

# Программирование в командном процессоре ОС UNIX. Командные файлы.

Лабораторная работа №10

---

Нати Ф. Б.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Нати Франсиску Бунда
- студент 1 курса, группа НММбд-02-22
- Российский университет дружбы народов



## Вводная часть

---

- Командный процессор ОС UNIX
- Командные файлы

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

- Ознакомиться с теоретическим материалом.
- Выполнить упражнения.
- Ответить на контрольные вопросы.

## Выполнение лабораторной работы №10

---



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

```
fbnati@fedora:~$ man zip
ZIP(1L)                                ZIP(1L)

NAME
    zip - package and compress (archive) files

SYNOPSIS
    zip [-aABcdDeEffghjklLmsqrRSTuvVwKyzI@] [--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
    (Manual page not available) (page 1, forward h for help or a to exit)
```

```
fbnati@fedora:~$ man bzip2
bzip2(1)                               General Commands Manual    bzip2(1)

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

SYNOPSIS
    bzip2 [-cdffkqtzvvVL123456789] [ filenames ... ]
    bunzip2 [-fkvaVL] [ filenames ... ]
    bzcata [-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.

    (Manual page not available) (page 1, forward h for help or a to exit)
```

# Первая программа

```
[fbnati@fedora ~]$ touch lab10_1.sh  
[fbnati@fedora ~]$
```

Открыть ▾ (+) • lab10\_1.sh  
~/

```
#!/bin/bash  
name="lab10_1.sh"  
mkdir ~/backup/  
bzip2 -k ${name}  
mv ${name}.bz2 ~/backup/  
echo
```

```
[fbnati@fedora ~]$ ls  
lab09.sh  work  документы  Изображения  Общедоступные  Шаблоны  
lab10_1.sh  Видео  Загрузки  Музыка  "Рабочий стол"  
[fbnati@fedora ~]$ chmod +x *.sh  
[fbnati@fedora ~]$ ./lab10_1.sh  
  
[fbnati@fedora ~]$ ls  
backup  work  Загрузки  Общедоступные  
lab09.sh  Видео  Изображения  "Рабочий стол"  
lab10_1.sh  документы  Музыка  Шаблоны  
lab10_1.sh.bz2  
[fbnati@fedora ~]$ ls backup  
lab10_1.sh.bz2  
[fbnati@fedora ~]$
```

## Вторая программа

```
[fbnati@fedora ~]$ touch lab10_2.sh
```

Открыть ▾ 

• lab10\_2.sh

~/


```
#!/bin/bash
```

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

```
[fbnati@fedora ~]$ chmod +x *.sh
[fbnati@fedora ~]$ ls
backup  lab10_2.sh  документы  Музыка  Шаблоны
lab09.sh  work      Загрузки  Общедоступные
lab10_1.sh  видео     Изображения  'Рабочий стол'
[fbnati@fedora ~]$ ./lab10_2.sh 2 2 2 2
2 2 2 2
2 2 2 2
2 2 2 2
2 2 2 2
[fbnati@fedora ~]$ ./lab10_2.sh 5 5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
[fbnati@fedora ~]$
```

# Третья программа

```
[fbnati@fedora ~]$ touch lab10_3.sh
```

Открыть ▾  lab10\_3.sh  
~/

```
#!/bin/bash
a="$1"
for i in ${a}/*
do
echo '$1'
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
```

```
[fbnati@fedora ~]$ chmod +x *.sh
[fbnati@fedora ~]$ ls
backup      lab10_2.sh  Видео      Изображения  'Рабочий стол'
lab09.sh    lab10_3.sh  Документы  Музыка        Яблоны
lab10_1.sh  work       Загрузки   Общедоступные
[fbnati@fedora ~]$ ./lab10_3.sh
$1
каталог
чтение разрешено
выполнение разрешено
$1
каталог
чтение разрешено
выполнение разрешено
$1
каталог
чтение разрешено
выполнение разрешено
$1
каталог
чтение разрешено
выполнение разрешено
$1
```

# Четвёртая программа

```
[fbnati@fedora ~]$ touch lab10_4.sh
[fbnati@fedora ~]$
```

```
Открыть ▾ + lab10_4.sh
~

#!/bin/bash
b="$1"
shift
for j in $(ls)
do
k=0
for i in $(b)/+.$(j)
do
if test -f "$i"
then
let k=k+1
fi
done
echo "$k файлов содержится в директории $b с расширением $j"
done
```

```
[fbnati@fedora ~]$ chmod +x *.sh
[fbnati@fedora ~]$ ls
backlog lab10_2.sh work Загрузки Общедоступные
lab09.sh lab10_3.sh Видео Изображения 'Рабочий стол'
lab10_1.sh lab10_4.sh Документы Музыка Шаблоны
[fbnati@fedora ~]$ ./lab10_4.sh ~ sh
5 файлов содержится в директории /home/fbnati с расширением sh
[fbnati@fedora ~]$ ls
backlog lab10_2.sh work Загрузки Общедоступные
lab09.sh lab10_3.sh Видео Изображения 'Рабочий стол'
lab10_1.sh lab10_4.sh Документы Музыка Шаблоны
```

13. При вызове командного файла на выполнение параметры ему могут быть переданы точно таким же образом, как и выполняемой программе. С точки зрения командного файла эти параметры являются позиционными. Символ \$ является метасимволом командного процессора. Он используется, в частности, для ссылки на параметры, точнее, для получения их значений в командном файле. В командный файл можно передать до девяти параметров. При использовании где-либо в командном файле комбинации символов \$i, где  $0 < i < 10$ , вместо неё будет осуществлена подстановка значения параметра с порядковым номером i, т. е. аргумента командного файла с порядковым номером i. Использование комбинации символов \$0

14. Специальные переменные:

\$\* – отображается вся командная строка или параметры оболочки; \$? – код завершения последней выполненной команды; \$\$ – уникальный идентификатор процесса, в рамках которого выполняется командный процессор; \$! – номер процесса, в рамках которого выполняется последняя вызванная на выполнение в командном режиме команда; \$- – значение флагов командного процессора; \$# – возвращает целое число – количество слов, которые были результатом \$; \${#name}

## Результаты

---

В ходе выполнения лабораторной работы ознакомились с операционной системой Linux. Получили практические навыки работы с редактором Emacs.