

Лабораторная работа №11

Дисциплина: операционные системы

Пронякова Ольга Максимовна

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Выполнение лабораторной работы | 7 |
| 4 | Выводы | 13 |
| | Список литературы | 14 |

Список иллюстраций

| | | |
|------|------------------------------|----|
| 3.1 | Код программы | 8 |
| 3.2 | Выполнение команды | 8 |
| 3.3 | Код программы | 9 |
| 3.4 | Выполнение команды | 9 |
| 3.5 | Код программы | 10 |
| 3.6 | Выполнение команды | 10 |
| 3.7 | Код программы | 11 |
| 3.8 | Выполнение команды | 11 |
| 3.9 | Выполнение команды | 11 |
| 3.10 | Выполнение команды | 12 |

Список таблиц

1 Цель работы

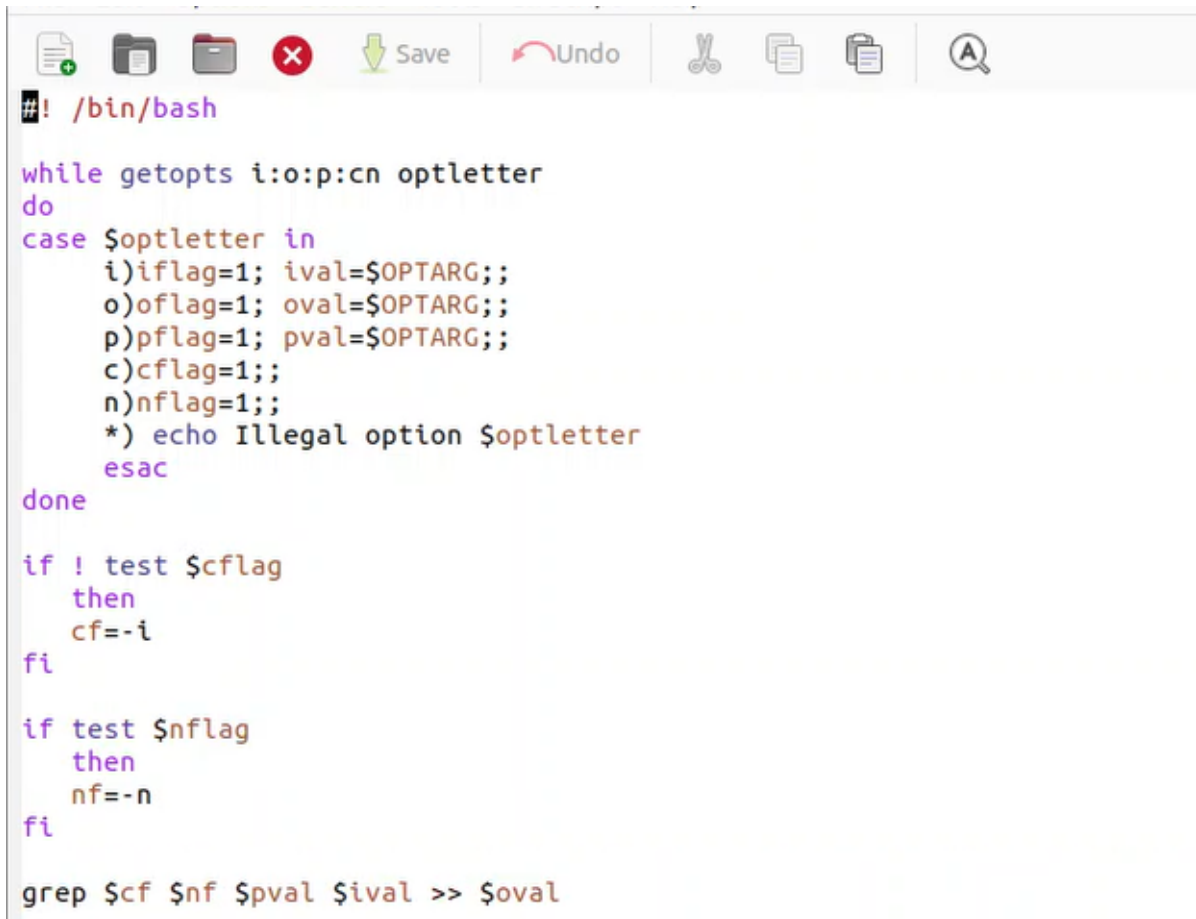
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

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; `-rшаблон` — 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 ∞ (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 `tag` 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

Используя команды `getopts` `grep`, пишу командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-rшаблон` — 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`.(рис. 3.1) (рис. 3.2).

A screenshot of a code editor window. The title bar shows icons for file operations (new, open, save, close) and editing (undo, redo, cut, copy, paste, search). The code is a shell script in a bash environment. It uses a while loop with getopt to parse command-line options. The options are: -i (iflag), -o (oflag), -p (pflag), -c (cflag), and -n (nflag). Each option sets a corresponding flag and a value. An asterisk (*) indicates an illegal option, which prints an error message. After the loop, it checks if the -c flag is set and sets cf=-i. Then it checks if the -n flag is set and sets nf=-n. Finally, it runs a grep command with the flags and values. The code is as follows:

```
#!/bin/bash

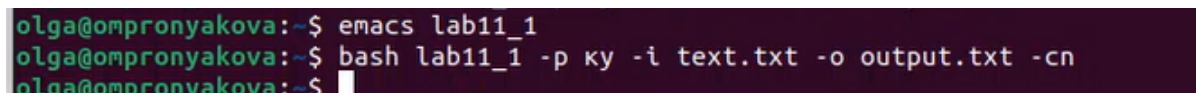
while getopts i:op:cn optletter
do
case $optletter in
i)iflag=1; ival=$OPTARG;;
o)oflag=1; oval=$OPTARG;;
p)pflag=1; pval=$OPTARG;;
c)cflag=1;;
n)nflag=1;;
*) echo Illegal option $optletter
esac
done

if ! test $cflag
then
cf=-i
fi

if test $nflag
then
nf=-n
fi

grep $cf $nf $pval $ival >> $oval
```

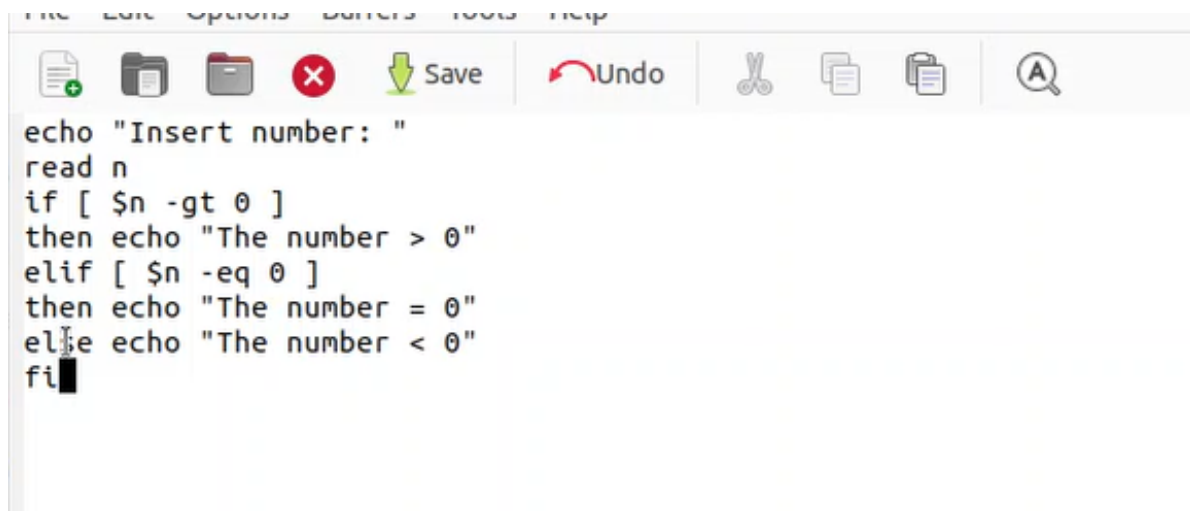
Рис. 3.1: Код программы

A screenshot of a terminal window. The prompt is 'olga@ompronyakova:~\$'. The user enters 'emacs lab11_1'. The prompt changes to 'olga@ompronyakova:~\$'. The user enters 'bash lab11_1 -p ку -i text.txt -o output.txt -cn'. The prompt changes to 'olga@ompronyakova:~\$' and a cursor is visible at the end of the line.

```
olga@ompronyakova:~$ emacs lab11_1
olga@ompronyakova:~$ bash lab11_1 -p ку -i text.txt -o output.txt -cn
olga@ompronyakova:~$
```

Рис. 3.2: Выполнение команды

Пишу 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 вызывает эту программу и, проанализировав с помощью команды `$?`, выдает сообщение о том, какое число было введено(рис. 3.3) (рис. 3.4).



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

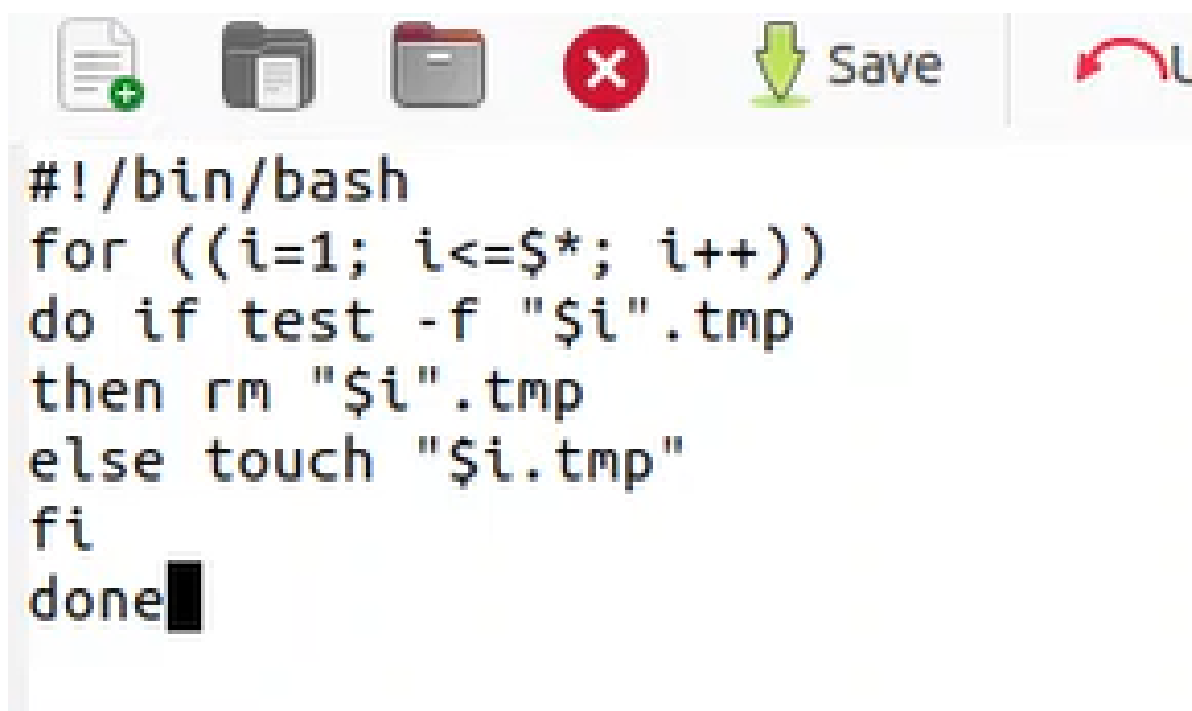
Рис. 3.3: Код программы



```
olga@ompronyakova:~$ emacs lab11_4  
olga@ompronyakova:~$ bash lab11_4  
Insert number:  
8  
The number > 0  
olga@ompronyakova:~$ bash lab11_4  
Insert number:  
0  
The number = 0  
olga@ompronyakova:~$ bash lab11_4  
Insert number:  
-2  
The number < 0
```

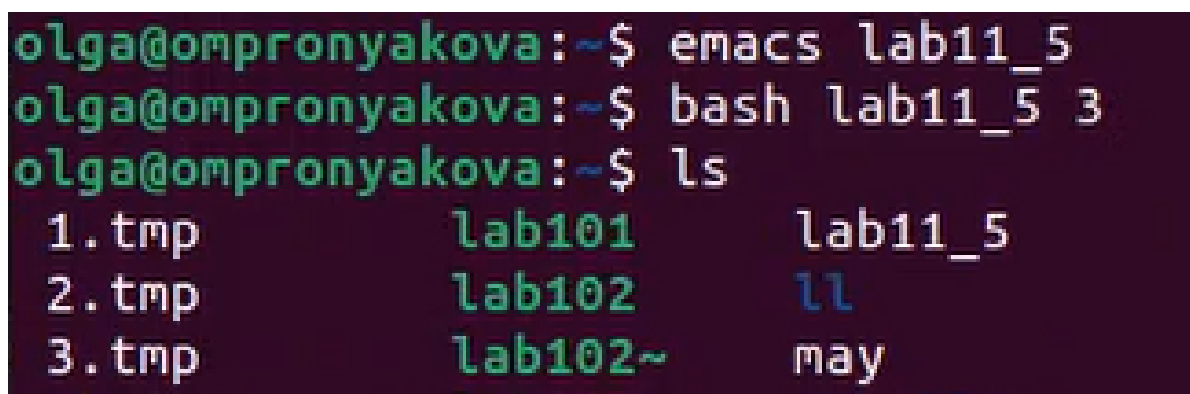
Рис. 3.4: Выполнение команды

Пишу командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ∞ (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)(рис. 3.5) (рис. 3.6).



```
#!/bin/bash
for ((i=1; i<= $*; i++))
do if test -f "$i".tmp
then rm "$i".tmp
else touch "$i".tmp
fi
done
```

Рис. 3.5: Код программы



```
olga@ompronyakova:~$ emacs lab11_5
olga@ompronyakova:~$ bash lab11_5 3
olga@ompronyakova:~$ ls
1.tmp      lab101      lab11_5
2.tmp      lab102      ll
3.tmp      lab102~     may
```

Рис. 3.6: Выполнение команды

Пишу командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицирую его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад(рис. 3.7) (рис. 3.8) (рис. 3.9) (рис. 3.10).



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

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

Рис. 3.7: Код программы



```
olga@ompronyakova:~$ emacs lab11_6
olga@ompronyakova:~$ chmod +x lab11_6
olga@ompronyakova:~$ ./lab11_6 /home/olga/
olga@ompronyakova:~$ ls
abc1          lab102      lab11_6      reports
archive.tar  lab102~    lab11_6~    ski.places
assets       lab103     ll           snap
australia    lab103~    may         text.txt
backup       lab104     monthly     text.txt~
bin          lab104~    my_os       work
blog         lab11      newdir      Видео
c.cpp.save   lab11_1    Olga1325.github.io  Документы
conf.txt     lab11_1~  output.txt  Загрузки
feathers     lab11_2    pandoc-2.19.2  Изображения
FILES.txt   lab11_3    pandoc-2.19.2-linux-amd64.tar.gz  Музыка
file.txt    lab11_3~  pandoc-crossref  Общедоступные
lab07.sh    lab11_4    pandoc-crossref.1  'Рабочий стол'
lab07.sh~   lab11_4~  pandoc-crossref-Linux.tar.xz  Шаблоны
lab101      lab11_5    play
```

Рис. 3.8: Выполнение команды



```
olga@ompronyakova:~$ mkdir tmp
```

Рис. 3.9: Выполнение команды

```

olga@ompronyakova:~$ tar -xf archive.tar -C /home/olga/tmp
olga@ompronyakova:~$ ls
abc1      lab102    lab11_6    reports
archive.tar lab102~   lab11_6~   ski.places
assets    lab103    ll          snap
australia lab103~   may        text.txt
backup    lab104    monthly    text.txt~
bin       lab104~   my_os      tmp
blog      lab11     newdir     work
c.cpp.save lab11_1   Olga1325.github.io Видео
conf.txt  lab11_1~ output.txt Документы
feathers  lab11_2   pandoc-2.19.2 Загрузки
FILES.txt lab11_3   pandoc-2.19.2-linux-amd64.tar.gz Изображения
file.txt  lab11_3~  pandoc-crossref Музыка
lab07.sh  lab11_4   pandoc-crossref.1 Общедоступные
lab07.sh~ lab11_4~  pandoc-crossref-Linux.tar.xz 'Рабочий стол'
lab101    lab11_5   play       Шаблоны
olga@ompronyakova:~$ cd tmp
olga@ompronyakova:~/tmp$ ls
backup lab101 lab102 lab102~ lab103 lab104 lab104~ snap work Загрузки
olga@ompronyakova:~/tmp$

```

Рис. 3.10: Выполнение команды

4 Выводы

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

Список литературы

1. Лабораторная работа №11