

Лабораторная работа No 10. Программирование процессоре ОС UNIX. Командные файлы

Leysan R. Abdullina

NEC-2022, 18 May

RUDN University, Moscow, Russian Federation

Лабораторная работа No 10.
Программирование процессоре ОС
UNIX. Командные файлы

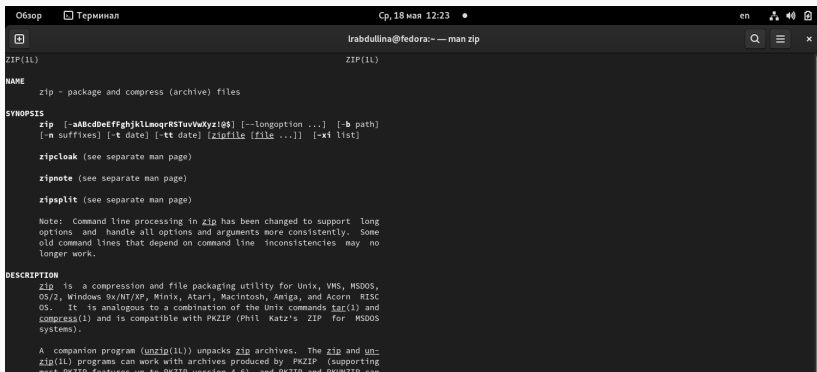
Изучить основы программирования в оболочке ОС UNIX/Linux.
Научиться писать небольшие командные файлы. Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: - оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; - C-оболочка (или csh)—надстройка на оболочке Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд;

- оболочка Корна (или ksh)—напоминает оболочку C,но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна),в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation). POSIX (Portable Operating System Interface for Computer Environments)— набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны на базе оболочки Корна.

Выполнение работы

1. Напишем скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в нашем домашнем каталоге. При этом файл будет архивироваться bzip2. Способ использования команд архивации zip, bzip2, tar узнаем,изучив справку. (скриншоты 1,2,3)



```
Обзор Терминал Cp, 18 мая 12:23 en
lrabduлина@fedora:~ — man zip

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

SYNOPSIS
zip [-aABcdDeEffghjklLmoqrRSTuvVwXyz!@#] [--longoption ...] [-b path]
[-n suffixes] [-t date] [-tt date] [zipfile [file ...]] [-x! list]

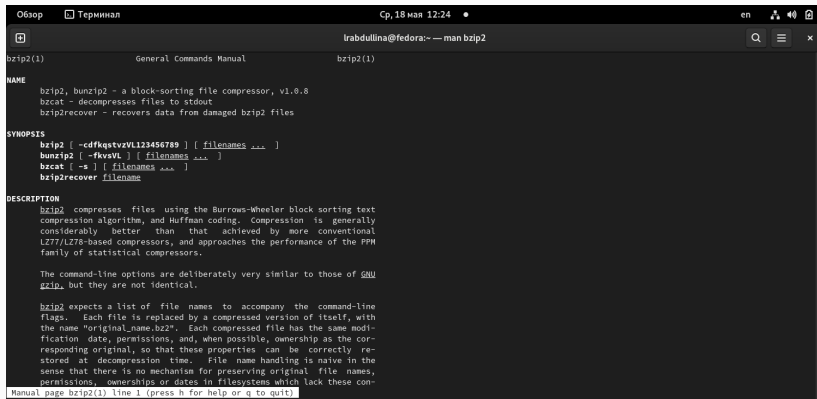
zipcloak (see separate man page)
zipnote (see separate man page)
zipsplit (see separate man page)

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, 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, but not all, PKZIP archives as seen in the "BUGS" section of the
unzip(1L) manual page).
```

Выполнение работы



The screenshot shows a terminal window with a dark background. At the top, there's a title bar with 'Обзор' (Overview) and 'Терминал' (Terminal) buttons, a clock showing 'Ср, 18 мая 12:24', and system icons. Below the title bar, the terminal title is 'lrabdullina@fedora:~ — man bzip2'. The main content is the manual page for 'bzip2(1)', titled 'General Commands Manual'. The page is divided into sections: 'NAME', 'SYNOPSIS', and 'DESCRIPTION'. The 'NAME' section lists 'bzip2', 'bunzip2', 'bzip2recover', and 'bzip2recover'. The 'SYNOPSIS' section shows the command-line syntax for each. The 'DESCRIPTION' section explains the compression algorithm and provides details about command-line options and file handling. At the bottom, a footer line reads 'Manual page bzip2(1) line 1 (press h for help or q to quit)'.

```
Обзор Терминал Ср, 18 мая 12:24 en
lrabdullina@fedora:~ — man bzip2

bzip2(1) General Commands Manual bzip2(1)

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

SYNOPSIS
  bzip2 [ -cdfkqstvzVL123456789 ] [ filenames ... ]
  bunzip2 [ -fkvsVL ] [ filenames ... ]
  bzip2recover [ -s ] [ filenames ... ]
  bzip2recover filename

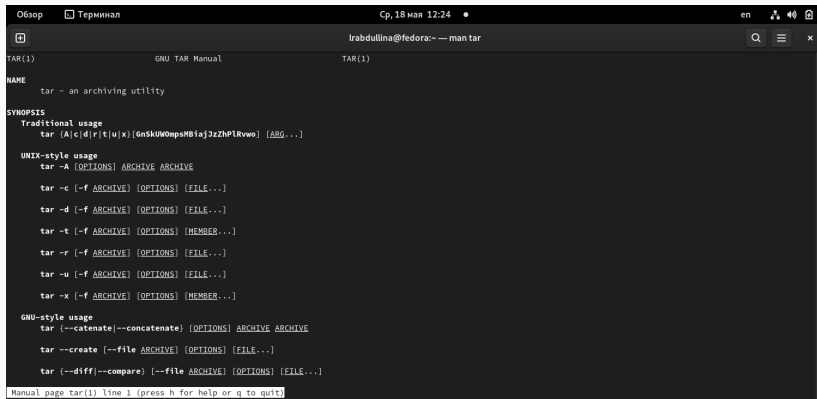
DESCRIPTION
  bzip2 compresses files using the Burrows-Wheeler block sorting text
  compression algorithm, and Huffman coding. Compression is generally
  considerably better than that achieved by more conventional
  LZ77/LZ78-based compressors, and approaches the performance of the PPM
  family of statistical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

  The command-line options are deliberately very similar to those of 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 mod-
  ification date, permissions, and, when possible, ownership as the cor-
  responding original, so that these properties can be correctly re-
  stored 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 con-
  Manual page bzip2(1) line 1 (press h for help or q to quit)
```

Figure 2: Справка об bzip2

Выполнение работы



```
Обзор Терминал                               Ср, 18 мая 12:24 • en
lrabdullina@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][GnSkUW0mpsMBiaJzZhPLRvw0] [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...]

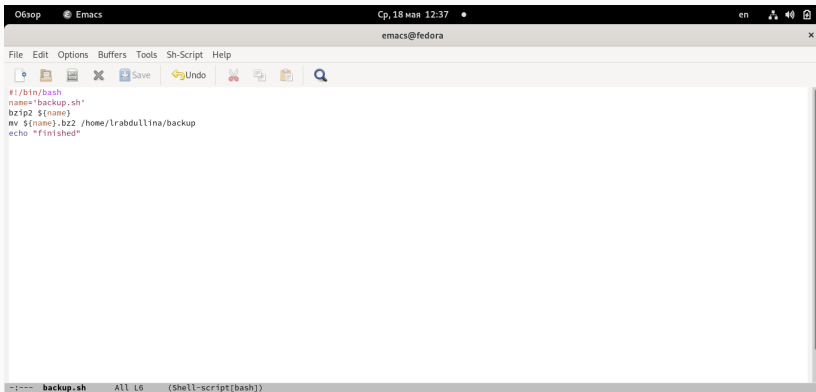
  GNU-style usage
  tar [--catenate|--concatenate] [OPTIONS] ARCHIVE ARCHIVE
  tar --create [--file ARCHIVE] [OPTIONS] [FILE...]
  tar [--diff|--compare] [--file ARCHIVE] [OPTIONS] [FILE...]

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

Figure 3: Справка об tar

Выполнение работы

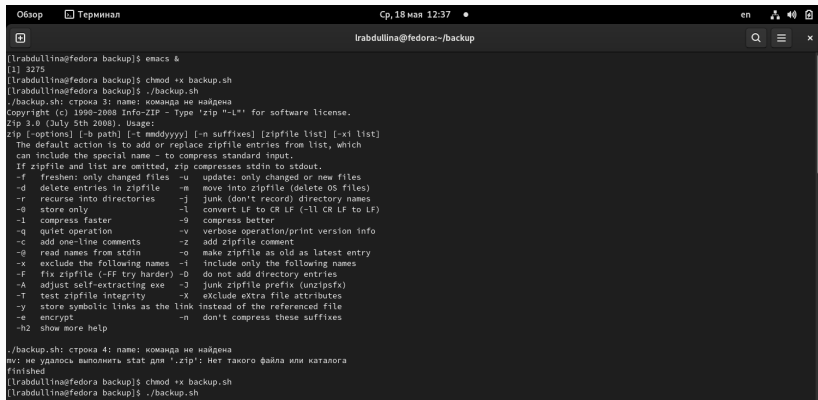
Для начала мы создадим директорию backup с помощью команды `mkdir`, далее - файл `backup.sh` через команду `touch`. После этого мы открываем редактор `emacs` в фоновом режиме (для этого ставим в конце команды знак амперсанда), открываем наш новосозданный файл и пишем программу. (скриншоты 4,5)



The screenshot shows the Emacs editor interface. The title bar at the top indicates the window is titled "Обзор" (Overview) and "Emacs", with the date and time "Ср, 18 мая 12:37" (Wed, 18 May 12:37) and the language "en". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for file operations like "Save", "Undo", and "Find". The main text area displays the following shell script:

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

The status bar at the bottom shows the current file is "backup.sh", the line number is "All 16", and the shell is "(Shell-script(bash))".



```
Обзор Терминал Ср, 18 мая 12:37 en
lrabdullina@fedora:~/backup

[lrabdullina@fedora backup]$ emacs &
[1] 3275
[lrabdullina@fedora backup]$ chmod +x backup.sh
[lrabdullina@fedora backup]$ ./backup.sh
./backup.sh: строка 3: name: команда не найдена
Copyright (c) 1999-2008 Info-ZIP - Type 'zip -L' for software license.
Zip 3.0 (July 5th 2008). Usage:
zip [-options] [-b path] [-t mddyyyyy] [-n suffixes] [zipfile list] [-x list]
The default action is to add or replace zipfile entries from list, which
can include the special name - to compress standard input.
If zipfile and list are omitted, zip compresses stdin to stdout.
-f freshen: only changed files      -u update: only changed or new files
-d delete entries in zipfile        -m move into zipfile (delete OS files)
-r recurse into directories         -j junk (don't record) directory names
-o store only                       -l convert LF to CR LF (-ll CR LF to LF)
-l compress faster                  -9 compress better
-q quiet operation                  -v verbose operation/print version info
-c add one-line comments            -z add zipfile comment
@ read names from stdin             -o make zipfile as old as latest entry
-x exclude the following names      -i include only the following names
-F fix zipfile (-FF try harder)     -D do not add directory entries
-A adjust self-extracting exe       -J junk zipfile prefix (unzipsfx)
-T test zipfile integrity           -X exclude extra file attributes
-y store symbolic links as the link instead of the referenced file
-e encrypt                          -n don't compress these suffixes
-h2 show more help

./backup.sh: строка 4: name: команда не найдена
mv: не удалось выполнить stat для '.zip': Нет такого файла или каталога
finished
[lrabdullina@fedora backup]$ chmod +x backup.sh
[lrabdullina@fedora backup]$ ./backup.sh
```

Figure 5: Написание программы

Выполнение работы

После этого мы делаем наш файл исполняемым при помощи команды `chmod +x backup.sh`. И проверяем работу нашего файла (скриншот 6)



```
Обзор Терминал Ср, 18 мая 12:37 en
lrabduullina@fedora: ~/backup

[lrabduullina@fedora backup]$ chmod +x backup.sh
[lrabduullina@fedora backup]$ ./backup.sh
./backup.sh: строка 3: name: команда не найдена
bzip2: I won't write compressed data to a terminal.
bzip2: For help, type: `bzip2 --help'.
./backup.sh: строка 4: name: команда не найдена
mv: не удалось выполнить stat для '.bz2': Нет такого файла или каталога
finished
[lrabduullina@fedora backup]$ chmod +x backup.sh
[lrabduullina@fedora backup]$ ./backup.sh
./backup.sh: строка 3: name: команда не найдена
bzip2: I won't write compressed data to a terminal.
bzip2: For help, type: `bzip2 --help'.
./backup.sh: строка 4: name: команда не найдена
mv: не удалось выполнить stat для '.bz2': Нет такого файла или каталога
finished
[lrabduullina@fedora backup]$ chmod +x backup.sh
[lrabduullina@fedora backup]$ ./backup.sh
zip warning: missing end signature--probably not a zip file (did you
zip warning: remember to use binary mode when you transferred it?)
zip warning: (if you are trying to read a damaged archive try -F)

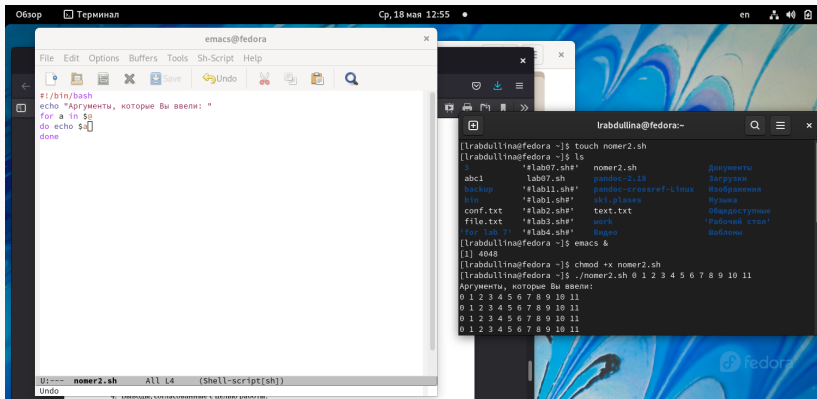
zip error: Zip file structure invalid (backup.sh)
mv: не удалось выполнить stat для 'backup.sh.zip': Нет такого файла или каталога
finished
[lrabduullina@fedora backup]$ chmod +x backup.sh
[lrabduullina@fedora backup]$ ./backup.sh
mv: 'backup.sh.bz2' и '/home/lrabduullina/backup/backup.sh.bz2' - один и тот же файл
finished
[lrabduullina@fedora backup]$ ls
backup.sh- backup.sh.bz2
[lrabduullina@fedora backup]$
```

Figure 6: Делаем файл исполняемым. Проверка работы файла

2. Напишем пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Скрипт может последовательно распечатывать значения всех переданных аргументов.

Выполнение работы

Для начала создадим новый файл `nomer2.sh` с помощью команды `touch`. Вызываем редактор `emacs` в фоновом режиме и пишем нашу программу. После этого делаем файл исполняемым и проверяем его работу. (скриншоты 7,8)



The image consists of two overlapping terminal window screenshots. The background window shows the execution of a script named `nomer2.sh`. The foreground window shows the editing of the same script in the `emacs` editor.

Background Terminal (lrabduellina@fedora):

```
[lrabduellina@fedora ~]$ touch nomer2.sh
[lrabduellina@fedora ~]$ ls
3                'elab07.sh#'    nomer2.sh        Документы
abcl             lab07.sh        pandoc-2.18      Загрузки
backup          'elab11.sh#'   pandoc-crossref-Linux  Изображения
bin             'elab1.sh#'    skt.plases       Музыка
conf.txt        'elab2.sh#'    text.txt         Общедоступные
file.txt        'elab3.sh#'    work             'Рабочий стол'
'for lab 7'     'elab4.sh#'    Bugco            Шаблоны

[lrabduellina@fedora ~]$ emacs &
[1] 4048
[lrabduellina@fedora ~]$ chmod +x nomer2.sh
[lrabduellina@fedora ~]$ ./nomer2.sh 0 1 2 3 4 5 6 7 8 9 10 11
Аргументы, которые Вы ввели:
0 1 2 3 4 5 6 7 8 9 10 11
0 1 2 3 4 5 6 7 8 9 10 11
0 1 2 3 4 5 6 7 8 9 10 11
0 1 2 3 4 5 6 7 8 9 10 11
```

Foreground Terminal (emacs@fedora):

```
#!/bin/bash
echo "Аргументы, которые Вы ввели: "
for a in $@
do echo $a
done
```

Figure 7: Делаем файл исполняемым

Выполнение работы

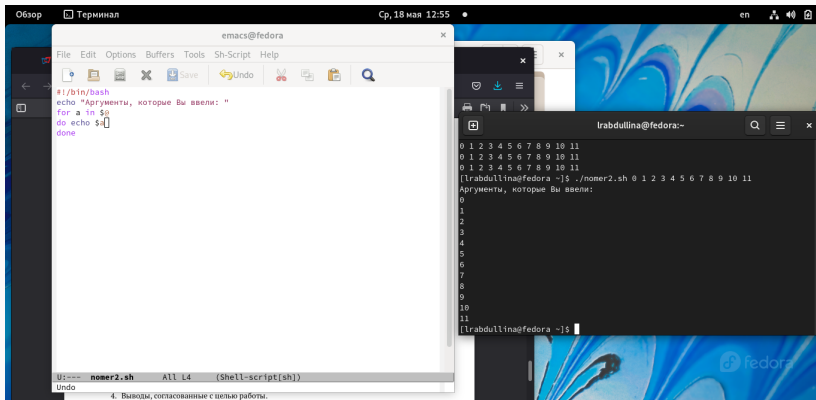


Figure 8: Проверка работы файла

3. Напишем командный файл—аналог команды `ls` (без использования самой этой команды и команды `dir`). Он будет выдавать информацию о нужном каталоге и выводить информацию о возможностях доступа к файлам этого каталога.

Выполнение работы

Для начала создадим новый файл `nomer3.sh` с помощью команды `touch`. Вызываем редактор `emacs` в фоновом режиме и пишем нашу программу. После этого делаем файл исполняемым и проверяем его работу. (скриншоты 9,10,11)

The screenshot displays a Fedora desktop environment with three overlapping windows. The background is a blue and white marbled wallpaper. The top window is a terminal titled 'Терминал' (Terminal) showing the date 'Ср, 18 мая 13:14' and the user 'en'. The middle window is an Emacs editor titled 'emacs@fedora' with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar. It shows the content of a file named 'nomer3.sh' with the following script:

```
#!/bin/bash
path="§1"
for i in $(path)+
do
    echo "$i"
    if test -f $i
    then echo "Файл"
    fi

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

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

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

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

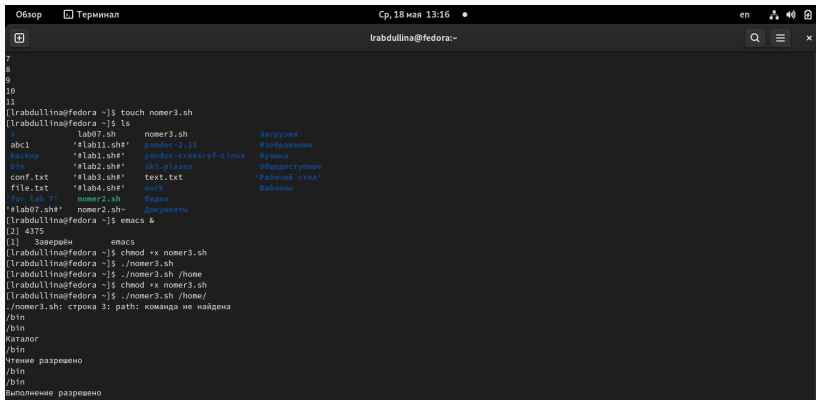
The bottom window is a terminal titled 'lrabduлина@fedora:-' showing the execution of the script:

```
8
9
10
11
[lrabduлина@fedora ~]$ touch nomer3.sh
[lrabduлина@fedora ~]$ ls
3      lab07.sh      nomer3.sh      Загрузки
abc1   '#lab1.sh#'     pandoc-2.18    Изображения
backup '#lab1.sh#'     pandoc-crossref- Linux  Музыка
bin    '#lab2.sh#'     ski.places     Общедоступные
conf.txt '#lab3.sh#'     text.txt       'Рабочий стол'
file.txt '#lab4.sh#'     work          Шаблоны
for lab 7' nomer2.sh     Видео
'#lab07.sh#' nomer2.sh-     Документы
[lrabduлина@fedora ~]$ emacs &
[2] 4375
[1] Завершен emacs
[lrabduлина@fedora ~]$
```

The status bar at the bottom of the Emacs window shows 'U:***- nomer3.sh All L28 (Shell-script[sh] -1)'.

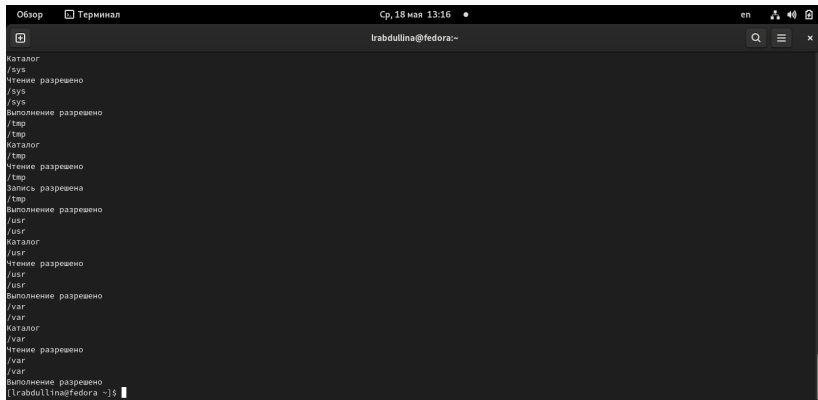
Figure 9: Написание программы

Выполнение работы



```
Обзор Терминал Ср, 18 мая 13:16 en
lrabdullina@fedora:~
7
8
9
10
11
[lrabdullina@fedora ~]$ touch nomer3.sh
[lrabdullina@fedora ~]$ ls
.          lab07.sh      nomer3.sh      Загрузки
abc1       'slab11.sh#'     pandoc-2.18    Изображения
backup     'slab1.sh#'      pandoc-crossref-Linux Музыка
bin        'slab2.sh#'      ski.places     Общедоступные
conf.txt   'slab3.sh#'      text.txt       'Рабочий стол'
file.txt   'slab4.sh#'      work          Шаблоны
'for lab 7' nomer2.sh      Видео
'slab07.sh#' nomer2.sh~     документы
[lrabdullina@fedora ~]$ emacs &
[2] 4375
[1] Завершён      emacs
[lrabdullina@fedora ~]$ chmod +x nomer3.sh
[lrabdullina@fedora ~]$ ./nomer3.sh
[lrabdullina@fedora ~]$ ./nomer3.sh /home
[lrabdullina@fedora ~]$ chmod +x nomer3.sh
[lrabdullina@fedora ~]$ ./nomer3.sh /home/
./nomer3.sh: строка 3: path: команда не найдена
/bin
/bin
Каталог
/bin
Чтение разрешено
/bin
/bin
Выполнение разрешено
```

Figure 10: Делаем файл исполняемым



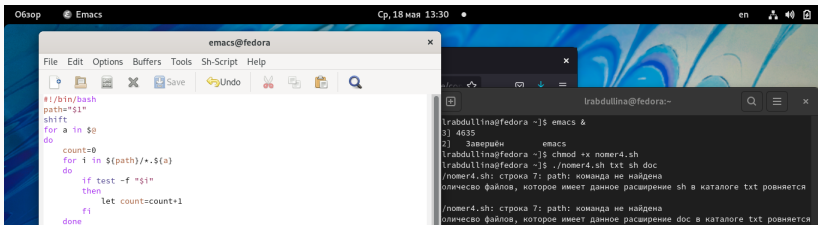
```
Обзор Терминал Ср, 18 мая 13:16 en 100% 100% 100%  
lrabdullina@fedora:~  
Каталог  
/sys  
Чтение разрешено  
/sys  
/sys  
Выполнение разрешено  
/tmp  
/tmp  
Каталог  
/tmp  
Чтение разрешено  
/tmp  
Запись разрешена  
/tmp  
Выполнение разрешено  
/usr  
/usr  
Каталог  
/usr  
Чтение разрешено  
/usr  
/usr  
Выполнение разрешено  
/var  
/var  
Каталог  
/var  
Чтение разрешено  
/var  
/var  
Выполнение разрешено  
lrabdullina@fedora ~]$
```

Figure 11: Проверка работы файла

Выполнение работы

4. Напишем 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.

Для начала создадим новый файл `nomer4.sh` с помощью команды `touch`. Вызываем редактор `emacs` в фоновом режиме и пишем нашу программу. После этого делаем файл исполняемым и проверяем его работу. (скриншоты 12,13)



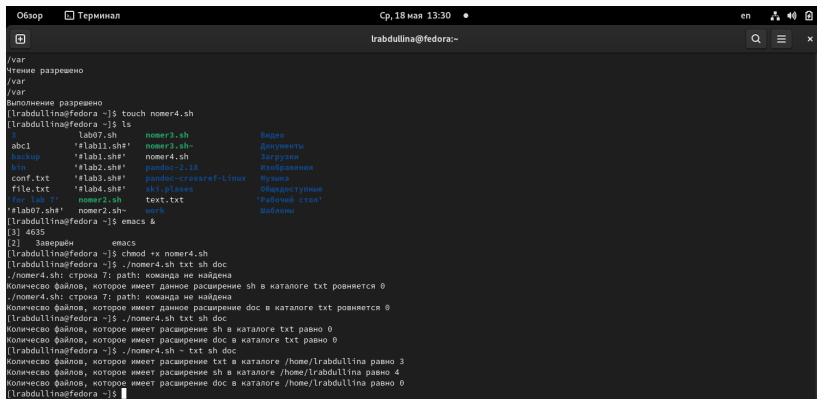
```
emacs@fedora
File Edit Options Buffers Tools Sh-Script Help

#!/bin/bash
path="$1"
shift
for a in $@
do
count=0
for i in $(path)/.*.$(a)
do
if test -f "$i"
then
let count=count+1
fi
done
done
```

```
lrabduлина@fedora:~
lrabduлина@fedora ~]$ emacs &
3) 4635
2) Завершён emacs
lrabduлина@fedora ~]$ chmod +x nomer4.sh
lrabduлина@fedora ~]$ ./nomer4.sh txt sh doc
./nomer4.sh: строка 7: path: команда не найдена
оличество файлов, которое имеет данное расширение sh в каталоге txt равняется

./nomer4.sh: строка 7: path: команда не найдена
оличество файлов, которое имеет данное расширение doc в каталоге txt равняется
```

Выполнение работы



```
Обзор Терминал Cp, 18 мая 13:30 en [Icons]
lrabduullina@fedora:~
/var
Чтение разрешено
/var
/var
Выполнение разрешено
[lrabduullina@fedora ~]$ touch nomer4.sh
[lrabduullina@fedora ~]$ ls
ls lab07.sh nomer3.sh Видео
abc1 'slab11.sh' nomer3.sh- Документы
backup 'slab1.sh' nomer4.sh Загрузки
bin 'slab2.sh' pandoc-2.18 Изображения
conf.txt 'slab3.sh' pandoc-crossref-Linux Музыка
file.txt 'slab4.sh' ski.places Общедоступные
'for lab 7' nomer2.sh text.txt 'Рабочий стол'
'slab07.sh' nomer2.sh- work Шаблоны
[lrabduullina@fedora ~]$ emacs &
[3] 4635
[2] Завершён emacs
[lrabduullina@fedora ~]$ chmod +x nomer4.sh
[lrabduullina@fedora ~]$ ./nomer4.sh txt sh doc
./nomer4.sh: строка 7: path: команда не найдена
Количество файлов, которое имеет данное расширение sh в каталоге txt равняется 0
./nomer4.sh: строка 7: path: команда не найдена
Количество файлов, которое имеет данное расширение doc в каталоге txt равняется 0
[lrabduullina@fedora ~]$ ./nomer4.sh txt sh doc
Количество файлов, которое имеет расширение sh в каталоге txt равно 0
Количество файлов, которое имеет расширение doc в каталоге txt равно 0
[lrabduullina@fedora ~]$ ./nomer4.sh - txt sh doc
Количество файлов, которое имеет расширение txt в каталоге /home/lrabduullina равно 3
Количество файлов, которое имеет расширение sh в каталоге /home/lrabduullina равно 4
Количество файлов, которое имеет расширение doc в каталоге /home/lrabduullina равно 0
[lrabduullina@fedora ~]$
```

Figure 13: Делаем файл исполняемым. Проверка работы файла

В ходе лабораторной работы мы изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать небольшие командные файлы