

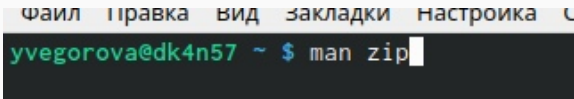
Презентация к лабораторной работе №10

Егорова Юлия Владимировна

¹RUDN University, Moscow, Russian Federation

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

1. Вводим команду `man zip`, получаем информацию о ней:

A screenshot of a terminal window. At the top, there is a menu bar with Russian text: 'Файл', 'Правка', 'Вид', 'Закладки', 'Настройка', and 'Справка'. Below the menu bar, the terminal prompt shows the username 'yvegorova@dk4n57' followed by a tilde '~' and a dollar sign '\$'. The command 'man zip' has been entered, and a white cursor is positioned at the end of the command.

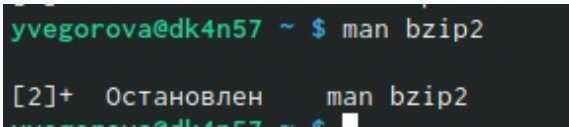
```
yvegorova@dk4n57 ~ $ man zip
```

Figure 1: Man zip.

```
zip(1)
NAME
zip - package and compress (archive) files
SYNOPSIS
zip [-subdirs][-q][-j][-mmy][-twtwstms] [--langoption ...] [-s path] [-o suffixes] [-t date] [zipfile file ...] [-w list]
ziplook (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
DESCRIPTION
zip is a compression and file packaging utility for Unix, VMS, MSDOS, OS/2, Windows 9x/NT/XP, MVS, Atari, Macintosh, Amiga, and Acorn RISC OS. It is analogous to a combination of the Unix commands tar(1) and is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems).
A compression program (unzip(1)) unpacks zip archives. The zip and unzip(1) programs can work with archives produced by PKZIP (supporting most PKZIP features up to PKZIP version 4.5), and PKZIP and PKUNZIP archives produced by zip (with some exceptions, rarely streamed archives, but recent changes in the zip file standard may facilitate better compatibility). zip version 3.0 is compatible with PKZIP 2.4x and Zinfo extensions of PKZIP 4.3 which allow archives as well as files to exceed the previous 2 GB limit (4 GB in some cases). zip also now supports bzip2 compression if the bzip2 library is installed. When built that PKUNZIP 1.0 cannot extract files produced by PKZIP 2.04 or zip 3.0, you must use PKUNZIP 2.04g or unzip 3.0g (or later versions) to extract them.
See the EXAMPLES section at the bottom of this page for examples of some typical uses of zip.
Large archives and splits
zip automatically uses the Zip64 extensions when files larger than 4 GB are added to an archive, an archive containing Zip64 entries is updated (if the resulting archive still size of the archive will exceed 4 GB, or when the number of entries in the archive will exceed about 64k). Zip64 is also used for archives streamed from standard input as the size of such archives are not 4, but the option -rvc can be used to force zip to create PKZIP 2 compatible archives (as long as Zip64 extensions are not needed). You must use a PKZIP 4.3 compatible unzip, such as unzip 3.0 or later, to extract the Zip64 extensions.
In addition, streamed archives, entries encrypted with standard encryption, or split archives created with the -ssize option may not be compatible with PKZIP as data descriptors are used and PKZIP at the time does not support data descriptors (but recent changes in the PKWare published zip standard now include some support for the data descriptor format (zip users).
Mac OS X
Though previous Mac versions had their own zip port, zip supports Mac OS X as part of the Unix port and most Unix features apply. References to "MacOS" below generally refer to MacOSX version support for some Mac OS features in the Unix Mac OS X port, such as resource forks. It is expected in the next zip release.
For a brief help on zip and unzip, run each without specifying any parameters on the command line.
BUGS
The program is useful for packaging a set of files for distribution, for archiving files, and for saving disk space by temporarily compressing unused files or directories.
The zip program puts one or more compressed files into a single zip archive, along with information about the files (owner, mode, date, time of last modification, protection, and other information to verify the entire directory structure can be packed into a zip archive with a single command. Compression ratios of 2:1 to 3:1 are common for text files. zip has one compression method (deflation) and can also do compression. (If total space is added, zip can also compress using bzip2 compression, but such entries require a reasonably modern unzip to decompress. When bzip2 compression is selected, it replaces the deflate method.)
EXAMPLES
To create a zip archive of the files in the directory named dir, use the following command. If dir is a subdirectory, use the following command to create a zip archive of the files in the directory named dir.
```

Figure 2: Информация о man zip.

2. Команду man bzip2 и также получаем информацию:

A terminal window with a dark background. The prompt is 'yvegorova@dk4n57 ~ \$'. The command 'man bzip2' has been entered. The output shows '[2]+ Остановлен man bzip2' followed by a cursor. The text is in a monospaced font with some color coding (green for the prompt, blue for the command).

```
yvegorova@dk4n57 ~ $ man bzip2  
[2]+ Остановлен man bzip2
```

Figure 3: Man bzip2.

Команда man bzip2

```

bzip2(1)                                General Commands Manual
bzip2
bzip2, bzip2c - a block-sorting file compressor, v1.0.6
bunzip2 - decompress files to which
bziprecover - recover data from damaged bzip2 files

SYNOPSIS
bzip2 [-cdvfhqzt] [-C compression] [-E filename ...]
bunzip2 [-cdvfhqzt] [-E filename ...]
bunzip2 -w compression [-E filename ...]
bziprecover 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/LZS-like, and approaches the performance of the PPM family of statistical compressors.

The command-line options are deliberately very similar to those of gzip, but they are not identical.

bzip2 expects a list of file names to accompany the command-line flags. Each file is replaced by a compressed version of itself, with the same "original.name.bz2". Each compressed file has the same metadata, and, where possible, ownership on the corresponding original, so that these properties can be correctly restored at decompression time. File name encoding is aware in the sense that there is no "serving" original file names, permissions, ownership or dates in filesystems which lack these concepts, or have serious file name length restrictions, such as MS-DOS.

bzip2 and bunzip2 will by default not overwrite existing files. If you want this to happen, specify the -f flag.

If no file names are specified, bzip2 compresses from standard input to standard output. In this case, bzip2 will decline to write compressed output to a terminal, as this would be entirely incompressible.

bzip2 [or bunzip2 -c] decompresses all specified files. Files which were not created by bzip2 will be detected and ignored, and a warning issued. bzip2 attempts to guess the filename for the decompressed file as follows:

    filename.bz2 becomes filename
    filename.bz becomes filename
    filename.tbz2 becomes filename.tar
    filename.tbz becomes filename.tar
    anyothername becomes anyothername.out

If the file does not end in one of the recognized endings, bzip2, bunzip2 or .tbz.bunzip2 complains that it cannot guess the name of the original file, and uses the original name with .bz2 appended.

As with compression, supplying no filenames causes decompression from standard input to standard output.

bzip2 will correctly decompress a file which is the concatenation of two or more compressed files. The result is the concatenation of the corresponding uncompressed files. Integrity testing (-t) of compressed files is also supported.

The -cdvfhqzt options or decompress files to the standard output by giving the -c flag. Multiple files may be uncompressed and decompressed like this. The resulting outputs are fed sequentially to other multiple files in this manner generates a stream containing multiple compressed file representations. Such a stream can be decompressed correctly only by bzip2 version 0.9.0 or later. Earlier versions of bzip2, after decompressing the first file in the stream.

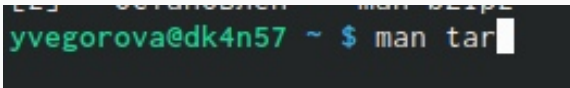
bzip2 [or bunzip2 -c] decompresses all specified files to the standard output.

bzip2 will read arguments from the environment variables BZIP2 and BUNZIP2, in that order, and will process them before any arguments read from the command line. This gives a convenient way to supply default compression. It always performs, even if the compressed file is slightly larger than the original. Files of less than one hundred bytes tend to get larger, since the compression mechanism has a 100-byte overhead. The compression ratio is about 2:1 for most files, and 3:1 for text files. The compression ratio is about 2:1 for most files, and 3:1 for text files. The compression ratio is about 2:1 for most files, and 3:1 for text files.

```

Figure 4: Информация о man bzip2.

3. Команду man tar и получаем информацию:

A terminal window with a dark background. The prompt 'yvegorova@dk4n57 ~ \$' is shown in green. The command 'man tar' is entered in white, followed by a white cursor block.

```
yvegorova@dk4n57 ~ $ man tar
```

Figure 5: Man tar.

Команда man tar

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

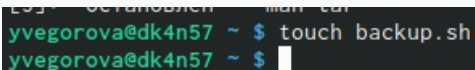
SYNOPSIS
    Traditional usage
    tar [A|c|d|r|t|u|x][G|nSk|W|O|p|s|M|B|a|j|zZ|P|I|H|vao] [ARG...]

    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...]
    tar [--diff|--compare] [--file ARCHIVE] [OPTIONS] [FILE...]
    tar --delete [--file ARCHIVE] [OPTIONS] [MEMBER...]
    tar --append [-f ARCHIVE] [OPTIONS] [FILE...]
    tar --list [-f ARCHIVE] [OPTIONS] [MEMBER...]
    tar --test-label [--file ARCHIVE] [OPTIONS] [LABEL...]
    tar --update [--file ARCHIVE] [OPTIONS] [FILE...]
    tar --update [-f ARCHIVE] [OPTIONS] [FILE...]
```

Figure 6: Информация о man tar.

4. Создала файл backup.sh:

A terminal window with a dark background. The prompt is 'yvegorova@dk4n57 ~ \$'. The command 'touch backup.sh' has been entered and executed. The prompt is now 'yvegorova@dk4n57 ~ \$' followed by a white cursor block.

```
yvegorova@dk4n57 ~ $ touch backup.sh  
yvegorova@dk4n57 ~ $
```

Figure 7: Создание файла.

5. Открыла emacs:

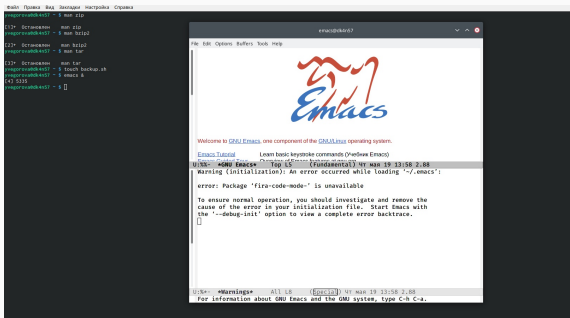
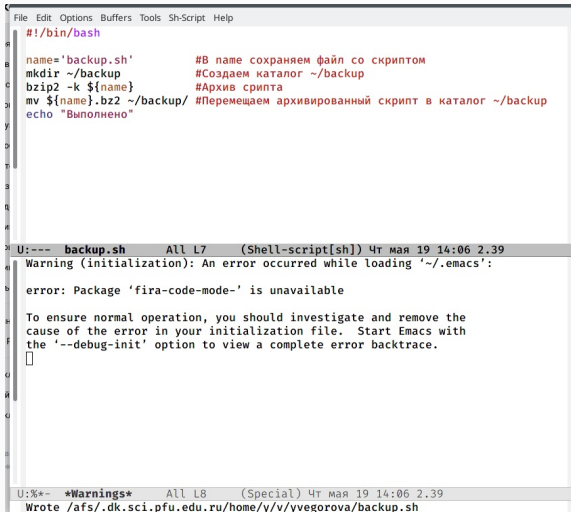


Figure 8: Открытие emacs.

6. По клавишам C-x C-f нашла созданный файл и ввела следующий текст:



The screenshot shows an Emacs editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a text area containing a shell script. The script is as follows:

```
#!/bin/bash

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

Below the script, a status bar indicates the file is 'backup.sh' and the mode is 'Shell-script[sh]'. A warning message is displayed:

```
U:--- backup.sh All L7 (Shell-script[sh]) Чт мая 19 14:06 2.39
Warning (initialization): An error occurred while loading '~/.emacs':

error: Package 'fira-code-mode-' is unavailable

To ensure normal operation, you should investigate and remove the
cause of the error in your initialization file. Start Emacs with
the '--debug-init' option to view a complete error backtrace.
□
```

The status bar at the bottom shows the current buffer is 'U:~*- *Warnings*' and the mode is '(Special)'. The file path is '/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/backup.sh'.

7. Через клавиши C-x C-s сохранила наш файл и проверила в консоли через следующие команды:

```
backup.sh 1022700
yvegorova@dk4n57 ~ $ chmod +x *.sh
yvegorova@dk4n57 ~ $ ./backup.sh
Выполнено
yvegorova@dk4n57 ~ $ █
```

Figure 10: Проверка.

```
yvegorova@dk4n57 ~ $ cd backup/  
yvegorova@dk4n57 ~/backup $ ls  
backup.sh.bz2  
yvegorova@dk4n57 ~/backup $
```

Figure 11: Проверка.


```
yvegorova@dk4n57 ~/backup $ bunzip2 -c backup.sh.bz2
#!/bin/bash

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

yvegorova@dk4n57 ~/backup \$

Figure 12: Проверка.

8. Создала файл prog2.sh и перешла в emacs:



```
yvegorova@dk4n57 ~ $ cd /backup  
yvegorova@dk4n57 ~ $ touch prog2.sh  
yvegorova@dk4n57 ~ $ emacs &
```

Figure 13: Создание файла и переход в emacs.

9. По клавишам C-x C-f нашла созданный файл и ввела следующий текст:



Figure 14: Поиск.


```
#!/bin/bash
echo "Аргументы"
for a in $@ #Цикл для прохода по введенным аргументам
do echo $a #Вывод аргумента
done
```

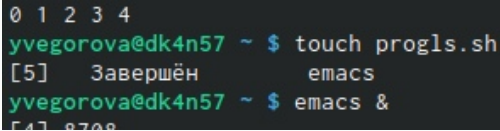
Figure 15: Заполнение.

10. Через клавиши C-x C-s сохранила наш файл и проверила в консоли через команду:

```
yvegorova@dk4n57 ~ $ ./prog2.sh 0 1 2 3 4
Аргументы
0 1 2 3 4
0 1 2 3 4
0 1 2 3 4
0 1 2 3 4
0 1 2 3 4
yvegorova@dk4n57 ~ $
```

Figure 16: Проверка.

11. Создала файл progl.s.sh и перешла в emacs:

A terminal window with a dark background and light-colored text. The prompt is 'yvegorova@dk4n57 ~'. The user enters 'touch progl.s.sh'. The prompt changes to '[5] Завершён emacs'. The user enters 'emacs &'. The prompt returns to 'yvegorova@dk4n57 ~'.

```
0 1 2 3 4
yvegorova@dk4n57 ~ $ touch progl.s.sh
[5] Завершён emacs
yvegorova@dk4n57 ~ $ emacs &
[47] 8708
```

Figure 17: Создание файла и переход в emacs.

12. Ввела текст:

```
#!/bin/bash
a="$1"          #В переменную а сохраняем путь до заданного каталога
for i in ${a}/*  #Цикл, который проходит по всем каталогам и файлам
do
    echo "$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
done
```

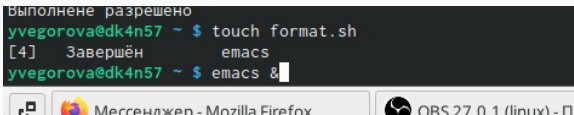
Figure 18: Заполнение файла текстом.

13. Осуществила проверку:

```
backup.sh course-directory-student-template 1000000000 1100000000 1100000000 1100000000 1000000000
yvegorova@dk4n57 ~ $ ./proglis.sh ~
/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/-
Обычный файл
Чтение разрешено
Запись разрешена
/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/backup
Каталог
Чтение разрешено
Запись разрешена
Выполнено разрешено
/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/backup.sh
Обычный файл
Чтение разрешено
Запись разрешена
Выполнено разрешено
/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/backup.sh~
Обычный файл
Чтение разрешено
Запись разрешена
/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/bin
Каталог
Чтение разрешено
Запись разрешена
Выполнено разрешено
/afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/course-directory-student-template
Каталог
Чтение разрешено
```

Figure 19: Проверка.

14. Снова создала файл под названием `format.sh` и перешла в `emacs`:

A terminal window with a dark background. The text shows a user named yvegorova@dk4n57 creating a file named format.sh using the 'touch' command. The prompt changes to [4] indicating a subshell, and then the user runs 'emacs &' to launch the Emacs editor. The terminal window is part of a desktop environment with a taskbar at the bottom showing icons for a file manager, a messenger, and a terminal window titled 'ORS 27 0 1 (linux) - П'.

```
Выполнение разрешено
yvegorova@dk4n57 ~ $ touch format.sh
[4]   Завершён      emacs
yvegorova@dk4n57 ~ $ emacs &
```

Figure 20: Создание файла и переход в `emacs`.

15. Нашла созданный файл и ввела текст:

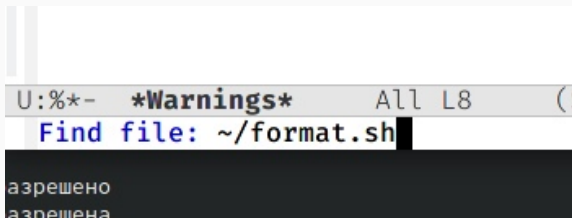


Figure 21: Поиск.

Работа с файлом в емакс

```
#!/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
```

U:--- **format.sh** All L16 (Shell-script[sh]) 4т мая 19 14:38 2.04
Warning (initialization): An error occurred while loading '~/.emacs':

error: Package 'fira-code-mode-' is unavailable

To ensure normal operation, you should investigate and remove the
cause of the error in your initialization file. Start Emacs with
the '--debug-init' option to view a complete error backtrace.
□

U:%*- ***Warnings*** All L8 (Special) 4т мая 19 14:38 2.04
Wrote /afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova/format.sh

Figure 22: Текст.

16. Ввела следующую команду и выполнила проверку:

```
backup.sh course-directory-student-template Fail4.txt File2.txt~ File4.txt File.txt~ 'id ~un' my_os prog2.sh
yvegorova@dk4n57 ~$ chmod tx *.sh
yvegorova@dk4n57 ~$ touch file.pdf file.doc file2.doc
yvegorova@dk4n57 ~$ ls
- backup.sh~ Fail12.txt Fail1.txt File2.txt~ File4.txt file.pdf format.sh lab07.sh
backup bin Fail13.txt File2.doc File3.txt File4.txt~ File.txt format.sh~ lab07.sh~
yvegorova@dk4n57 ~$ ./format.sh ~ pdf sh txt doc
yvegorova@dk4n57 ~$
1 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova с разрешением pdf
5 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova с разрешением sh
9 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova с разрешением txt
2 файлов содержится в каталоге /afs/.dk.sci.pfu.edu.ru/home/y/v/yvegorova с разрешением doc
yvegorova@dk4n57 ~$
```

Figure 23: Команда и проверка.

В ходе выполнения лабораторной работы я познакомилась с операционной системой Linux. Получила практические навыки работы с редактором vi, установленным по умолчанию практически во всех дистрибутивах