

# **"Отчёт по лабораторной работе №11"**

**"Дисциплина: Операционные системы"**

Горпинич Елена Михайловна

## Цель работы

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

# Выполнение лабораторной работы

1. Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами:
  1. `-iinputfile`—прочитать данные из указанного файла;
  2. `-ooutputfile`—вывести данные в указанный файл;
  3. `-p шаблон` —указать шаблон для поиска;
  4. `-C`—различать большие и малые буквы;
  5. `-n`—выдавать номера строк, а затем ищет в указанном файле нужные строки, определяемые ключом `-p`. Для данной задачи я создала файл `prog1.sh` и написала соответствующие скрипты

```
emgorpinich@emgorpinich:~$ touch pr1.sh
emgorpinich@emgorpinich:~$ emacs &
```

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:op:Cn optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) pflag=1; pval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo illegal option $optletter
    esac
done
if (($pflag==0))
then echo "Шаблон не найден"
else
    if (($iflag==0))
    then echo "Файл не найден"
    else
        if (($oflag==0))
        then if (($Cflag==0))
            then if ((nflag==0))
                then grep $pval $ival
                else grep -n $pval $ival
                fi
            else if (($nflag==0))
                then grep -l $pval $ival
                else grep -i -n $pval $ival
                fi
            fi
        fi
    fi
fi
```

```
else if (($Cflag==0))
  then if (($nflag==0))
    then grep $pval $ival > $oval
    else grep -n $pval $ival > $oval
    fi
  else if (($nflag==0))
    then grep -i $pval $ival > $oval
    else grep -i -n $pval $ival > $oval
    fi
  fi
fi
fi
fi
```

Проверила работу написанного скрипта, используя различные опции, предварительно добавив право на исполнение файла и создав два файла, которые необходимы для выполнения программы: lab11\_1.txt и lab11\_2.txt. Скрипт работает исправно.

```
emgorpinich@emgorpinich:~$ chmod +x pr1.sh  
emgorpinich@emgorpinich:~$ touch lab11_1.txt lab11_2.txt
```

```
emgorpinich@emgorpinich:~$ cat lab11_1.txt
lab11
stage1
abc abc
ok
1
emgorpinich@emgorpinich:~$ ./pr1.sh -i lab11_1.txt -o lab11_2.txt -p abc -n
emgorpinich@emgorpinich:~$ cat lab11_2.txt
3:abc abc
emgorpinich@emgorpinich:~$ ./pr1.sh -i lab11_1.txt -o lab11_2.txt -p abc -C -n
emgorpinich@emgorpinich:~$ cat lab11_2.txt
3:abc abc
emgorpinich@emgorpinich:~$ ./pr1.sh -i lab11_1.txt -C -n
Шаблон не найден
emgorpinich@emgorpinich:~$ ./pr1.sh -i lab11_2.txt -p abc -C -n
1:3:abc abc
emgorpinich@emgorpinich:~$ ./pr1.sh -o lab11_2.txt -p abc -C -n
Файл не найден
```

2. Написала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. Для данной задачи я создала 2 файла: `ch.c` и `ch.sh` и написала соответствующие скрипты.

```
emgorpinich@emgorpinich:~$ touch ch.c ch.sh
emgorpinich@emgorpinich:~$ emacs &
```



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     printf("Введите число\n");
6     int a;
7     scanf ("%d",&a);
8     if (a<0) exit(0);
9     if (a>0) exit(1);
10    if (a==0) exit(2);
11    return 0;
12 }
```

```
#!/bin/bash
gcc chslo.c -o chslo
./chslo
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число больше 0";;
    2) echo "Число равно 0"
esac
```

Проверила работу написанных скриптов , предварительно добавив право на исполнение файла. Скрипты работают верно.

```
emgorpinich@emgorpinich:~$ chmod +x ch.sh
[1]+  Done                  emacs
emgorpinich@emgorpinich:~$ ./ch.sh
Введите число
8
Число > 0
emgorpinich@emgorpinich:~$ ./ch.sh
Введите число
0
Число 0
emgorpinich@emgorpinich:~$ ./ch.sh
Введите число
-9
Число < 0
```

3. Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы. Для данной задачи я создала файл: [file.sh](#) и написала соответствующий скрипт.

```
emgorpinich@emgorpinich:~$ touch file.sh  
emgorpinich@emgorpinich:~$ emacs &
```

```
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function File()
{
    for (( i=1; i<=$number; i++ )) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
File
```

Далее я проверила работу написанного скрипта, предварительно добавив право на исполнение файла. Сначала я создала несколько файлов, а потом удалила их

```
emgorpinich@emgorpinich:~$ chmod +x file.sh
[1]+  Done                  emacs
emgorpinich@emgorpinich:~$ ls
abc          ch.sh~      example2.txt  '#lab07.sh#'  pr4.sh~
abc1         '#chslo.sh#' '#example3.txt#' lab07.sh      prog1.sh~
backup       chslo.sh~   example3.txt  Lab11         prog2.sh
backup.sh    Desktop    '#example4.txt#' lab11_1.txt   prog2.sh~
backup.sh~   Documents  example4.txt  lab11_2.txt   progl.s.sh
ch           Downloads  file.sh       Music         progl.s.sh~
ch.c         '#example1.txt#' file.sh~      Pictures      Public
ch.c~        example1.txt format.sh     pr1.sh        Templates
ch.sh        '#example2.txt#' format.sh~    pr4.sh        Videos
emgorpinich@emgorpinich:~$ ./file.sh -c l11_#.txt 5
emgorpinich@emgorpinich:~$ ls
abc          '#chslo.sh#' example3.txt  l11_4.txt     pr4.sh
abc1         chslo.sh~   '#example4.txt#' l11_5.txt     pr4.sh~
backup       Desktop     example4.txt  '#lab07.sh#' prog1.sh~
backup.sh    Documents  file.sh       lab07.sh      prog2.sh
backup.sh~   Downloads  file.sh~      Lab11         prog2.sh~
ch           '#example1.txt#' format.sh     lab11_1.txt   progl.s.sh
ch.c         example1.txt format.sh~    lab11_2.txt   progl.s.sh~
ch.c~        '#example2.txt#' l11_1.txt     Music         Public
ch.sh        example2.txt l11_2.txt     Pictures      Templates
ch.sh~       '#example3.txt#' l11_3.txt     pr1.sh        Videos
emgorpinich@emgorpinich:~$ ./file.sh -r l11_#.txt 5
emgorpinich@emgorpinich:~$ ls
abc          ch.sh~      example2.txt  '#lab07.sh#'  pr4.sh~
abc1         '#chslo.sh#' '#example3.txt#' lab07.sh      prog1.sh~
backup       chslo.sh~   example3.txt  Lab11         prog2.sh
backup.sh    Desktop    '#example4.txt#' lab11_1.txt   prog2.sh~
backup.sh~   Documents  example4.txt  lab11_2.txt   progl.s.sh
ch           Downloads  file.sh       Music         progl.s.sh~
ch.c         '#example1.txt#' file.sh~      Pictures      Public
ch.c~        example1.txt format.sh     pr1.sh        Templates
ch.sh        '#example2.txt#' format.sh~    pr4.sh        Videos
```

4. Написала командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад. Для данной задачи я создала файл: [pr4.sh](#) и написала соответствующий скрипт

```
emgorpinich@emgorpinich:~$ touch pr4.sh  
emgorpinich@emgorpinich:~$ emacs &
```

```
#!/bin/bash
files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```



Далее я проверила работу написанного скрипта, предварительно добавив право на исполнение файла и создав отдельный Catalog1 с несколькими файлами. Видно что файлы, измененные более недели назад, заархивированы не были. Скрипт работает корректно.

```
emgorpinich@emgorpinich:~/Catalog1$ ~/pr4.sh
a1.txt
a2.txt
chslo
chslo.c
chslo.sh
tar: Catalog1.tar: файл является архивом; не сброшен
emgorpinich@emgorpinich:~/Catalog1$
emgorpinich@emgorpinich:~/Catalog1$ ./pr4.sh
a1.txt
a2.txt
chslo
chslo.c
chslo.sh
tar: Catalog1.tar: файл является архивом; не сброшен
prog4.sh
emgorpinich@emgorpinich:~/Catalog1$ tar -tf Catalog1.tar
a1.txt
a2.txt
chslo
chslo.c
chslo.sh
prog4.sh
```

# Вывод

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX/Linux и научилась писать небольшие командные файлы.