txt  Загрузки   Шаблоны

kvpodjhyarova@dk6n61 ~ $ ./code3.sh ~
/afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova/-
Обычный файл
Чтение разрешено
Запись разрешена
/afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova/abc1
Обычный файл
Чтение разрешено
Запись разрешена
/afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova/Architecture_PC
Каталог
Чтение разрешено
Запись разрешено
Выполнение разрешено
/afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova/australia
Каталог
Чтение разрешено
Запись разрешена
Выполнение разрешено
```

Figure 13: Проверка работы скрипта



4. 1) Для четвертого скрипта создала файл и открыла его в редакторе emacs (рис. 14)



```
kvpodjhyarova@dk6n61 ~ $ touch format.sh  
kvpodjhyarova@dk6n61 ~ $ emacs &
```

A terminal window with a dark background and light green text. The first line shows the user 'kvpodjhyarova' at host 'dk6n61' in the home directory '~' running the command 'touch format.sh'. The second line shows the same user and host running 'emacs &', with a cursor visible at the end of the command.

Figure 14: Создание файла

- 2) Написала командный файл, который получает в качестве аргумента командной строки формат файла и вычисляет количество таких файлов в указанной директории. Путь к директории также передается в виде аргумента командной строки (рис. 15)

```
#!/bin/bash
b="$1"
shift
for a in $@
do
    k=0
    for i in ${b}/*.${a}
    do
        if test -f "$i"
        then
            let k=k+1
        fi
    done
    echo "$k файлов содержится в каталоге $b с разрешением $a"
done
```

Figure 15: Скрипт 4

- 3) Проверила работу написанного скрипта, предварительно добавив для него право на выполнение, а также создав дополнительные файлы с разными расширениями. Скрипт работает корректно(рис. 16)

```
kvpodjhyarova@dkén61 ~ $ touch format.sh
kvpodjhyarova@dkén61 ~ $ enacs &
[4] 41090
kvpodjhyarova@dkén61 ~ $ chmod +x *.sh
kvpodjhyarova@dkén61 ~ $ touch for.pdf for.docx for2.docx
kvpodjhyarova@dkén61 ~ $ ls
-          backup.sh  code3.sh  for2.docx  fun        may        public      tar        Изображения
abc1       backup.sh~  code3.sh~  for.docx   games      monthly   public_html  work       Музыка
Architecture_PC  bin        conf.txt  format.sh  lab        my_os     reports     Видео     Общедоступные
australia  code2.sh  feathers  format.sh~  lab07.sh  os-intro  ski_places  Документы 'Рабочий стол'
backup     code2.sh~  file.txt  for.pdf    lab07.sh~  play      text.txt    Загрузки  Шаблоны
kvpodjhyarova@dkén61 ~ $ ./format.sh - pdf txt docx
1 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova с разрешением pdf
3 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova с разрешением txt
2 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/k/v/kvpodjhyarova с разрешением docx
```

Figure 16: Проверка работы

## Выводы

---

В ходе выполнения лабораторной работы я изучила основы программирования в оболочке Linux и научилась писать небольшие командные файлы.