

Отчёт по лабораторной работе №12

Серегин Денис Алексеевич

Содержание

1	Цель работы	5
2	Задания	6
3	Выполнение лабораторной работы	8
4	Выводы	15
5	Использованные ресурсы	16

List of Tables

List of Figures

3.1	Код файл	9
3.2	Текст файла ввода	10
3.3	Проверка работы файла	10
3.4	Код файла на языке C++	11
3.5	Код скрипта	12
3.6	Результат работы исполняемого файла	12
3.7	Код файла	13
3.8	Результат работы исполняемого файла	13
3.9	Код файла	14
3.10	Код файла	14
3.11	Полученный архив	14
3.12	Код файла	14

1 Цель работы

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

2 Задания

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-r`шаблон — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк.

А затем ищет в указанном файле нужные строки, определяемые ключом `-r`.

2. . Написать 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 в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например `1.tmp`, `2.tmp`, `3.tmp`, `4.tmp` и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

3 Выполнение лабораторной работы

1. Используя команды `getopts` `grep`, написал командный файл, который анализирует командную строку с ключами:

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — вывести данные в указанный файл;
- `-р`шаблон — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк.

А затем ищет в указанном файле нужные строки, определяемые ключом `-р`.

(рис 1. 3.1)


```
#!/bin/bash
read=*;find=*;
oflag=0;nflag=0;Cflag=0;iflag=0;pflag=0;
while getopts i:op:Cn optletter
do case $optletter in
    i)iflag=1;
        read=$OPTARG;;
    o)oflag=1;
        write=$OPTARG;;
    p)pflag=1;
        find=$OPTARG;;
    C)Cflag=1;;
    n)nflag=1;;
esac
done
if (let $iflag==1)&&(let $pflag==1)
then if (let $oflag==1)
    then if (let $Cflag==0)
        then if (let $nflag==0)
            then grep -i $find $read >$write
            else grep -i -n $find $read >$write
            fi
        else if (let $nflag==0)
            then grep $find $read >$write
            else grep -n $find $read >$write
            fi
        fi
    else if (let $Cflag==0)
        then if (let $nflag==0)
            then grep -i $find $read
            else grep -i -n $find $read
            fi
        else if (let $nflag==0)
            then grep $find $read
            else grep -n $find $read
            fi
        fi
    fi
else echo "Не было найдено"
fi
```

Figure 3.1: Код файл

(рис 2. 3.2)

```

Hello world, it`s me Mario!
Hello everyone
Hello buddy
WASD
WASD
Hello humans|

```

Figure 3.2: Текст файла ввода

(рис 3. 3.3)

```

Hello everyone
Hello buddy
Hello humans
[daseregin@localhost ~]$ ./lab12.sh -i l12input.txt -p "Hello" -o output.txt
[daseregin@localhost ~]$ cat output.txt
Hello world, it`s me Mario!
Hello everyone
Hello buddy
Hello humans
[daseregin@localhost ~]$ ./lab12.sh -i l12input.txt -p world C
Hello world, it`s me Mario!
[daseregin@localhost ~]$ ./lab12.sh -i l12input.txt -p Hello -n
1:Hello world, it`s me Mario!
2:Hello everyone
3:Hello buddy
6:Hello humans
[daseregin@localhost ~]$ ./lab12.sh -i l12input.txt -p Hello -n -o out2.txt
[daseregin@localhost ~]$ cat out2.txt
1:Hello world, it`s me Mario!
2:Hello everyone
3:Hello buddy
6:Hello humans

```

Figure 3.3: Проверка работы файла

2. Написал на языке C 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 вызывает эту программу и, проанализировав с помощью команды `$?`, выдаёт сообщение о том, какое число было введено.

(рис 4. 3.4)

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main(){
int pr;
cout<<"Vvedite chislo \n";
cin>>pr;
if (pr !=0) {
    if (pr > 0)
        exit(1);
    else
        exit(2);
}
return 0;
}
```

Figure 3.4: Код файла на языке C++

(рис 5. 3.5)

```
#!/bin/bash
./lab122
case $? in
0)echo "число равно 0" ;;
1)echo "больше нуля" ;;
2)echo "меньше нуля" ;;
*)echo "ошибка" ;;
esac
```

Figure 3.5: Код скрипта

(рис 6. 3.6)

```
[daseregin@localhost ~]$ g++ main.cpp -o main
[daseregin@localhost ~]$ ./lab12.1.sh
bash: ./lab12.1.sh: Отказано в доступе
[daseregin@localhost ~]$ chmod +x lab12.1.sh
[daseregin@localhost ~]$ ./lab12.1.sh
Vvedite chislo
0
число равно 0
[daseregin@localhost ~]$ ./lab12.1.sh
Vvedite chislo
5
больше нуля
[daseregin@localhost ~]$ ./lab12.1.sh
Vvedite chislo
-1
меньше нуля
```

Figure 3.6: Результат работы исполняемого файла

3. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например, 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передается в аргументы командной строки. Этот же командный файл удаляет все созданные им файлы

(если они существуют).

(рис 7. 3.7)

```
#!/bin/bash
typeset -i count=0;
for pr in *.tmp
do rm $pr
done
for ((count=1;count<$1+1;count++))
do
    touch $count.tmp
done
```

Figure 3.7: Код файла

(рис 8. 3.8)

```
[daseregin@localhost ~]$ ls|grep .tmp
1.tmp
2.tmp
3.tmp
4.tmp
5.tmp
6.tmp
[daseregin@localhost ~]$ ./lab12.2.sh 5
[daseregin@localhost ~]$ ls|grep .tmp
1.tmp
2.tmp
3.tmp
4.tmp
5.tmp
```

Figure 3.8: Результат работы исполняемого файла

4. Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировал его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовав команду find).

(рис 9. 3.9)

```
#!/bin/bash
cd $1
tar -cf archiv.tar *
```

Figure 3.9: Код файла

(рис 10. 3.10)

```
[daseregin@localhost ~]$ chmod +x lab12.3.sh
[daseregin@localhost ~]$ ./lab12.3.sh
[daseregin@localhost ~]$
```

Figure 3.10: Код файла

(рис 11. 3.11)

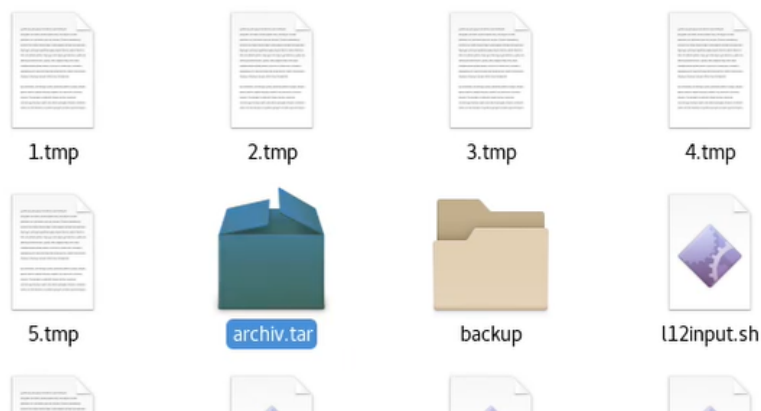


Figure 3.11: Полученный архив

(рис 12. 3.12)

```
#!/bin/bash
cd $1
tar -cf archiv.tar $(find -type f -mtime -7)
```

Figure 3.12: Код файла

4 Выводы

В ходе лабораторной работы мне удалось изучить основы программирования в оболочке ОС UNIX, а также научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

5 Используемые ресурсы

<http://cs.mipt.ru/cpp/labs/lab1.html>

<https://losst.ru/komanda-tar-v-linux>