

lab10

19 May, 2022 Moscow, Russia

Цель работы

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

Задача

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 распечатывать 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.

Выполнение лабораторной работы

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.

1. Я изучила справку о zip, bzip2, tar



```
epmitichkina@fedora:~  
[epmitichkina@fedora ~]$ man zip  
[epmitichkina@fedora ~]$ man bzip2  
[epmitichkina@fedora ~]$ man tar  
[epmitichkina@fedora ~]$
```

```
epmitichkina@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).

    A companion program (unzip(1L)) unpacks zip archives. The zip and un-
zip(1L) programs can work with archives produced by PKZIP (supporting
    most PKZIP features up to PKZIP version 4.6), and PKZIP and PKUNZIP can
    work with archives produced by zip (with some exceptions, notably
    streamed archives, but recent changes in the zip file standard may fa-
    cilitate better compatibility). zip version 3.0 is compatible with
    PKZIP 2.04 and also supports the Zip64 extensions of PKZIP 4.5 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 included when zip is compiled. Note that PKUNZIP 1.10 can-
    not extract files produced by PKZIP 2.04 or zip 3.0. You must use PKUN-
    ZIP 2.04g or unzip 5.0p1 (or later versions) to extract them.
```

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

```
epmitichkina@fedora:~ — man bzip2
bzip2(1)                                General Commands Manual                                bzip2(1)

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

SYNOPSIS
    bzip2 [ -cdfkqstvzVL123456789 ] [ filenames ... ]
    bunzip2 [ -fkvsVL ] [ filenames ... ]
    bzcat [ -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.

    The command-line options are deliberately very similar to those of GNU 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 name "original_name.bz2". Each compressed file has the same modification date, permissions, and, when possible, ownership as the corresponding original, so that these properties can be correctly restored at decompression time. File name handling is naive in the sense that there is no mechanism for preserving original file names, permissions, ownerships 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 incomprehensible and therefore pointless.

    bunzip2 (or bzip2 -d) decompresses all specified files. Files which were not
```

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

NAME
    tar - an archiving utility

SYNOPSIS
    Traditional usage
        tar {A|c|d|r|t|u|x}[GnSkUW0mpsMBiajJzZhPlRvwo] [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...]

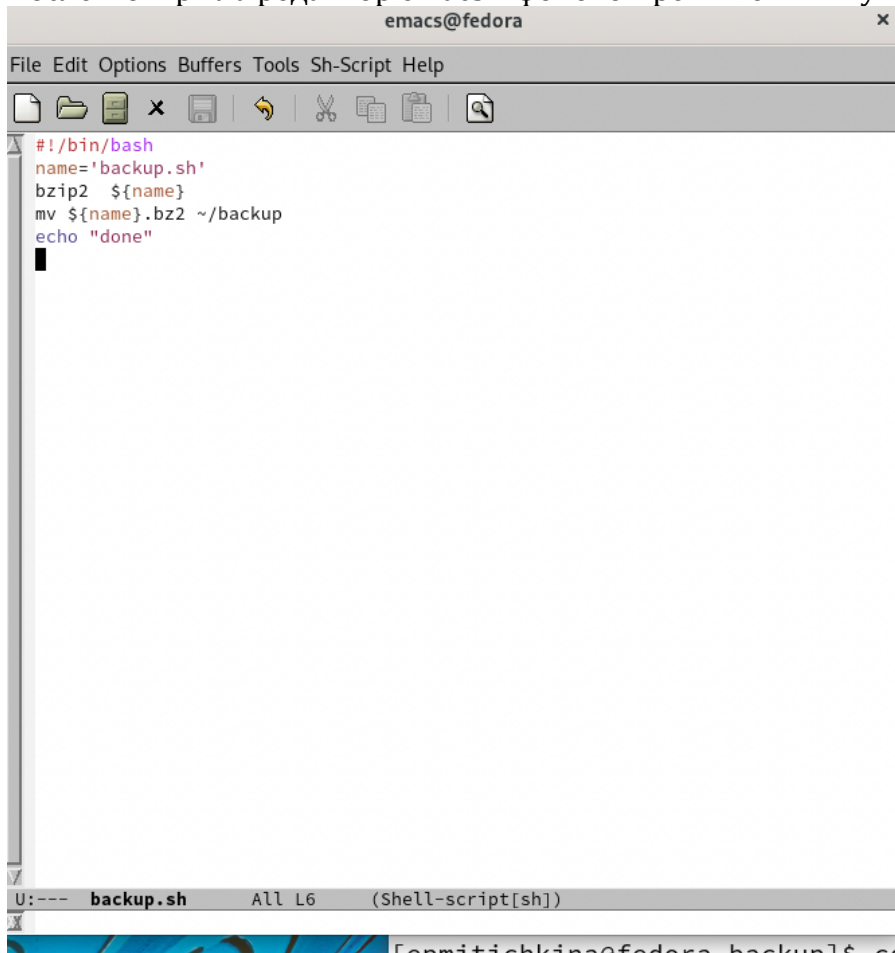
        tar --test-label [--file ARCHIVE] [OPTIONS] [LABEL...]

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

2. Создала директорию backup и файл backup.sh. Также проверила их наличие

```
[epmitichkina@fedora ~]$ mkdir backup
[epmitichkina@fedora ~]$ ls
01      conf.txt  text.txt  Документы  Музыка      Шаблоны
5       file.txt   work      Загрузки   Общедоступные
backup  montly      Видео     Изображения 'Рабочий стол'
[epmitichkina@fedora ~]$ 
[epmitichkina@fedora ~]$ touch backup.sh
[epmitichkina@fedora ~]$ ls
01      backup.sh  montly      Видео     Изображения 'Рабочий стол'
5       conf.txt  text.txt   Документы  Музыка      Шаблоны
backup  file.txt   work      Загрузки   Общедоступные
[epmitichkina@fedora ~]$
```

3. После я открыла редактор emacs в фоновом режиме и пишу код



```
#!/bin/bash
name='backup.sh'
bzip2 ${name}
mv ${name}.bz2 ~/backup
echo "done"
```

4. Предоставила право на выполнение и проверка файла

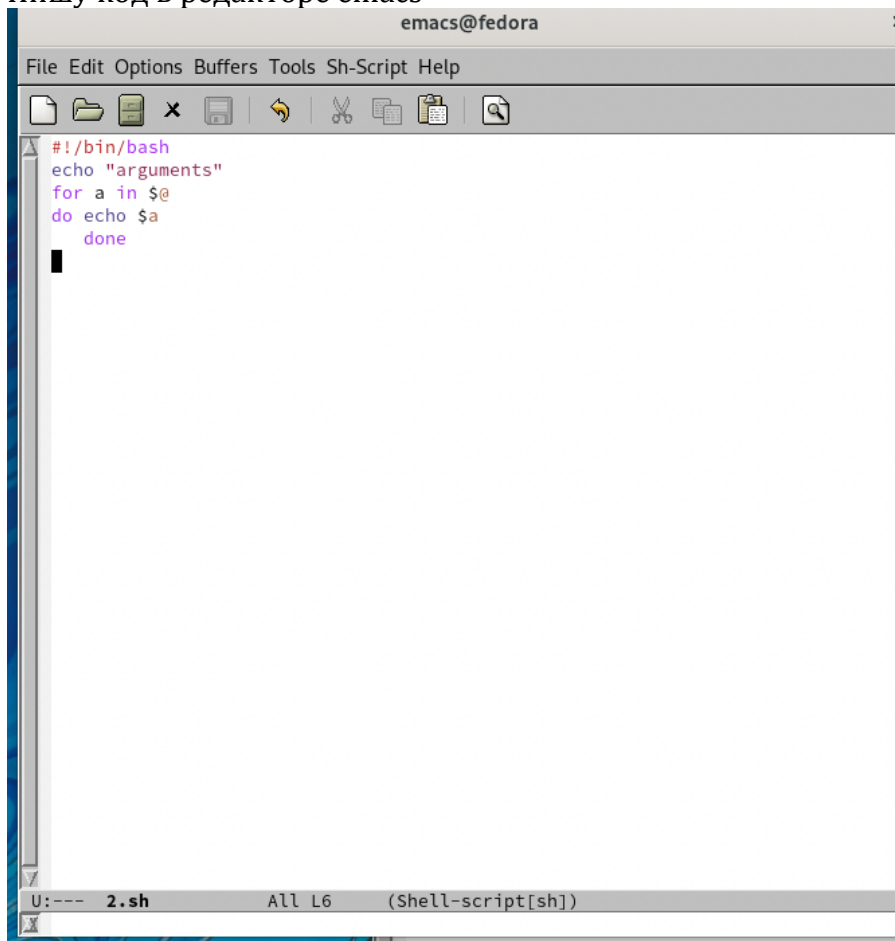
```
[epmitichkina@fedora ~]$ ./backup.sh
done
[epmitichkina@fedora ~]$ cd backup/
[epmitichkina@fedora backup]$ ls
backup.sh.bz2
[epmitichkina@fedora backup]$
```

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.

1. Создаю файл 2.sh и проверяю

```
[epmitichkina@fedora ~]$ touch 2.sh
[epmitichkina@fedora ~]$ ls
01 backup file.txt work Загрузки Общедоступные
2.sh backup.sh~ montly Видео Изображения 'Рабочий стол'
5 conf.txt text.txt Документы Музыка Шаблоны
[epmitichkina@fedora ~]$
```

2. Пишу код в редакторе emacs

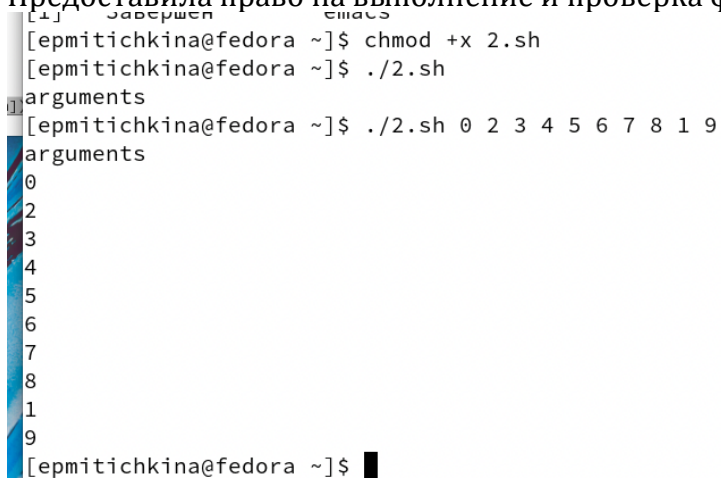


The screenshot shows the Emacs editor window titled 'emacs@fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for file operations and editing. The main text area contains a shell script with the following content:

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

The status bar at the bottom indicates the current file is '2.sh', the cursor is at line 6, and the buffer is '(Shell-script[sh])'.

3. Предоставила право на выполнение и проверка файла



The screenshot shows a terminal window with the following commands and output:

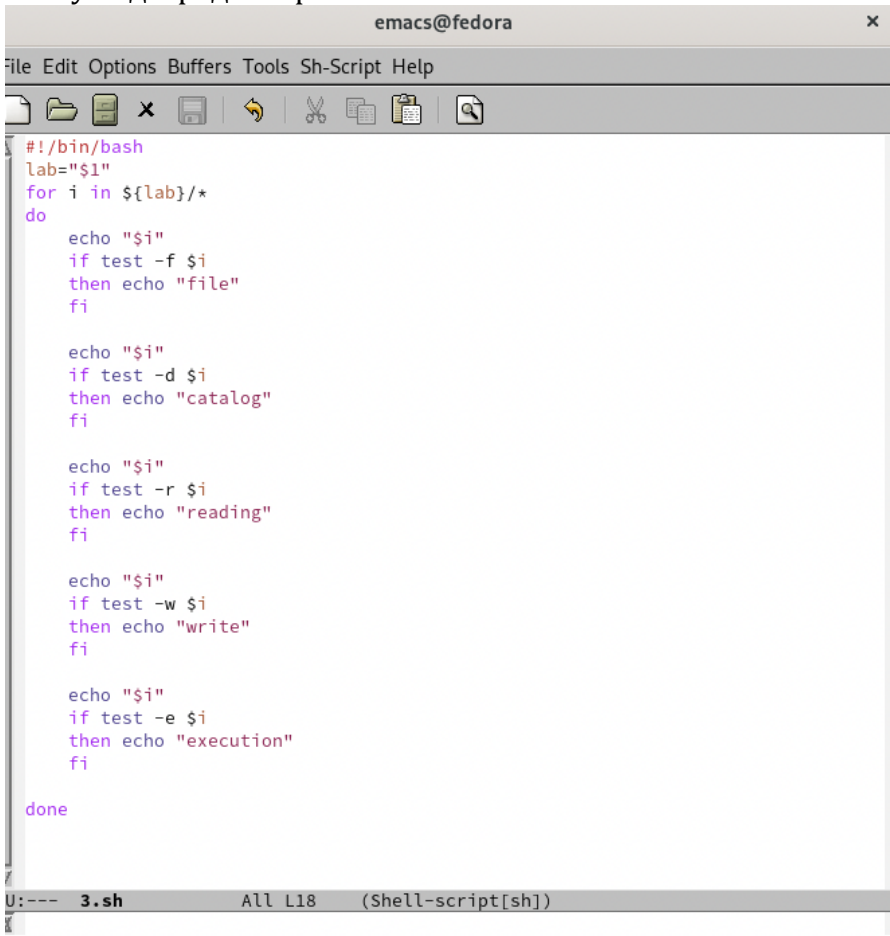
```
[epmitichkina@fedora ~]$ chmod +x 2.sh
[epmitichkina@fedora ~]$ ./2.sh
arguments
[epmitichkina@fedora ~]$ ./2.sh 0 2 3 4 5 6 7 8 1 9
arguments
0
2
3
4
5
6
7
8
1
9
[epmitichkina@fedora ~]$
```

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.

1. Создаю файл 3.sh и проверяю

```
[epmitichkina@fedora ~]$ touch 3.sh
[epmitichkina@fedora ~]$ ls
01      3.sh      backup.sh~  montly      Видео      Изображения  'Рабочий сто
2.sh    5          conf.txt   text.txt    Документы  Музыка      Шаблоны
2.sh~   backup     file.txt   work        Загрузки   Общедоступные
[epmitichkina@fedora ~]$
```

2. Пишу код в редакторе emacs



The screenshot shows the emacs editor window titled 'emacs@fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for file operations. The script content is as follows:

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

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

    echo "$i"
    if test -r $i
    then echo "reading"
    fi

    echo "$i"
    if test -w $i
    then echo "write"
    fi

    echo "$i"
    if test -e $i
    then echo "execution"
    fi
done
```

The status bar at the bottom shows 'U:--- 3.sh All L18 (Shell-script[sh])'.

3. Предоставила право на выполнение и проверка файла

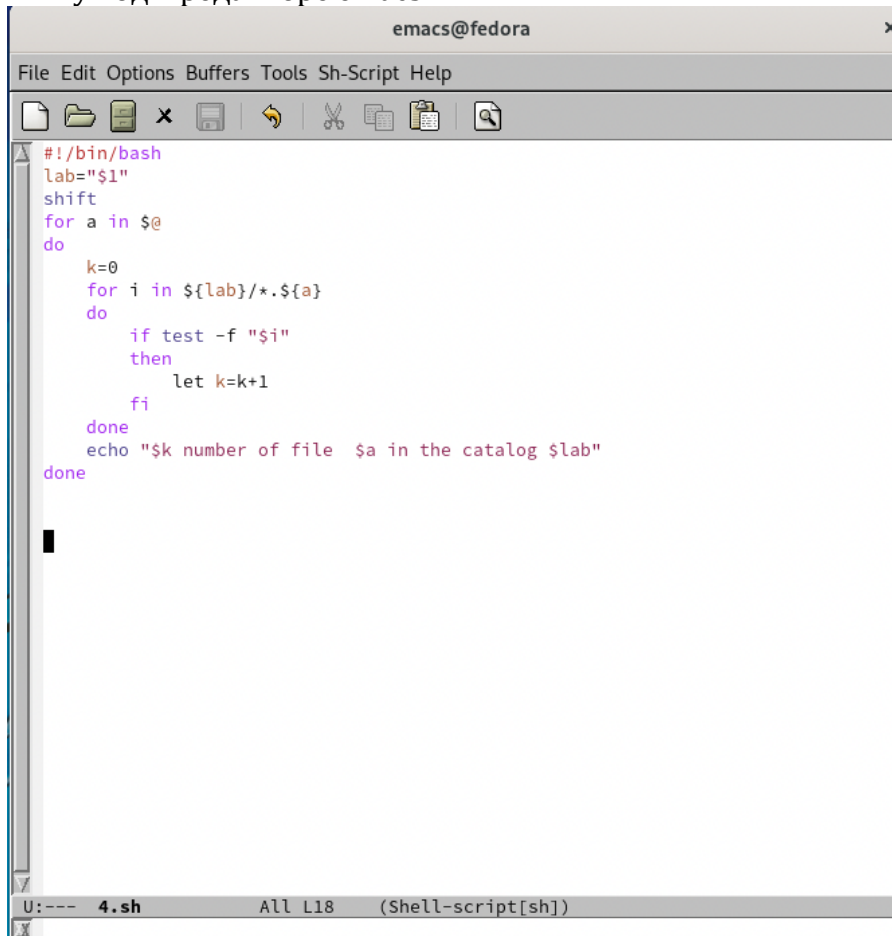
```
[epmitichkina@fedora ~]$ chmod +x 3.sh
[epmitichkina@fedora ~]$ ./3.sh ~
/home/epmitichkina/01
/home/epmitichkina/01
catalog
/home/epmitichkina/01
reading
/home/epmitichkina/01
write
/home/epmitichkina/01
execution
/home/epmitichkina/2.sh
file
/home/epmitichkina/2.sh
/home/epmitichkina/2.sh
reading
/home/epmitichkina/2.sh
write
/home/epmitichkina/2.sh
execution
/home/epmitichkina/2.sh~
file
/home/epmitichkina/2.sh~
/home/epmitichkina/2.sh~
reading
/home/epmitichkina/2.sh~
write
/home/epmitichkina/2.sh~
```

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.

1. Создаю файл 4.sh и проверяю

```
[epmitichkina@fedora ~]$ touch 4.sh
[2]- Завершён      emacs
[3]+ Завершён      emacs
[epmitichkina@fedora ~]$ ls
01      3.sh    5      conf.txt  text.txt  Документы  Музыка      Шаблоны
2.sh    3.sh~   backup  file.txt  work      Загрузки   Общедоступные
2.sh~   4.sh    backup.sh~  montly   Видео     Изображения 'Рабочий стол'
```


2. Пишу код в редакторе emacs



The screenshot shows the Emacs editor window titled 'emacs@fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for file operations and editing. The main text area displays a shell script with the following content:

```
#!/bin/bash
lab="$1"
shift
for a in $@
do
    k=0
    for i in ${lab}/*.${a}
    do
        if test -f "$i"
        then
            let k=k+1
        fi
    done
    echo "$k number of file  $a in the catalog $lab"
done
```

The status bar at the bottom indicates the file is '4.sh', it is 'All' buffer, 'L18' lines, and the mode is '(Shell-script[sh])'.

3. Предоставила право на выполнение и проверка файла

```
[epmitichkina@fedora ~]$ chmod +x 4.sh
[epmitichkina@fedora ~]$ ./4.sh ~ sh
3 number of file  sh in the catalog /home/epmitichkina
[epmitichkina@fedora ~]$
```

Выводы

В результате работы изучила основы программирования в оболочке ОС UNIX/Linux. Научилась писать небольшие командные файлы.