

Отчет по лабораторной работе №12

Дисциплина: Операционные системы

Чекалова Лилия Руслановна, ст.б. 1032201654

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	11
4	Библиография	12

List of Figures

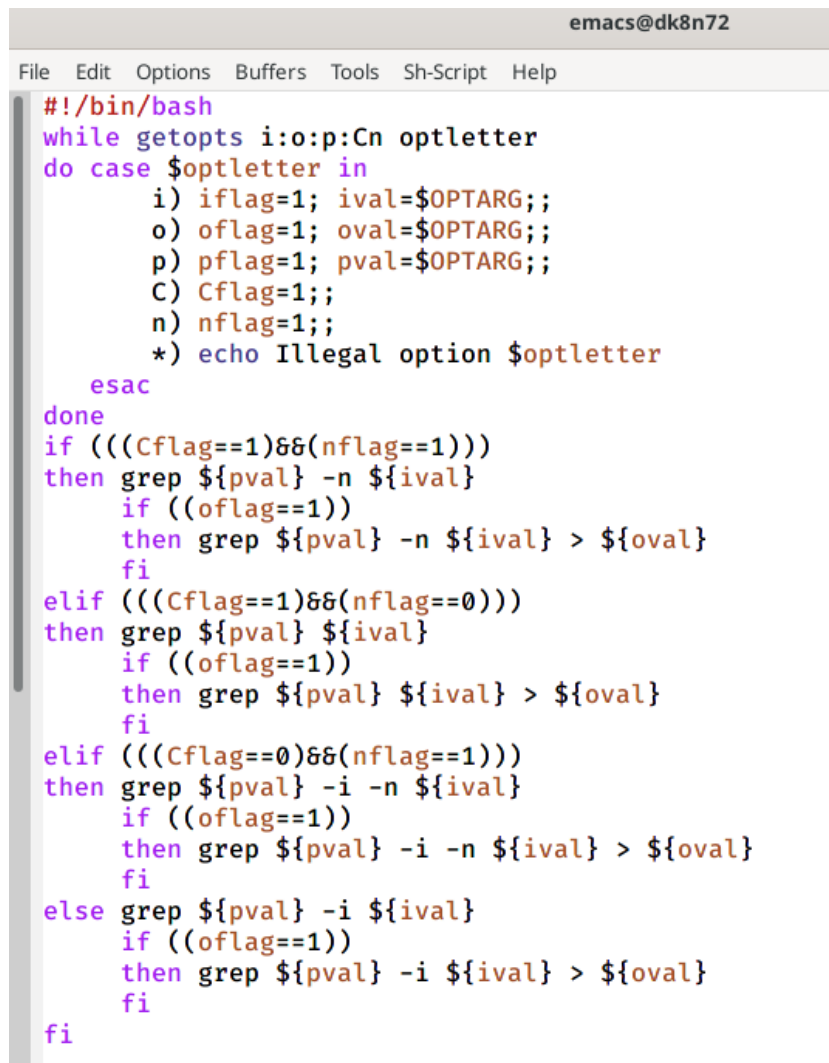
2.1	Использование getopts	6
2.2	Ввод текста	6
2.3	Результат работы	7
2.4	Прорама на Си	7
2.5	Командный файл	8
2.6	Проверка работы файла	8
2.7	Файл, создающий и удаляющий файлы	9
2.8	Создание файлов	9
2.9	Удаление файлов	9
2.10	Командный файл архивирования	10
2.11	Проверка создания архива	10

1 Цель работы

Изучение основ программирования в оболочке UNIX/Linux, приобретение навыков написания более сложных командных файлов с использованием логических управляющих конструкций и циклов.

2 Выполнение лабораторной работы

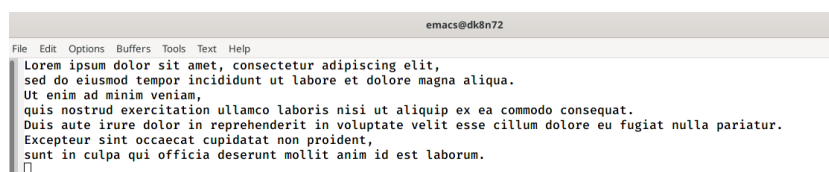
Пишу командный файл, анализирующий командную строку с ключами `-iinputfile` (чтение данных из указанного файла), `-ooutputfile` (запись данных в указанный файл), `-rшаблон` (поиск по указанному шаблону), `-C` (различение больших и малых букв) и `-n` (нумерация строк). Для этого в цикле `while` использую команду `getopts` с конструкцией `case`, а затем проверяю полученные результаты условным оператором `if-elif-else`, осуществляя командой `grep` соответствующий поиск и вывод на экран и/или в файл (рис. 2.1)



```
emacs@dk8n72
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
while getopts i:o:p: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 (((Cflag==1)&&(nflag==1)))
then grep ${pval} -n ${ival}
    if ((oflag==1))
    then grep ${pval} -n ${ival} > ${oval}
    fi
elif (((Cflag==1)&&(nflag==0)))
then grep ${pval} ${ival}
    if ((oflag==1))
    then grep ${pval} ${ival} > ${oval}
    fi
elif (((Cflag==0)&&(nflag==1)))
then grep ${pval} -i -n ${ival}
    if ((oflag==1))
    then grep ${pval} -i -n ${ival} > ${oval}
    fi
else grep ${pval} -i ${ival}
    if ((oflag==1))
    then grep ${pval} -i ${ival} > ${oval}
    fi
fi
```

Figure 2.1: Использование getopts

Создаю текстовый файл, по которому будет осуществляться поиск, записываю в него фрагмент текста (рис. 2.2)



```
emacs@dk8n72
File Edit Options Buffers Tools Text Help
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.
```

Figure 2.2: Ввод текста

Запускаю командный файл с различными опциями и сравниваю полученные результаты. Все ключи работают корректно (рис. 2.3)

```

lrcchekalova@dk8n72 ~ $ emacs
lrcchekalova@dk8n72 ~ $ touch noltext.txt
lrcchekalova@dk8n72 ~ $ emacs
lrcchekalova@dk8n72 ~ $ bash nol.sh -inoltext.txt -onolt.txt -plor
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
lrcchekalova@dk8n72 ~ $ bash nol.sh -inoltext.txt -onolt.txt -plor -n
1:Lorem ipsum dolor sit amet, consectetur adipiscing elit,
2:sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
5:Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
lrcchekalova@dk8n72 ~ $ bash nol.sh -inoltext.txt -onolt.txt -plor -C
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
lrcchekalova@dk8n72 ~ $ bash nol.sh -inoltext.txt -onolt.txt -pLor -C
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
lrcchekalova@dk8n72 ~ $

```

Figure 2.3: Результат работы

Пишу на языке Си программу, сравнивающую введенное число с нулем: если оно равно нулю, она завершается функцией `exit(0)`, если больше нуля - `exit(1)`, если меньше нуля - `exit(2)` (рис. 2.4)

```

emacs@dk8n72
File Edit Options Buffers Tools C Help
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int x;
    printf("Input number: ");
    scanf("%i", &x);
    if (x==0)
        exit(0);
    else if (x>0)
        exit(1);
    else exit(2);
    return 3;
}

```

Figure 2.4: Прорамма на Си

Пишу командный файл, в котором запускаю программу на Си и анализирую ее результат с помощью конструкции `case` и специальной переменной `$?`. На основе этого вывожу сообщение о том, которое число было введено (рис. 2.5)

```
#!/bin/bash
gcc -o num2 no2.c
./num2
case $? in
    0) echo 'number equals 0';;
    1) echo 'number is above 0';;
    2) echo 'number is below 0';;
esac
```

Figure 2.5: Командный файл

Запускаю командный файл и проверяю его работу на разных числах (рис. 2.6)

```
lrchekalova@dk8n72 ~ $ bash no2.sh
Input number: -3
number is below 0
lrchekalova@dk8n72 ~ $ bash no2.sh
Input number: 4
number is above 0
lrchekalova@dk8n72 ~ $ bash no2.sh
Input number: 0
number equals 0
lrchekalova@dk8n72 ~ $
```

Figure 2.6: Проверка работы файла

Пишу командный файл, создающий указанное количество файлов, пронумерованных по порядку от 1, и способный удалять созданные им файлы. Для этого использую команду `getopts` и, проанализировав результаты ее работы условным оператором `if-else`, создаю или удаляю указанное количество файлов командами `touch` и `rm` (рис. 2.7)


```
emacs@dk8n72
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
let dflag=0;
while getopts m:d optletter
do case $optletter in
    m) mflag=1; mval=$OPTARG;;
    d) dflag=1;;
    *) echo Illegal option $optletter
    esac
done
if ((dflag==0))
then for ((i=1;i<=mval;i++))
do touch ${i}.txt
done
else for ((i=1;i<=mval;i++))
do rm ${i}.txt
done
fi
```

Figure 2.7: Файл, создающий и удаляющий файлы

Запускаю файл с ключом -m3, чтобы он создал три текстовых файла (рис. 2.8)

```
irc@kalovadk8n72 ~$ emacs
irc@kalovadk8n72 ~$ bash no3.sh -m3
irc@kalovadk8n72 ~$ ls
1.txt  a1.asm  a2.o      no1.sh  no2.c~  no3.sh~  public_html  script4.sh~  typescript  Загрузки  Общедоступные
2.txt  a1.o    backup    no1.txt  no2.c~  no3.sh~  script2.sh~  script4.sh~  work        Видео     Изображения
3.txt  a2     backupscript.sh  no1.txt  no2.sh  num2     script3.sh~  script.sh   Documents  Музыка    'Рабочий стол'
a1     a2.asm  GNUstep   no1.txt  no2.sh~  public   script3.sh~  tmp         Документы  Музыка    'Рабочий стол'
irc@kalovadk8n72 ~$ ls
1.txt  a1.asm  a2.o      no1.sh  no2.c~  no3.sh~  public_html  script4.sh~  typescript  Загрузки  Общедоступные
2.txt  a1.o    backup    no1.txt  no2.c~  no3.sh~  script2.sh~  script4.sh~  work        Видео     Изображения
3.txt  a2     backupscript.sh  no1.txt  no2.sh  num2     script3.sh~  script.sh   Documents  Музыка    'Рабочий стол'
a1     a2.asm  GNUstep   no1.txt  no2.sh~  public   script3.sh~  tmp         Документы  Музыка    'Рабочий стол'
irc@kalovadk8n72 ~$
```

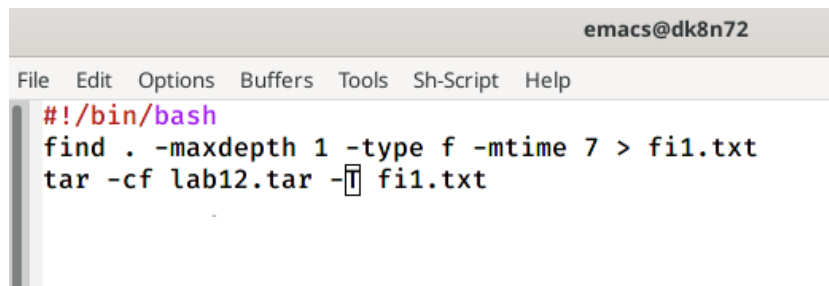
Figure 2.8: Создание файлов

Вновь запускаю файл, теперь с ключами -m3 и -d, чтобы удалить созданные файлы (рис. 2.9)

```
irc@kalovadk8n72 ~$ bash no3.sh -m3 -d
irc@kalovadk8n72 ~$ ls
a1     a2.asm  GNUstep   no1.txt  no2.sh~  public   script3.sh~  tmp         Документы  Музыка    'Рабочий стол'
a1.asm  a2.o    backup    no1.txt  no2.c~  no3.sh~  public_html  script4.sh~  typescript  Загрузки  Общедоступные
a1.o    backup  backupscript.sh  no1.txt  no2.sh  num2     script3.sh~  script.sh   Видео     Изображения
a2     backupscript.sh  no1.txt  no2.sh~  public   script3.sh~  tmp         Документы  Музыка    'Рабочий стол'
irc@kalovadk8n72 ~$
```

Figure 2.9: Удаление файлов

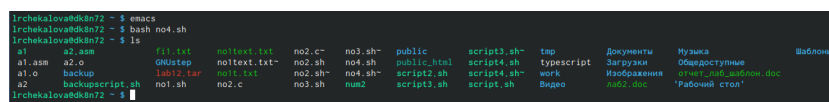
Пишу командный файл, добавляющий в архив все файлы текущей директории, измененные менее недели назад. Для этого совершаю поиск командой `find` с опцией `-mtime 7`, указывающей последнее изменение в днях, и записываю результат в текстовый файл `fl1.txt`. Затем командой `tar` с опцией `-T` добавляю в архив `lab12.tar` все файлы, указанные в текстовом файле (рис. 2.10)



```
emacs@dk8n72
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
find . -maxdepth 1 -type f -mtime 7 > fi1.txt
tar -cf lab12.tar -T fi1.txt
```

Figure 2.10: Командный файл архивирования

Запускаю командный файл и командой ls проверяю наличие созданного архива в папке (рис. 2.11)



```
lrchekalova@dk8n72 ~$ emacs
lrchekalova@dk8n72 ~$ bash no4.sh
lrchekalova@dk8n72 ~$ ls
a1      a2.asm  /fi1.txt  no1.sh  no2.c~  no3.sh~  public  script3.sh~  tmp      Документы  Музыка  Шаблоны
a1.asm  a2.o    GNUstep  no1.sh  no2.sh  no4.sh  public_html  script4.sh  typescript  Загрузки  Общедоступные
a1.o    backup  lab12.tar no1.txt no2.sh~ no4.sh~  script2.sh  script4.sh~  work      Изображения  отключ_наб_адаптов.doc
a2      backupscript.sh no1.sh  no2.c   no3.sh  num2     script3.sh  script.sh   Видео     lab01.doc  "Рабочий стол"
```

Figure 2.11: Проверка создания архива

3 Выводы

После выполнения данной лабораторной работы я научилась писать более сложные командные файлы с использованием условных конструкций if-elif-else и case, циклов for и while и команды getopt.

4 Библиография

1. Команда find в Linux: <https://losst.ru/komanda-find-v-linux>
2. Поиск внутри файлов Linux: <https://losst.ru/gerp-poisk-vnutri-fajlov-v-linux>
3. Теоретические материалы к лабораторной работе: https://esystem.rudn.ru/pluginfile.php/1111111/mod_resource/content/1/lab_shell_prog_1.pdf