

# Отчёт по лабораторной работе №8

## Программирование в командном процессоре ОС UNIX.

### Командные файлы

Архипов Юрий Денисович

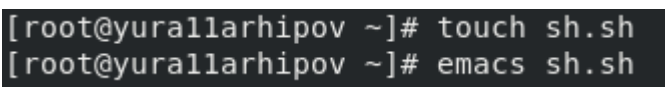
## Цель работы

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

## Ход работы

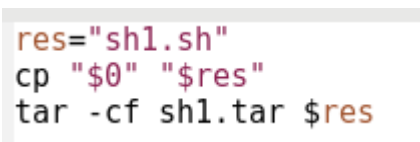
1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию backup в моём домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор zip, bzip2 или tar.

а) Создадим новый файл, запустим его в редакторе emacs.

Рис.1 

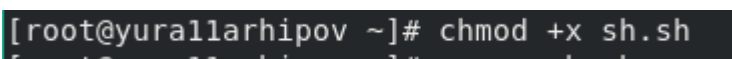
```
[root@yurallarhipov ~]# touch sh.sh
[root@yurallarhipov ~]# emacs sh.sh
```

б) Напишем скрипт.

Рис.2 

```
res="sh1.sh"
cp "$0" "$res"
tar -cf sh1.tar $res
```

в) Изменим права доступа.

Рис.3 

```
[root@yurallarhipov ~]# chmod +x sh.sh
```

г) Запускаем скрипт и проверяем работу программы.

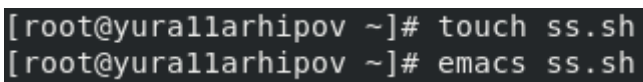
Рис.4 

```
[root@yurallarhipov ~]# ls
anaconda-ks.cfg  feathers  lab07.sh  newfile.txt  sh.sh  ski.plases
australia        initial-setup-ks.cfg  my_os     play         sh.sh~
[root@yurallarhipov ~]# ./sh.sh
[root@yurallarhipov ~]# ls
anaconda-ks.cfg  initial-setup-ks.cfg  newfile.txt  sh1.tar  ski.plases
australia        lab07.sh              play         sh.sh
feathers          my_os                 sh1.sh      sh.sh~
```

2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе

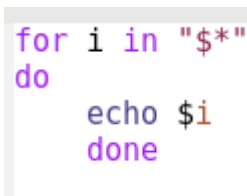
превышающие десять. Например, скрипт может последовательно распечатать значения всех переданных аргументов.

а) Создадим новый файл, запустим его в редакторе emacs.

Рис.5 

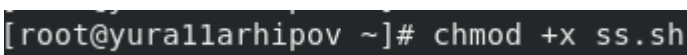
```
[root@yurallarhipov ~]# touch ss.sh
[root@yurallarhipov ~]# emacs ss.sh
```

б) Напишем скрипт.

Рис.6 

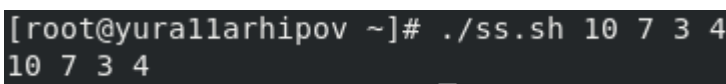
```
for i in "$*"
do
    echo $i
done
```

с) Изменим права доступа.

Рис.7 

```
[root@yurallarhipov ~]# chmod +x ss.sh
```

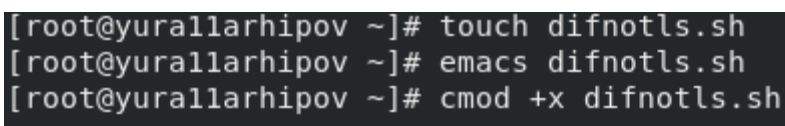
д) Запускаем скрипт и проверяем работу программы.

Рис.8 

```
[root@yurallarhipov ~]# ./ss.sh 10 7 3 4
10 7 3 4
```

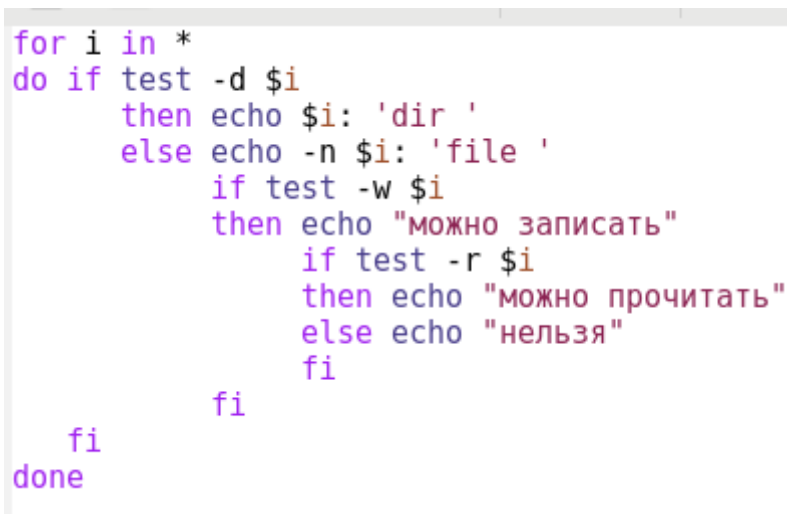
3. Написать командный файл – аналог команды ls (без использования самой этой команды и команды dir). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.

а) Создадим новый файл, запустим его в редакторе emacs и меняем права доступа.

Рис.9 

```
[root@yurallarhipov ~]# touch difnotls.sh
[root@yurallarhipov ~]# emacs difnotls.sh
[root@yurallarhipov ~]# chmod +x difnotls.sh
```

б) Напишем скрипт.

Рис.10 

```
for i in *
do if test -d $i
    then echo $i: 'dir '
    else echo -n $i: 'file '
        if test -w $i
        then echo "можно записать"
            if test -r $i
            then echo "можно прочитать"
            else echo "нельзя"
            fi
        fi
    fi
done
```

с) Запускаем скрипт и проверяем работу.

```
[root@yurallarhipov ~]# ./difnotls.sh
anaconda-ks.cfg: file можно записать
можно прочитать
australia: dir
difnotls.sh: file можно записать
можно прочитать
difnotls.sh~: file можно записать
можно прочитать
feathers: file можно записать
можно прочитать
initial-setup-ks.cfg: file можно записать
можно прочитать
lab07.sh: file можно записать
можно прочитать
my_os: file можно записать
можно прочитать
newfile.txt: file можно записать
можно прочитать
play: dir
sh1.sh: file можно записать
можно прочитать
sh1.tar: file можно записать
можно прочитать
sh.sh: file можно записать
можно прочитать
sh.sh~: file можно записать
можно прочитать
ski.places: dir
ss.sh: file можно записать
можно прочитать
ss.sh~: file можно записать
можно прочитать
```

Рис.11

4. Написать командный файл, который получает в качестве аргумента командной строки формат файла и выдаёт количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.

а) Создадим новый файл, запустим его в редакторе emacs и меняем права доступа.

```
[root@yurallarhipov ~]# touch file.sh
[root@yurallarhipov ~]# emacs file.sh
[root@yurallarhipov ~]# chmod +x file.sh
```

Рис.12

б) Напишем скрипт.

```
find $1 -name "*. $2" -type f | wc -l
```

Рис.13

с) Запускаем скрипт и проверяем работу.

Рис.14

```
[root@yurallarhipov ~]# ./file.sh /root sh
6
[root@yurallarhipov ~]# ls
anaconda-ks.cfg  feathers          lab07.sh          sh1.sh           ski.plases
australia        file.sh          my_os            sh1.tar          ss.sh
difnotls.sh      file.sh~         newfile.txt      sh.sh            ss.sh~
difnotls.sh~     initial-setup-ks.cfg play              sh.sh~
```

## Вывод

Я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать небольшие командные файлы.