

Отчёт по лабораторной работе №12

Дисциплина: Операционные системы

Ездаков Егор Андреевич

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	15

Список таблиц

Список иллюстраций

3.1	Написал первый скрипт	9
3.2	Проверил первый скрипт	10
3.3	Проверил первый скрипт	10
3.4	Создал файлы	10
3.5	Написал первый скрипт	11
3.6	Написал второй скрипт	11
3.7	Проверил второй скрипт	12
3.8	Написал третий скрипт	12
3.9	Проверил третий скрипт	13
3.10	Написал четвёртый скрипт	14
3.11	Проверил четвёртый скрипт	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 Задание

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;
- `-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`.

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 (использовать команду find).

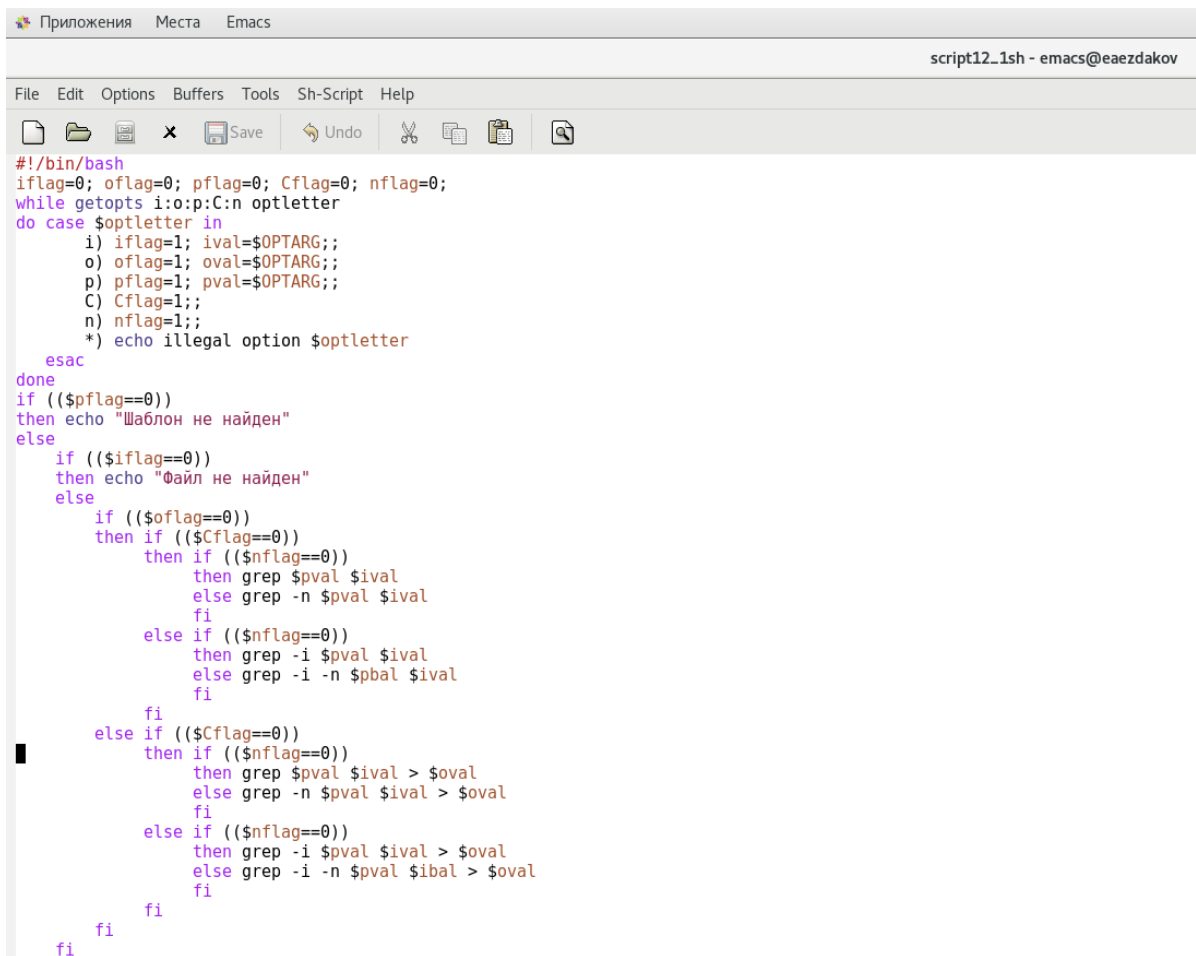
3 Выполнение лабораторной работы

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;
- `-С` — различать большие и малые буквы;
- `-п` — выдавать номера строк,

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

Для данной задачи я создал файл `script12_1.sh` и написал соответствующие скрипты (рис. -fig. 3.1).



```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:o:p:C:n 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 -i $pval $ival
else grep -i -n $pval $ival
fi
fi
else if (($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
```

Рис. 3.1: Написал первый скрипт

Далее я проверил работу написанного скрипта, используя различные опции, предварительно добавив право на исполнение файла (команда «`chmod +x *.sh`») и создав 2 файла, которые необходимы для выполнения программы: `a.txt` и `b.txt` (рис. -fig. 3.2) (рис. -fig. 3.3). Скрипт работает корректно.

```

[eaездakov@eaездakov ~]$ chmod +x script12_1.sh
[eaездakov@eaездakov ~]$ cat a.txt
debug mode
Debug mode
Dbug mode
[eaездakov@eaездakov ~]$ ./script12_1.sh -i a.txt -o b.txt -p debug -C -n
[eaездakov@eaездakov ~]$ cat b.txt
debug mode
Debug mode
[eaездakov@eaездakov ~]$ ./script12_1.sh -i a.txt -o b.txt -p debug -n
[eaездakov@eaездakov ~]$ cat b.txt
1:debug mode

```

Рис. 3.2: Проверил первый скрипт

```

debug mode
[eaездakov@eaездakov ~]$ ./script12_1.sh -i a.txt -C -n
Шаблон не найден
[eaездakov@eaездakov ~]$ ./script12_1.sh -o b.txt -p debug -C -n
Файл не найден
[eaездakov@eaездakov ~]$ █

```

Рис. 3.3: Проверил первый скрипт

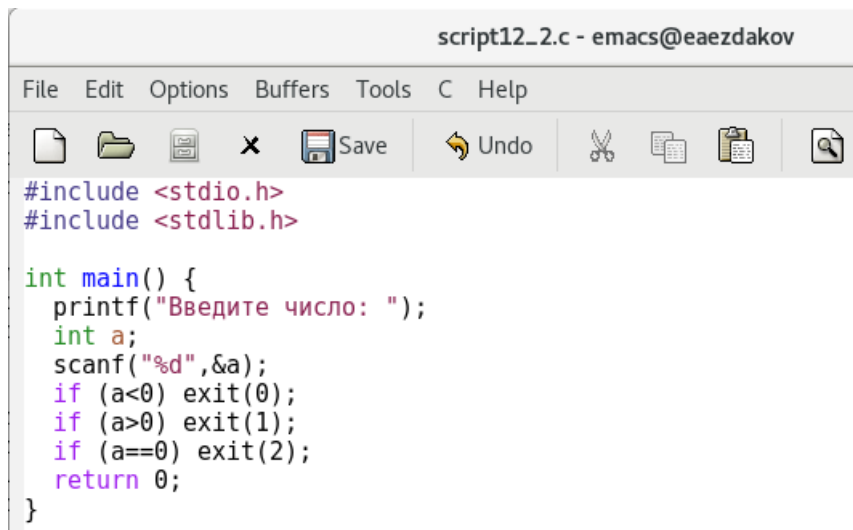
2. Написал на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. Для данной задачи я создал 2 файла: `script12_2.c` и `script12_2.sh` (рис. -fig. 3.4) и написал соответствующие скрипты (рис. -fig. 3.5) (рис. -fig. 3.6).

```

[eaездakov@eaездakov ~]$ touch script12_2.c script12_2.sh
[eaездakov@eaездakov ~]$ emacs &

```

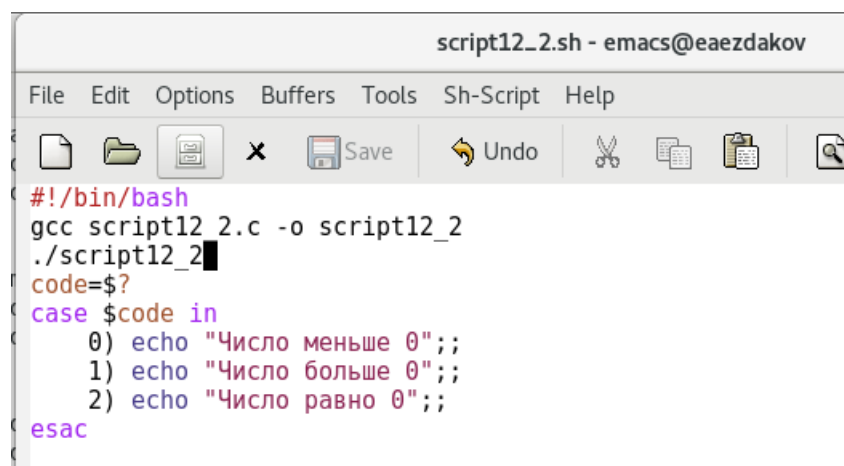
Рис. 3.4: Создал файлы



```
#include <stdio.h>
#include <stdlib.h>

int main() {
    printf("Введите число: ");
    int a;
    scanf("%d",&a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```

Рис. 3.5: Написал первый скрипт



```
#!/bin/bash
gcc script12_2.c -o script12_2
./script12_2
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число больше 0";;
    2) echo "Число равно 0";;
esac
```

Рис. 3.6: Написал второй скрипт

Далее я проверил работу написанных скриптов (команда «./script12_2.sh»), предварительно добавив право на исполнение файла (команда «chmod +x *.sh») (рис. -fig. 3.7). Скрипты работают корректно.

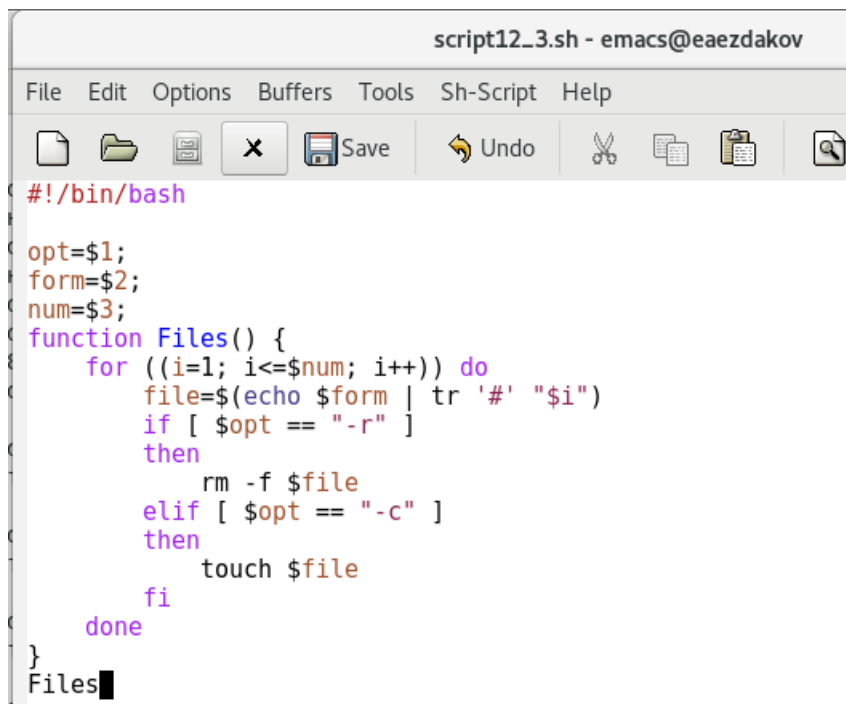
```

[eaездakov@eaездakov ~]$ chmod +x *.sh
[1]+  Done                  emacs
[eaездakov@eaездakov ~]$ ./script12_2.sh
Введите число: 1
Число больше 0
[eaездakov@eaездakov ~]$ ./script12_2.sh
Введите число: -1
Число меньше 0
[eaездakov@eaездakov ~]$ ./script12_2.sh
Введите число: 0
Число равно 0

```

Рис. 3.7: Проверил второй скрипт

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 должен уметь удалять все созданные им файлы (если они существуют). Для данной задачи я создал файл: script12_3.sh и написал соответствующий скрипт (рис. -fig. 3.8).



```

#!/bin/bash

opt=$1;
form=$2;
num=$3;
function Files() {
    for ((i=1; i<=$num; i++)) do
        file=$(echo $form | tr '#' "$i")
        if [ $opt == "-r" ]
        then
            rm -f $file
        elif [ $opt == "-c" ]
        then
            touch $file
        fi
    done
}
Files

```

Рис. 3.8: Написал третий скрипт

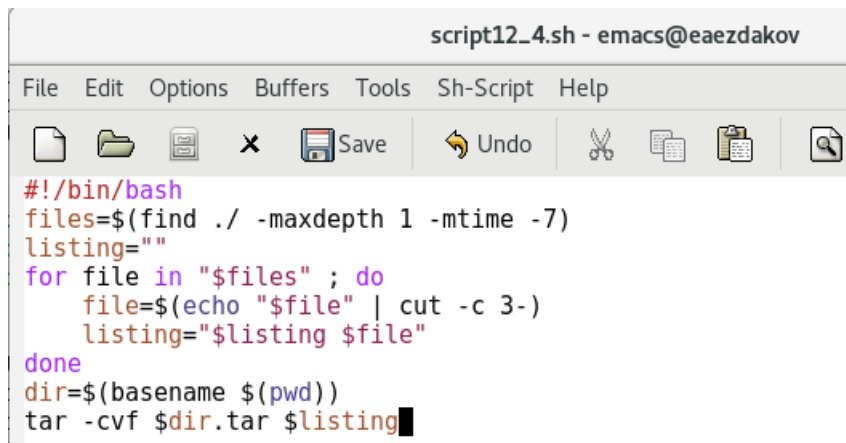
Далее я проверил работу написанного скрипта (команда «./script12_3.sh»), пред-

варительно добавив право на исполнение файла (команда «chmod +x *.sh»). Сначала я создал четыре файла (команда «./script12_3.sh -c file#.tmp 4»), удовлетворяющие условию задачи, а потом удалил их (команда «./script12_3.sh -r file#.tmp 4») (рис. -fig. 3.9). Скрипт работает корректно.

```
[eaezdakov@eaezdakov ~]$ chmod +x *.sh
[eaezdakov@eaezdakov ~]$ ls
a2.txt  backup.sh  conf.txt  folder1  monthly  pandoc
abc1    backup.sh~ equipment LICENSE  monthy.00 plans
a.txt   bin        feathers  may      my_os    play
backup b.txt      file.txt  mayh     pandoc   prog2.
[eaezdakov@eaezdakov ~]$ ./script12_3.sh -c file#.tmp 4
[eaezdakov@eaezdakov ~]$ ls
a2.txt  backup.sh  conf.txt  file2.tmp  folder1  monthl
abc1    backup.sh~ equipment file3.tmp  LICENSE  monthy
a.txt   bin        feathers  file4.tmp  may      my_os
backup b.txt      file1.tmp file.txt   mayh     pandoc
[eaezdakov@eaezdakov ~]$ ./script12_3.sh -r file#.tmp 4
[eaezdakov@eaezdakov ~]$ ls
a2.txt  backup.sh  conf.txt  folder1  monthly  pandoc
abc1    backup.sh~ equipment LICENSE  monthy.00 plans
a.txt   bin        feathers  may      my_os    play
backup b.txt      file.txt  mayh     pandoc   prog2.
[eaezdakov@eaezdakov ~]$ █
```

Рис. 3.9: Проверил третий скрипт

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 (использовать команду find). Для данной задачи я создал файл: script12_4.sh и написал соответствующий скрипт (рис. -fig. 3.10).



```
#!/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
```

Рис. 3.10: Написал четвёртый скрипт

Далее я проверил работу написанного скрипта (команды «`sudo ~/work/2020-2021/os-intro/laboratory/lab12/prog4.sh`» и «`tar -tf lab12.tar`»), предварительно добавив право на исполнение файла (команда «`chmod +x *.sh`») (рис. -fig. 3.11). Скрипт работает корректно.

```
[eaezdakov@eaezdakov ~]$ chmod +x script12_4.sh
[eaezdakov@eaezdakov ~]$ cd ~/debugfolder
[eaezdakov@eaezdakov debugfolder]$ ls -l
итого 6928
-rw-rw-r--. 1 eaezdakov eaezdakov      22 май 29 20:14 a2.txt
-rw-rw-r--. 1 eaezdakov eaezdakov       0 май 15 13:19 abc1
-rw-rw-r--. 1 eaezdakov eaezdakov      32 май 29 20:13 a.txt
-rw-rw-r--. 1 eaezdakov eaezdakov       0 май 15 13:24 feathers
-rw-rw-r--. 1 eaezdakov eaezdakov    3262 май 15 13:48 file.txt
drwxrwxr-x. 2 eaezdakov eaezdakov       6 май 15 12:49 folder1
-rwxrwxrwx. 1 eaezdakov eaezdakov 7081416 май 13 15:49 pandoc-crossref
drwxrwxr-x. 2 eaezdakov eaezdakov       6 май 15 12:39 plans
drwx--x--x. 3 eaezdakov eaezdakov      35 май 15 13:29 play
[eaezdakov@eaezdakov debugfolder]$ sudo ~/script12_4.sh
[sudo] пароль для eaezdakov:
a.txt
a2.txt
[eaezdakov@eaezdakov debugfolder]$ tar -tf debugfolder.tar
a.txt
a2.txt
[eaezdakov@eaezdakov debugfolder]$
```

Рис. 3.11: Проверил четвёртый скрипт

4 Выводы

В ходе выполнения данной лабораторной работы я изучил основы программирования в оболочке ОС UNIX и научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.