

Отчет по лабораторной работе №9

Шмаков Максим¹

2022, 20 мая , Москва

¹RUDN University, Moscow, Russian Federation

Цель работы

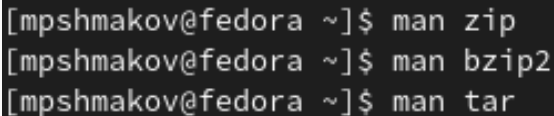
Изучить основы программирования в оболочке ОС UNIX/Linux.
Научиться писать небольшие командные файлы.

Выполнение лабораторной 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.

Изучил документацию zip, bzip2, tar (рис. [-@fig:001]) (рис. [-@fig:002])
(рис. [-@fig:003]) (рис. [-@fig:004])

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию бэкап 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.

A terminal window with a dark background and light gray text. It shows three lines of commands entered at a prompt: [mpshmakov@fedora ~]\$ man zip, [mpshmakov@fedora ~]\$ man bzip2, and [mpshmakov@fedora ~]\$ man tar.

```
[mpshmakov@fedora ~]$ man zip  
[mpshmakov@fedora ~]$ man bzip2  
[mpshmakov@fedora ~]$ man tar
```

Рис. 1: рис. 1

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию бэкап 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.

```
mpshmakov@fedora:~ — man zip
ZIP(1L)                                ZIP(1L)

NAME
    zip - package and compress (archive) files

SYNOPSIS
    zip [-aABcdDeEfFghjklLmoqrRSTuvVwXyz!@&] [--longoption ...] [-b path]
        [-n suffixes] [-t date] [-tt date] [zipfile [file ...]] [-xi list]

    zipcloak (see separate man page)


    zipnote (see separate man page)

    zipsplit (see separate man page)

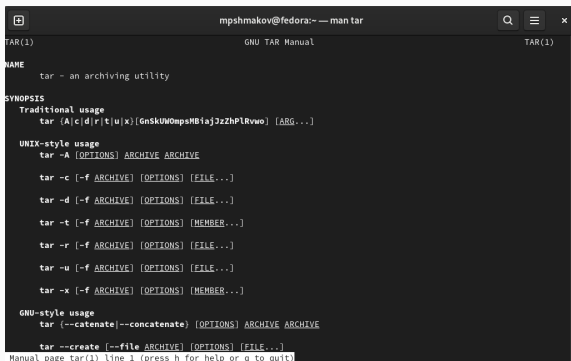
    Note: Command line processing in zip has been changed to support long
    options and handle all options and arguments more consistently. Some
    old command lines that depend on command line inconsistencies may no
    longer work.

DESCRIPTION
    zip is a compression and file packaging utility for Unix, VMS, MSDOS,
    OS/2, Windows 9x/NT/XP, Minix, Atari, Macintosh, 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).
```

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию бэкап 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.

A screenshot of a terminal window titled 'mpshmakov@fedora:~ — man bzip2'. The window displays the manual page for the 'bzip2' command. The page is divided into sections: NAME, SYNOPSIS, and DESCRIPTION. The NAME section lists 'bzip2, bunzip2 - a block-sorting file compressor, v1.0.8', 'bzipcat - decompresses files to stdout', and 'bzip2recover - recovers data from damaged bzip2 files'. The SYNOPSIS section shows the command-line syntax for 'bzip2', 'bunzip2', 'bzipcat', and 'bzip2recover'. The DESCRIPTION section explains that 'bzip2' uses the Burrows-Wheeler block sorting text compression algorithm and Huffman coding, and that its command-line options are similar to those of GNU gzip but not identical. At the bottom of the terminal window, a status bar reads 'Manual page bzip2(1) line 1 (press h for help or q to quit)'.

1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию бэкап 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.



```
mpshmakov@fedora:~ — man tar
GNU TAR Manual
TAR(1)

NAME
    tar - an archiving utility

SYNOPSIS
    Traditional usage
    tar {A|c|d|r|t|u|x}[GnSkUW0mpsMBiajJzZhPlRvw0] [ARG...]

    UNIX-style usage
    tar -A [OPTIONS] ARCHIVE ARCHIVE

    tar -c [-f ARCHIVE] [OPTIONS] [FILE...]
    tar -d [-f ARCHIVE] [OPTIONS] [FILE...]
    tar -t [-f ARCHIVE] [OPTIONS] [MEMBER...]
    tar -r [-f ARCHIVE] [OPTIONS] [FILE...]
    tar -u [-f ARCHIVE] [OPTIONS] [FILE...]
    tar -x [-f ARCHIVE] [OPTIONS] [MEMBER...]

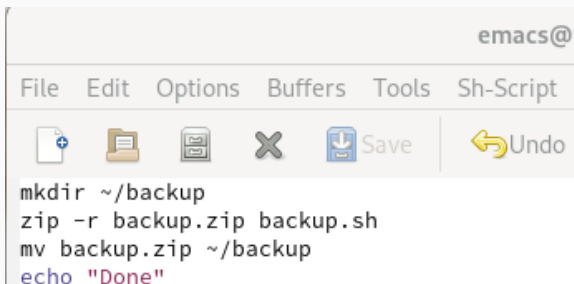
    GNU-style usage
    tar [--catenate|--concatenate] [OPTIONS] ARCHIVE ARCHIVE

    tar --create [--file ARCHIVE] [OPTIONS] [FILE...]

Manual page tar(1) line 1 (press h for help or q to quit)
```


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.

Создаю файл backup.sh и пишу в нем скрипт. Затем даю права на исполнение этому файлу и тестирую скрипт. Проверяю результат с помощью cd и ls. (рис. [-@fig:005]) (рис. [-@fig:006]) (рис. [-@fig:007])



```
emacs@1  
File Edit Options Buffers Tools Sh-Script  
[Icons: New File, Open File, Save File, Close Window, Download] Save Undo  
mkdir ~/backup  
zip -r backup.zip backup.sh  
mv backup.zip ~/backup  
echo "Done"
```

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.

A terminal window showing a command being executed. The prompt is [mpshmakov@fedora ~]\$. The command is chmod +x backup.sh.

```
[mpshmakov@fedora ~]$ chmod +x backup.sh
```

Рис. 6: рис. 6

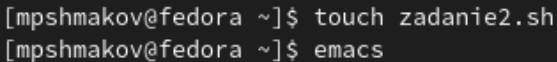
1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar. Способ использования команд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

```
[mpshmakov@fedora ~]$ ./backup.sh
  adding: backup.sh (deflated 35%)
Done
[mpshmakov@fedora ~]$ cd backup
[mpshmakov@fedora backup]$ ls
backup.zip
```

Рис. 7: рис. 7

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.

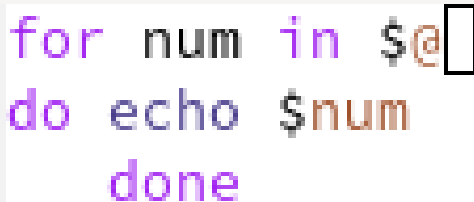
Создаю файл `zadanie2.sh` и пишу в нем скрипт. Даю права на исполнение и проверяю задав больше и меньше 10 аргументов. (рис. [-@fig:008]) (рис. [-@fig:009]) (рис. [-@fig:010])

A terminal window with a dark background and light-colored text. The prompt is [mpshmakov@fedora ~]. The first command is touch zadanie2.sh and the second command is emacs.

```
[mpshmakov@fedora ~]$ touch zadanie2.sh  
[mpshmakov@fedora ~]$ emacs
```

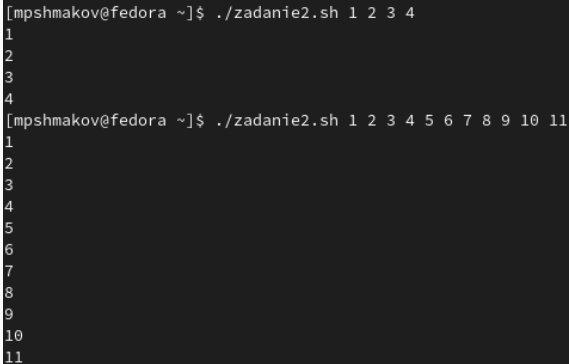
Рис. 8: рис. 8

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.



```
for num in $@  
do echo $num  
done
```

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.

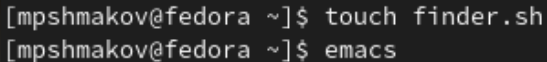
A terminal window with a dark background and light-colored text. The prompt is [mpshmakov@fedora ~]. The first command is ./zadanie2.sh 1 2 3 4, followed by the output 1, 2, 3, 4 on separate lines. The second command is ./zadanie2.sh 1 2 3 4 5 6 7 8 9 10 11, followed by the output 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 on separate lines.

```
[mpshmakov@fedora ~]$ ./zadanie2.sh 1 2 3 4
1
2
3
4
[mpshmakov@fedora ~]$ ./zadanie2.sh 1 2 3 4 5 6 7 8 9 10 11
1
2
3
4
5
6
7
8
9
10
11
```

Рис. 10: рис. 10

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.

Создаю файл `finder.sh` и пишу в нем скрипт. Даю права на исполнение и проверяю работу скрипта. (рис. [-@fig:011]) (рис. [-@fig:012]) (рис. [-@fig:013])

A terminal window with a dark background and light gray text. The prompt is [mpshmakov@fedora ~]. The first command is touch finder.sh and the second is emacs.

```
[mpshmakov@fedora ~]$ touch finder.sh  
[mpshmakov@fedora ~]$ emacs
```

Рис. 11: рис. 11

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.

```
for a in *
do if test -d $a
then echo $a: Директория
else echo -n $a: Файл и
    if test -w $a
    then echo    можно редактировать
    elif test -r $a
    then echo    можно прочитать
    elif test -w $a
    then echo    можно выполнить
    else echo    нельзя ни редактировать, ни прочитать
    fi
fi
done
```

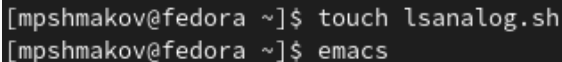
Рис. 12: рис. 12

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.

```
[mpshmakov@fedora ~]$ chmod +x lsanalog.sh
[mpshmakov@fedora ~]$ ./lsanalog.sh
abc1: Файл иможно редактировать
australia: Директория
backup: Директория
backup.sh: Файл иможно редактировать
backup.sh~: Файл иможно редактировать
bin: Директория
conf.txt: Файл иможно редактировать
Desktop: Директория
dfsdafs: Файл иможно редактировать
Documents: Директория
Downloads: Директория
feathers: Файл иможно редактировать
filetest: Директория
filetest.txt: Директория
file.txt: Файл иможно редактировать
index.html: Файл иможно редактировать
index.html.1: Файл иможно редактировать
#lab07.sh#: Файл иможно редактировать
lab07.sh: Файл иможно редактировать
```

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.

Создаю файл lsanalog.sh и пишу в нем скрипт. Даю права на исполнение, проверяю работу скрипта задав разные форматы файлов на вход. (рис. [-@fig:014]) (рис. [-@fig:015]) (рис. [-@fig:016])

A terminal window with a dark background and light-colored text. The prompt is [mpshmakov@fedora ~]. The first command is touch lsanalog.sh and the second is emacs.

```
[mpshmakov@fedora ~]$ touch lsanalog.sh  
[mpshmakov@fedora ~]$ emacs
```

Рис. 14: рис. 14

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.

```
ls $2/*$1 | wc -l
```

Рис. 15: рис. 15

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.

```
[mpshmakov@fedora ~]$ chmod +x finder.sh
[mpshmakov@fedora ~]$ ./finder.sh .zip backup
3
[mpshmakov@fedora ~]$ ./finder.sh .txt backup
2
[mpshmakov@fedora ~]$ ls backup
1.txt 2.txt backup.zip ghhhh.zip gwsekjfhskjsdh.zip
```

Рис. 16: рис. 16

Выводы

В ходе работы я изучил основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.