

Лабораторная работа 10 Программирование в командном процессоре ОС UNIX. Командные файлы

Хусаинова Динара Айратовна

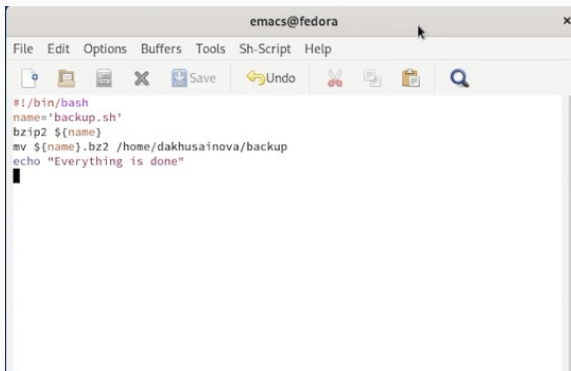
28.04.2022

RUDN

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

Первый командный файл

Это скрипт, который при запуске будет делать резервную копию самого себя в другую директорию backup в домашнем каталоге. При этом файл будет архивироваться архиватором bzip2(рис. 1,2).

A screenshot of an Emacs editor window titled 'emacs@fedora'. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with icons for opening files, saving, undo, redo, and search. The main text area contains a shell script with the following content:

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

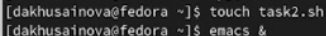
A cursor is visible at the end of the last line of the script.

Figure 1: Командный код

```
[dakhusainova@fedora ~]$ cd backup  
[dakhusainova@fedora backup]$ ls  
backup.sh.bz2  
[dakhusainova@fedora backup]$
```

Figure 2: Проверяем работу

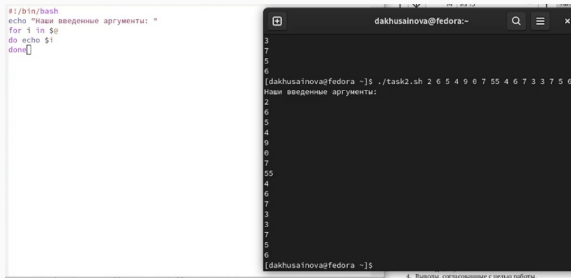
Файл, обрабатывающий любое число аргументов командной строки.
Выведем последовательно введенные числа(рис. 3, 4).



```
[dakhusainova@fedora ~]$ touch task2.sh  
[dakhusainova@fedora ~]$ emacs &
```

Figure 3: Создание файла для второго скрипта

Второй командный файл Проверка



The image displays two terminal windows side-by-side. The left window shows a shell script for processing command-line arguments. The right window shows the execution of a script named `task2.sh` with a list of numbers as arguments, resulting in a list of names.

```
#!/bin/bash
echo "Наши введенные аргументы: "
for i in $@
do echo $i
done
```

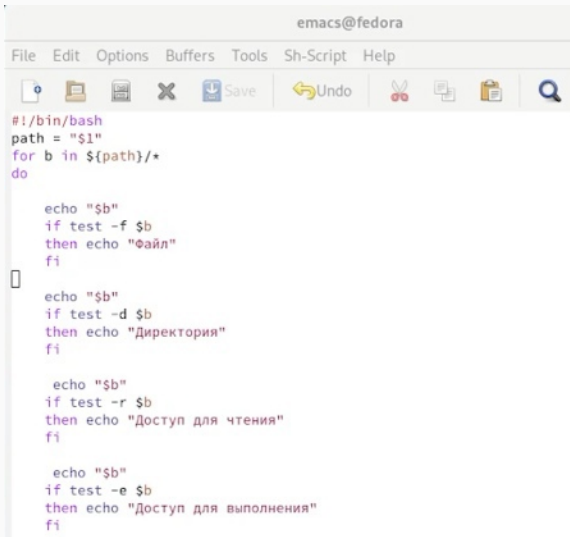
```
dakhusainova@fedora:~$ ./task2.sh 2 6 5 4 9 0 7 55 4 6 7 3 3 7 5 6
Наши введенные аргументы:
2
6
5
4
9
0
7
55
4
6
7
3
3
7
5
6
dakhusainova@fedora:~$
```

4. Результаты, полученные с помощью `task2.sh`.

Figure 4: Проверка работы файла

Третий командный файл

Напишем аналог команды `ls` без использования самой этой команды и команды `dir` (рис. 5,6).

The image shows a screenshot of the Emacs text editor running on a Fedora system. The title bar at the top reads 'emacs@fedora'. Below it is a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. A toolbar contains icons for opening a file, saving, undo, redo, and search. The main text area contains a shell script written in a syntax-highlighted language. The script is a function that takes a path as an argument and lists its contents, including file types and permissions. The script is as follows:

```
#!/bin/bash
path = "$1"
for b in ${path}/*
do

    echo "$b"
    if test -f $b
    then echo "Файл"
    fi

    echo "$b"
    if test -d $b
    then echo "Директория"
    fi

    echo "$b"
    if test -r $b
    then echo "Доступ для чтения"
    fi

    echo "$b"
    if test -e $b
    then echo "Доступ для выполнения"
    fi
```

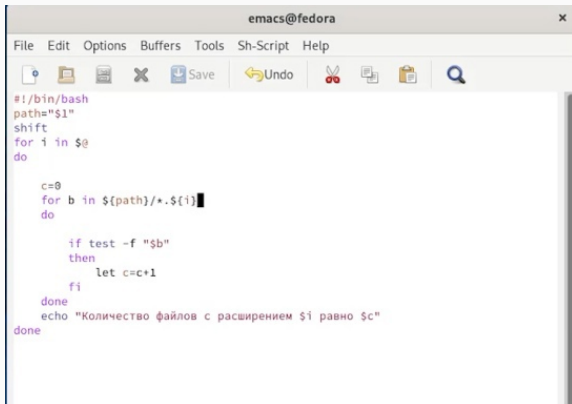
Третий командный файл Проверка

```
/tmp
Доступ для чтения
/tmp
Доступ для выполнения
/tmp
Доступ для записи
/usr
/usr
Директория
/usr
Доступ для чтения
/usr
Доступ для выполнения
/usr
/var
/var
Директория
/var
Доступ для чтения
/var
Доступ для выполнения
/var
```

Figure 6: Наблюдаем вывод информации о файлах и каталогах

Пишем 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. Создаем файл, пишем код и запускаем, видим, сколько файлов заданного формата есть в проверяемом каталоге(рис. 7,8).

Четвертый командный файл Заполнение файла

A screenshot of the Emacs editor window titled 'emacs@fedora'. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with icons for file operations like 'Save', 'Undo', 'Cut', 'Copy', 'Paste', and 'Find'. The main text area contains a shell script in Bash. The script starts with a shebang line, sets a path, shifts arguments, and then iterates over each argument to count the number of files in that directory with a specific extension. The script ends with an echo statement and a 'done' keyword.

```
#!/bin/bash
path="$1"
shift
for i in $@
do
    c=0
    for b in ${path}/*.${i}
    do
        if test -f "$b"
        then
            let c=c+1
        fi
    done
    echo "Количество файлов с расширением $i равно $c"
done
```

Figure 7: Вводим команды в файл

```
[dakhusainova@fedora ~]$ ./task4.sh /home/dakhusainova txt md sh  
Количество файлов с расширением txt равно 8  
Количество файлов с расширением md равно 0  
Количество файлов с расширением sh равно 7
```

Figure 8: Запускаем

Изучили основы программирования в оболочке ОС UNIX/Linux и научились писать небольшие командные файлы.