

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`, написал командный файл, который анализирует командную строку с ключами:

- `-i inputfile` — прочитать данные из указанного файла;
- `-o outputfile` — 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`.

```

1 #!/bin/bash
2 read=\*;find=\*;oflag=0;nflag=0;Cflag=0;iflag=0;pflag=0;
3 while getopts i:o:p:Cn: optletter
4 do case $optletter in
5     i)iflag=1;read=$OPTARG;;
6     o)oflag=1;write=$OPTARG;;
7     p)pflag=1;find=$OPTARG;;
8     C)Cflag=1;;
9     n)nflag=1;;
10 esac
11 done
12 if (let $iflag==1)&&(let $pflag==1)
13 then if (let $oflag==1)
14     then if (let $Cflag==0)
15         then if (let $nflag==0)
16             then grep -i $find $read >$write
17             else grep -i -n $find $read >$write
18             fi
19         else if (let $nflag==0)
20             then grep $find $read >$write
21             else grep -n $find $read >$write
22             fi
23         fi
24     else if (let $Cflag==0)
25         then if (let $nflag==0)
26             then grep -i $find $read
27             else grep -i -n $find $read
28             fi
29         else if (let $nflag==0)
30             then grep $find $read
31             else grep -n $find $read
32             fi
33         fi
34 fi
35 else echo "no necessary options"
36 fi

```

(Рис. 1.1)

Его работа:

```

parallels@nsfedorov:~$ chmod +x lab12_1.sh
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt one
./lab12_1.sh: line 6: syntax error near unexpected token `)'
./lab12_1.sh: line 6: `o)oflag=1;write=$OPTARG;'
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt one
no necessary options
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p one
I was run over by the truth one day.
Made a marble phone book and I carved out all the names
parallels@nsfedorov:~$

```

(Рис. 1.2)

```
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p "one" -o output.txt
parallels@nsfedorov:~$ cat output.txt
I was run over by the truth one day.
Made a marble phone book and I carved out all the names
parallels@nsfedorov:~$
```

(Рис. 1.3)

```
I was run over by the truth one day.
Made a marble phone book and I carved out all the names
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p coat -C
So coat my eyes with butter
parallels@nsfedorov:~$
```

(Рис. 1.4)

```
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p Coat -C
Coat my eyes with butter
Coat my eyes with butter
Coat my eyes with butter
parallels@nsfedorov:~$
```

(Рис. 1.5)

```
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p Coat -n
14:So coat my eyes with butter
22:Coat my eyes with butter
31:Coat my eyes with butter
40:Coat my eyes with butter
```

(Рис. 1.6)

```
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p Coat -n -o output02.txt
parallels@nsfedorov:~$ cat output02.txt
14:So coat my eyes with butter
22:Coat my eyes with butter
31:Coat my eyes with butter
40:Coat my eyes with butter
```

(Рис. 1.7)

```
parallels@nsfedorov:~$ ./lab12_1.sh -i input_12lab.txt -p Coat -C -n
22:Coat my eyes with butter
31:Coat my eyes with butter
40:Coat my eyes with butter
parallels@nsfedorov:~$
```

(Рис. 1.8)

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

Код на C++:

```

1 #include <iostream>
2 #include <stdlib.h>
3 using namespace std;
4 int main() {
5     int op;
6     cout << "input number \n";
7     cin >> op;
8     if (op != 0) {
9         if (op > 0)
10             exit(1);
11         else
12             exit(2);
13     }
14     return 0;
15 }

```

(Рис. 2.1)

Код скрипта:

```

1 #!/bin/bash
2 ./lab12_2
3 case $? in
4 0) echo "number was 0";;
5 1) echo "number was positive";;
6 2) echo "number was negative";;
7 *) echo "error";;
8 esac

```

(Рис. 2.2)

Его работа:

```

parallels@nsfedorov:~$ g++ main.cpp -o main
parallels@nsfedorov:~$

```

(Рис. 2.3)

```
parallels@nsfedorov:~$ ./lab12_2.sh
input number
0
number was 0
parallels@nsfedorov:~$ ./lab12_2.sh
input number
1
number was positive
parallels@nsfedorov:~$ ./lab12_2.sh
input number
-4
number was negative
parallels@nsfedorov:~$
```

(Рис. 2.4)

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, передаётся в аргументы командной строки. Этот же командный файл удаляет все созданные им файлы (если они существуют).

Код скрипта:

```
1 #!/bin/bash
2 typeset -i count=0;
3 for prov in *.tmp
4 do rm $prov
5 done
6 for ((count=1;count<$1+1;count++))
7 do
8     touch $count.tmp
9 done
```

(Рис. 3.1)

Его работа:

```
parallels@nsfedorov:~$ ./lab12_3.sh 13
parallels@nsfedorov:~$ ls|grep .tmp
10.tmp
11.tmp
12.tmp
13.tmp
1.tmp
2.tmp
3.tmp
4.tmp
5.tmp
6.tmp
7.tmp
8.tmp
9.tmp
parallels@nsfedorov:~$ ./lab12_3.sh 5
parallels@nsfedorov:~$ ls|grep .tmp
1.tmp
2.tmp
3.tmp
4.tmp
5.tmp
parallels@nsfedorov:~$
```

(Рис. 3.2)

4. Написал 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 (использовав команду find).

Код скрипта:

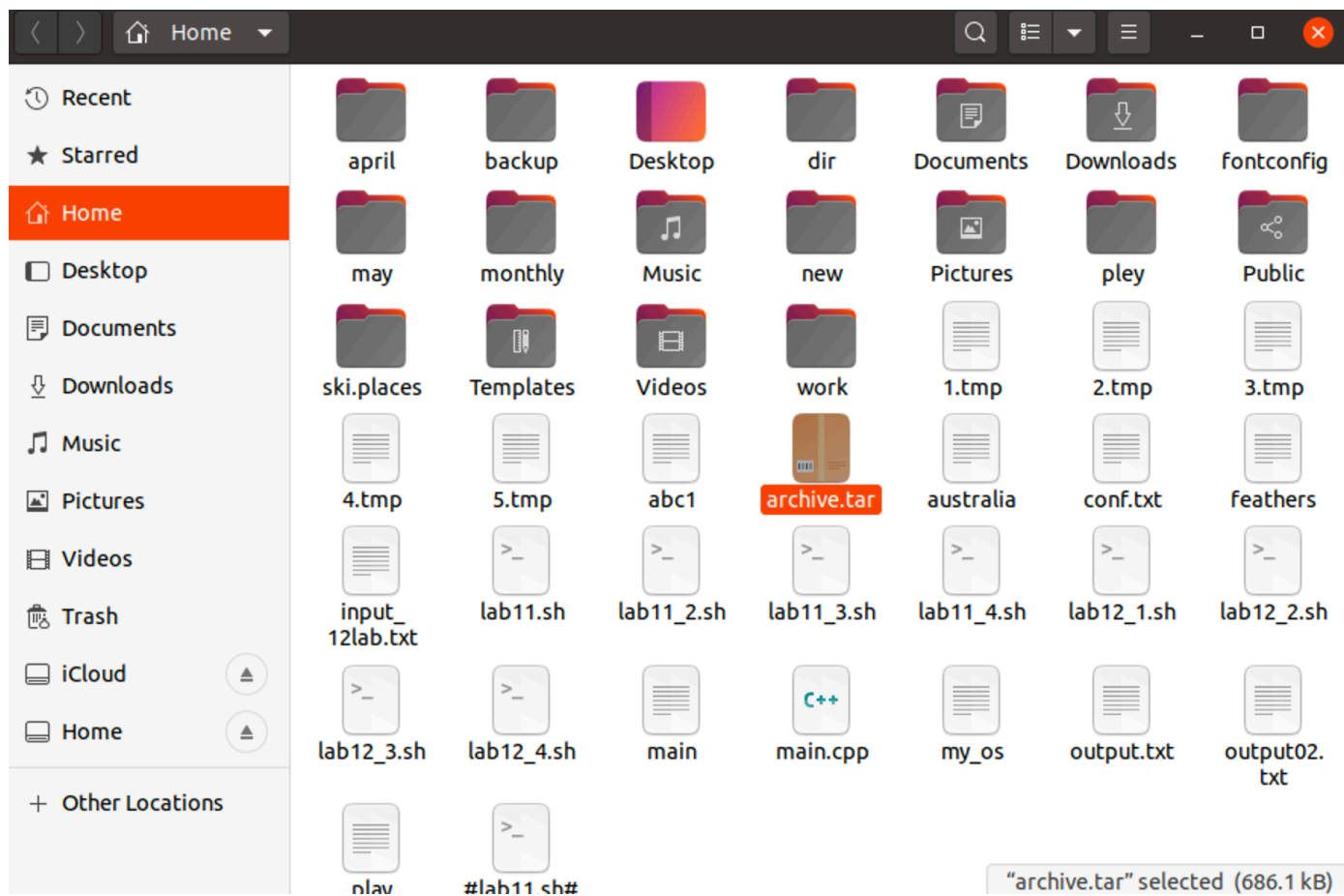
```
1 #!/bin/bash
2 cd $1
3 tar -cf archive.tar *
```

(Рис. 4.1)

Его работа:

```
parallels@nsfedorov:~$ touch lab12_4.sh
parallels@nsfedorov:~$ chmod +x lab12_4.sh
parallels@nsfedorov:~$ ./lab12_4.sh
parallels@nsfedorov:~$
```

(Рис. 4.2)

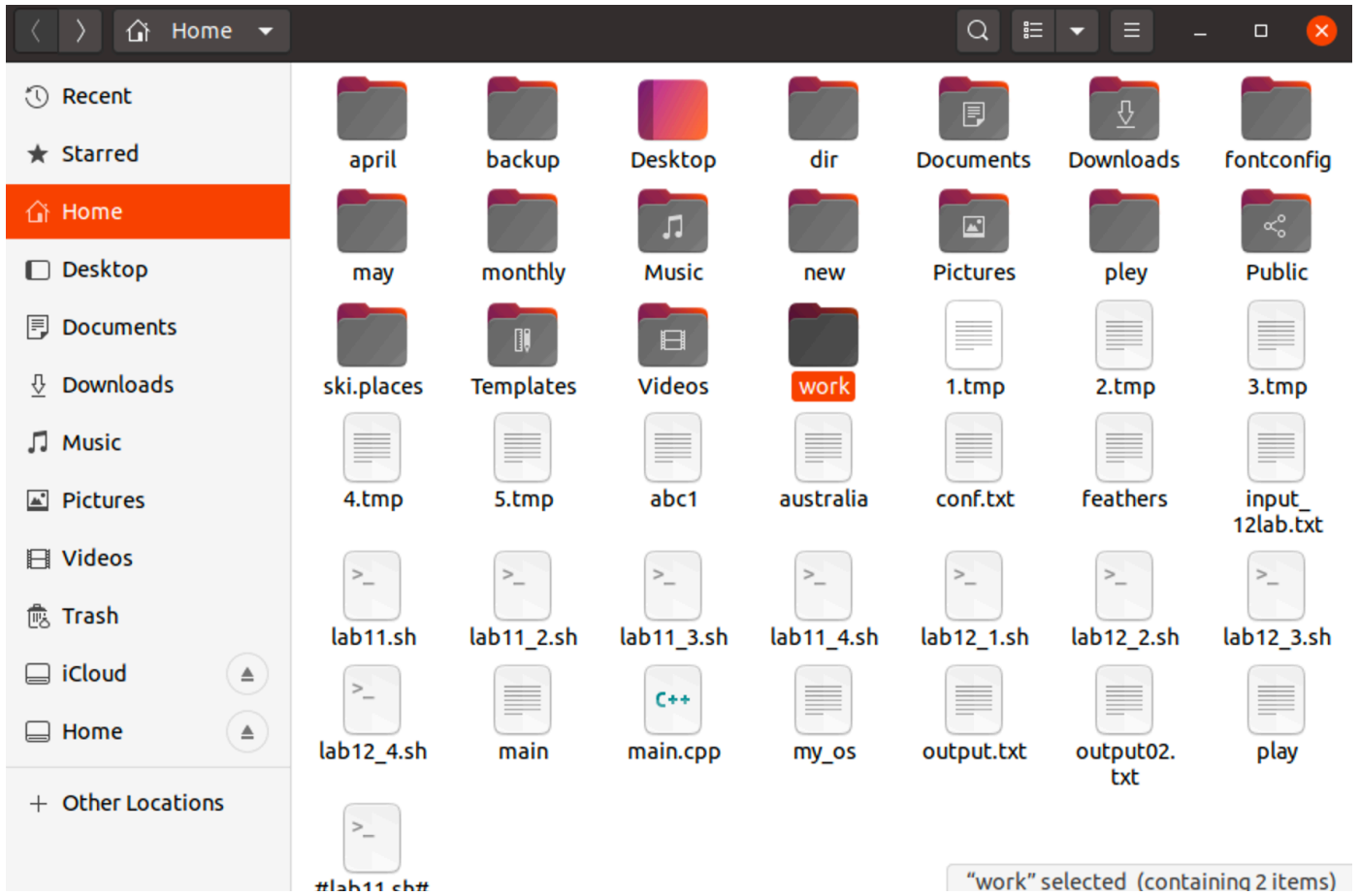


(Рис. 4.3)

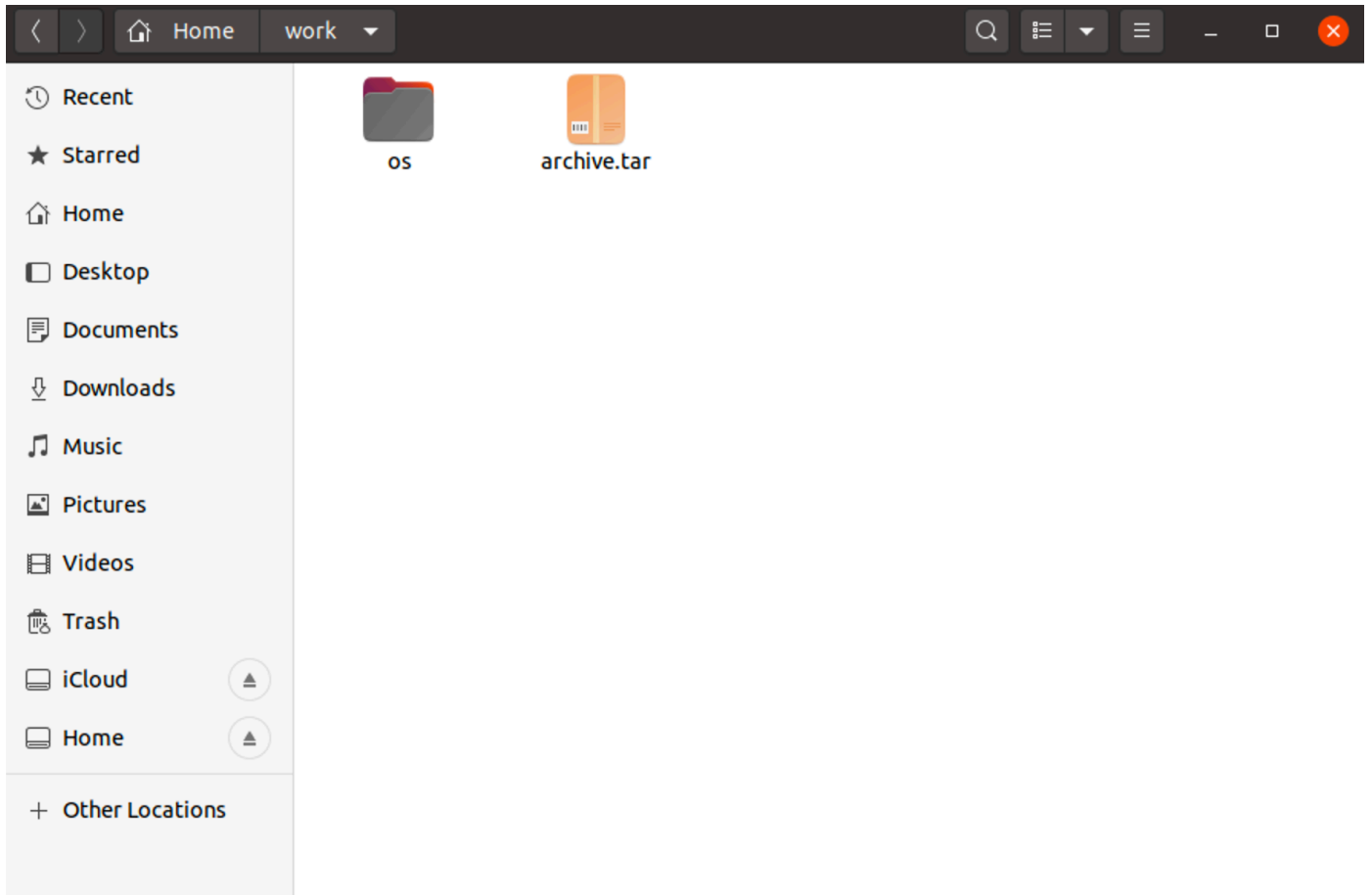
После модификации:

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

(Рис. 4.4)



(Рис. 4.5)



(Рис. 4.6)

Выводы

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

Использованные ресурсы

<https://ruvds.com/doc/bash.pdf>

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