

# Лабораторная работа 11

Программирование в командном процессоре ОС UNIX. Ветвления и  
циклы

Гарут Александр Игоревич

# Содержание

Цель работы	3
Ход работы	4
Вывод	8
Контрольные вопросы	9

## Цель работы

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

# Ход работы

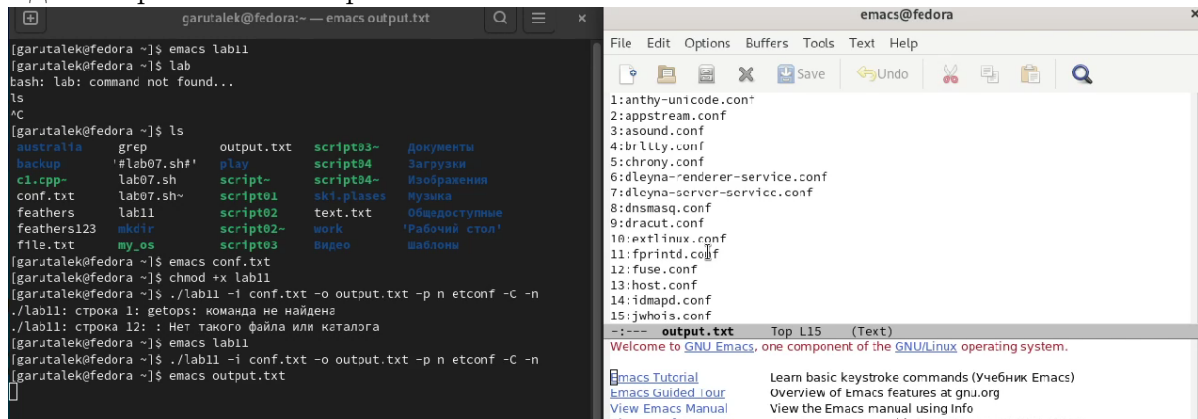
1. Выполним пункт 1 лабораторной работы

```
while getopts "i:o:p:C:n" opt
do
case $opt in
i) inputfile="$OPTARG";;
o) outputfile="$OPTARG";;
p) shablon="$OPTARG";;
C) registr="";;
n) number="";;
esac
done

grep -n "$shablon" "$inputfile" > "$outputfile"
```

Изображение1.1: Код скрипта

Удостоверимся что всё работает



Изображение1.2: Запуск скрипта

2. Напишем скрипт который указывает больше нуля, равно нулю или меньше нуля наше число

```

echo "Insert num"
read n
if [ $n -gt 0 ]
then echo ">0"
elif [ $n -eq 0 ]
then echo "=0"
else echo "<0"
fi

```

Изображение2.1: Код скрипта

Удостовермся что всё работает

```

[garutalek@fedora ~]$ chmod +x lab11_2
[garutalek@fedora ~]$ ./lab11_2
Insert num
2
>0
[garutalek@fedora ~]$ ./lab11_2
Insert num
0
=0
[garutalek@fedora ~]$ ./lab11_2
Insert num
-3
<0

```

Изображение2.2: Запуск скрипта

3. Напишем скрипт, создающий и удаляющий tmp файлы в указанном нами количестве

```

while getopts "c:r" opt
do
case $opt in
c)n="$OPTARG"; for i in $(seq 1 $n); do touch "$i.tmp"; done;;
r)for i in $(find -name "*.tmp"); do rm $i; done;;
esac
done

```

Изображение3.1: Код скрипта

Удостовермся что всё работает

```
[garutalek@fedora ~]$ ./lab11 3 -c 2
[garutalek@fedora ~]$ ls
1.tmp      feathers123  lab11~      play        script04     Загрузки
2.tmp      file.txt    lab11_2     script~     script04~    Изображения
australia  grep        lab11_3     script01    ski.plases   Музыка
backup     '#lab07.sh#' lab11_3~    script02    text.txt     Общедоступные
c1.cpp~    lab07.sh    mkdir       script02~   work         'Рабочий стол'
conf.txt   lab07.sh~   my_os       script03    Видео        Шаблоны
feathers    lab11       output.txt  script03~   Документы
```

```
[garutalek@fedora ~]$ ./lab11 3 -r
[garutalek@fedora ~]$ ls
australia  '#lab07.sh#'  mkdir       script03     Документы
backup     lab07.sh       my_os       script03~    Загрузки
c1.cpp~    lab07.sh~     output.txt  script04     Изображения
conf.txt   lab11         play        script04~    Музыка
feathers    lab11~        script~     ski.plases   Общедоступные
feathers123 lab11_2       script01    text.txt     'Рабочий стол'
file.txt   lab11_3       script02    work         Шаблоны
grep       lab11_3~      script02~   Видео
```

Изображение3.2: Запуск скрипта

4. Напишем скрипт, архивирующий недавно изменённые файлы

```
while getopts ":p:" opt; do
case $opt in
p)dir="$OPTARG";;
esac
done

find $dir -mtime +0 -mtime -7 -print0 | xargs -0 tar -cf archive.tar
```

Изображение4.1: Код скрипта

Удостовермся что всё работает

```

[garutalek@fedora ~]$ ls
archive.tar  '#lab07.sh#'  lab11_4~  script03  Документы
australia    lab07.sh      mkdir      script03~  Загрузки
backup       lab07.sh~     my_os      script04    Изображения
c1.cpp~      lab11         output.txt  script04~  Музыка
conf.txt     lab11~        play       ski.plases  Общедоступные
feathers     lab11_2       script~    tar         'Рабочий стол'
feathers123  lab11_3       script01   text.txt    Шаблоны
file.txt     lab11_3~     script02   work
grep        lab11_4       script02~  Видео

[garutalek@fedora ~]$ cd archive.tar
bash: cd: archive.tar: Это не каталог
[garutalek@fedora ~]$ tar -xf archive.tar -C /home/garutalek/tmp
tar: /home/garutalek/tmp: Функция open завершилась с ошибкой: Нет такого файла и
ли каталога
tar: Error is not recoverable: exiting now
[garutalek@fedora ~]$ mkdir tmp
[garutalek@fedora ~]$ tar -xf archive.tar -C /home/garutalek/tmp
[garutalek@fedora ~]$ cd tmp
[garutalek@fedora tmp]$ ls
backup      lab07.sh      script01     script03     script04~
c1.cpp~     lab07.sh~     script02     script03~    text.txt
'#lab07.sh#' script~       script02~    script04     work

```

Изображение4.2: Запуск скрипта

## Вывод

Изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.



# Контрольные вопросы

1. Команда `getopts` возвращает нулевой код завершения, если опция найдена. Если в командной строке больше не остается аргументов или же текущий аргумент не начинается с дефиса, `getopts` возвращает ненулевой код завершения.
2. При генерации имен используют метасимволы: `*`, произвольная (возможно пустая) последовательность символов