

# Программирование в командном процессоре ОС UNIX. Командные файлы.

Лабораторная работа №10

---

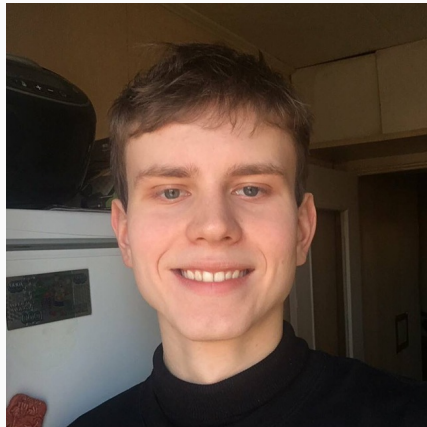
Татур С. А.

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Татур Стефан Андреевич
- студент 1 курса, группа НММбд-03-22
- 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

---



```
~]$ man zip  
~]$ man bzip2  
~]$ man tar
```

```
ZIP(1L)                                ZIP(1L)  
  
NAME  
    zip - package and compress (archive) files  
  
SYNOPSIS  
    zip [-aABcdDeffGghjklLmoqrRSTuvVwXyz!@#] [--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  
    Manual name zip(1) line 1 (press h for help or q to quit)
```

```
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 [-cdfkqstvwVL123456789] [filenames ...]  
    bunzip2 [-fkvsVL] [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 name bzip2(1) line 1 (press h for help or q to quit)
```

# Первая программа

```
~]$ touch lab10_1.sh
```

```
lab10_1.sh
~/
Открыть ▾ +

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

|            |           |             |                |
|------------|-----------|-------------|----------------|
| backup     | work      | Загрузки    | Общедоступные  |
| lab09.sh~  | Видео     | Изображения | 'Рабочий стол' |
| lab10_1.sh | Документы | Музыка      | Шаблоны        |

## Вторая программа

```
~]$ touch lab10_2.sh
```

```
Открыть ▾ [icon] lab10_2.sh
~j

#!/bin/bash

echo
for a in $(ls)
do echo $a
done
```

[illegible]

## Третья программа

```
~]$ touch lab10_3.sh
```

```
Открыть ▾ [icon] lab10_3.sh
#!/bin/bash
a="11"
for i in $(a)
do
    echo "11"

    if test -f $1
    then echo "доступ дано"
    fi

    if test -d $1
    then echo "каталог"
    fi

    if test -r $1
    then echo "чтение разрешено"
    fi

    if test -w $1
    then echo "запись разрешена"
    fi

    if test -x $1
    then echo "выполнение разрешено"
    fi
done
```

```
каталог
чтение разрешено
выполнение разрешено
```

```
каталог
чтение разрешено
выполнение разрешено
```

```
каталог
чтение разрешено
выполнение разрешено
```

```
каталог
чтение разрешено
выполнение разрешено
```

```
каталог
чтение разрешено
выполнение разрешено
```

```
каталог
чтение разрешено
выполнение разрешено
```

## Четвёртая программа

```
~]$ touch lab10_4.sh
```

```
Открыть ▾  lab10_4.sh
#!/bin/bash
b="33"
shift
for j in $(
do
    k=0
    for i in $(N)/$(k(j)
    do
        if test -f "$i"
        then
            let k=k+1
        fi
    done
    echo "$k файлов содержится в директории $b с расширением $j"
done
```

|            |            |           |               |                |
|------------|------------|-----------|---------------|----------------|
| lab09.sh~  | lab10_3.sh | Видео     | Изображения   | 'Рабочий стол' |
| lab10_1.sh | lab10_4.sh | Документы | Музыка        | Шаблоны        |
| lab10_2.sh | work       | Загрузки  | Общедоступные |                |

1. 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 Сподобный 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).
2. 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

12. Команду «set» можно использовать для вывода списка переменных окружения. В системах Ubuntu и Debian команда «set» также выведет список функций командной оболочки после списка переменных командной оболочки. Поэтому для ознакомления со всеми элементами списка переменных окружения при работе с данными системами рекомендуется использовать команду «set | more». Команда «typeset» предназначена для наложения ограничений на переменные. Команду «unset» следует использовать для удаления переменной из окружения командной оболочки.
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 приводит к подстановке вместо неё имени данного командного файла.

## Результаты

---

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