

Лабораторная работа-11

**Программирование в командном процессоре ОС UNIX. Ветвления и
циклы**

Доленко Дарья Васильевна НБИбд-01-21

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Вывод	14
4	Ответы на контрольные вопросы:	15

Список иллюстраций

2.1	Пример работы программы	5
2.2	Содержание текстового файла	6
2.3	Текст программы	7
2.4	Пример работы программы	8
2.5	Файл sh	8
2.6	Файл C	9
2.7	Пример работы программы (состояние до применения)	9
2.8	Пример работы программы (после применения)	10
2.9	Текст программы	10
2.10	До применения программы	11
2.11	После применения программы	12
2.12	Текст программы	13
2.13	Команды архивации	13

1 Цель работы

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

2 Выполнение лабораторной работы

Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами: `-iinputfile`; `-ooutputfile`; `-ршаблон`; `-C`; `-n`, затем ищет в указанном файле нужные строки, определяемые ключом `-р`. (рис. 2.1 2.2 2.3)

```
[dvdolenko@fedora ~]$ bash lab11.sh -ilab11.txt -olab11-1.txt -pice
[dvdolenko@fedora ~]$ cat ~/lab11.txt
Дале-далеко, в той стране, где всходит солнце
Ранним утром милый больше не проснётся
Льётся алая река, лиса смеётся
С восточным ветром пусть сказание несётся
В забытом ныне городе жила девица
Великим самураям от любви не спится
Лица жадные горят. Как не влюбиться?
Но в хитрых девичьих глазах кошмар таится
Не беги за нею, глупый
По её следам идут лишь трупы
Зубы крепче, чем металл
Ты попал!
[dvdolenko@fedora ~]$ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p C -C -n
[dvdolenko@fedora ~]$ cat ~/lab11-1.txt
Дале-далеко, в той стране, где всходит солнце
Ранним утром милый больше не проснётся
Льётся алая река, лиса смеётся
С восточным ветром пусть сказание несётся
Великим самураям от любви не спится
Лица жадные горят. Как не влюбиться?
Но в хитрых девичьих глазах кошмар таится
По её следам идут лишь трупы
[dvdolenko@fedora ~]$ ./lab11.sh -i ~/lab11.txt -o ~/lab11-1.txt -p C -n
[dvdolenko@fedora ~]$ cat ~/lab11-1.txt
4:С восточным ветром пусть сказание несётся
[dvdolenko@fedora ~]$ ./lab11.sh -i ~/lab11.txt -C -n
./lab11.sh: строка 13: ((: --: синтаксическая ошибка: ожидается operand (перевор...
```

Рис. 2.1: Пример работы программы

```
1 Далё-далеко, в той стране, где всходит солнце
2 Ранним утром милый больше не проснётся
3 Льётся алая река, лиса смеётся
4 С восточным ветром пусть сказание несётся
5 В забытом ныне городе жила девица
6 Великим самураем от любви не спится
7 Лица жадные горят. Как не влюбиться?
8 Но в хитрых девичьих глазах кошмар таится
9 Не беги за нею, глупый
10 По её следам идут лишь трупы
11 Зубы крепче, чем металл
12 Ты попал!
```

Рис. 2.2: Содержание текстового файла

```

1 #!/bin/bash
2 inflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
3 while getopts i:o:p:C:n optletter
4 do case $optletter in
5     i) iflag=1; ival=$OPTARG;;
6     o) oflag=1; oval=$OPTARG;;
7     p) pflag=1; pval=$OPTARG;;
8     C) Cflag=1;;
9     n) nflag=1;;
10    *) echo illegal option $optletter
11    esac
12 done
13 if (($pflag==0))
14 then echo "not found"
15 else
16     if (($iflag==0))
17     then echo "file not found"
18     else
19         if (($oflag==0))
20         then if ((Cflag==0))
21             then if ((nflag==0))
22                 then grep $pval $ival
23                 else grep -n $pval $ival
24                 fi
25             else if ((nflag==0))
26                 then grep -i $pval $ival
27                 else grep -i -n $pval $ival
28                 fi
29             fi
30         else if ((Cflag==0))
31             then if ((nflag==0))
32                 then grep $pval $ival > $oval
33                 else grep -n $pval $ival > $oval
34                 fi
35             else if ((nflag==0))
36                 then grep -i $pval $ival > $oval
37                 else grep -i -n $pval $ival > $oval
38                 fi
39             fi
40         fi
41     fi
42 fi

```

Рис. 2.3: Текст программы

Написала 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. (рис. 2.4 2.5 2.6)

```
[dvdolenko@fedora ~]$ ./l11.sh
write number: 67
>0
[dvdolenko@fedora ~]$ ./l11.sh
write number: -90
<0
[dvdolenko@fedora ~]$ ./l11.sh
write number: 0
=0
```

Рис. 2.4: Пример работы программы

```
1 #!/bin/bash
2
3 gcc l11.c -o l11
4 ./l11
5 code=$?
6 case $code in
7     0) echo "<0";;
8     1) echo ">0";;
9     2) echo "=0";;
10 esac
```

Рис. 2.5: Файл sh


```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     printf("write number: ");
6     int n;
7     scanf("%d", &n);
8     if (n<0) exit(0);
9     if (n>0) exit(1);
10    if (n==0) exit(2);
11    return 0;
12 }

```

Рис. 2.6: Файл С

Написала 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). (рис. 2.7 2.8 2.9)

```

[dvdolenko@fedora ~]$ touch lll1.sh
[dvdolenko@fedora ~]$ chmod +x *.sh
[dvdolenko@fedora ~]$ ls
abcl.cpp      lll1      may        ski.plases  Музыка
australia    lll.c     monthly    snap        Общедоступные
backup       lll1.sh   my_os      text.txt    'Рабочий стол'
backup.sh    '#lab07.sh#'  play      tutorial    c.sh
b.sh        lab07.sh  reports    work        c.sh~
b.sh~       lab11-1.txt  script.sh  Видео      Шаблоны
conf.txt     lab11.sh  script.sh~  Документы
feathers     lab11.txt  scriptttt.sh  Загрузки
fille.txt    lll1.sh   scriptttt.sh~  Изображения

```

Рис. 2.7: Пример работы программы (состояние до применения)

```
[dvdolenko@fedora ~]$ ./l11.sh -c VD#.txt 8
[dvdolenko@fedora ~]$ la
bash: la: command not found...
[dvdolenko@fedora ~]$ ls
abcl.cpp      l11.c      my_os      tutorial    Видео
australia    l11.sh     play       VD1.txt     Документы
backup       '#lab07.sh#' reports     VD2.txt     Загрузки
backup.sh    lab07.sh   script.sh  VD3.txt     Изображения
b.sh        lab11-1.txt script.sh~  VD4.txt     Музыка
b.sh~       lab11.sh   scriptttt.sh VD5.txt     Общедоступные
conf.txt     lab11.txt  scriptttt.sh~ VD6.txt     'Рабочий стол'
feathers     l11.sh     ski.plases VD7.txt     c.sh
fille.txt    may        snap       VD8.txt     c.sh~
l11          monthly    text.txt   work        Шаблоны
```

Рис. 2.8: Пример работы программы (после применения)

```
1 #!/bin/bash
2
3 opt=$1;
4 form=$2;
5 num=$3;
6 function Files() {
7     for ((i=1; i<=$num; i++)) do
8         file=$(echo $form | tr '#' 'i')
9         if [ $opt == "-r" ]
10            then
11                rm -f $file
12            elif [ $opt == "-c" ]
13                then
14                    touch $file
15            fi
16        done
17    }
18 Files
```

Рис. 2.9: Текст программы

Написала командный файл, который с помощью команды tar запаковывает архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find). (рис. 2.10 2.11 2.12 2.13)

```
dvddenko@fedora:~  
[dvddenko@fedora ~]$ touch llllll.sh  
[dvddenko@fedora ~]$ chmod +x *.sh  
[dvddenko@fedora ~]$ ls -l  
итого 96  
-rw-rw-r--. 1 dvddenko dvddenko 93 мая 11 12:39 abc1.cpp  
drwxr--r--. 1 dvddenko dvddenko 0 мая 4 15:59 australia  
drwxrwxr-x. 1 dvddenko dvddenko 20 мая 19 15:36 backup  
-rwxrwxr-x. 1 dvddenko dvddenko 76 мая 19 15:33 backup.sh  
-rwxrwxr-x. 1 dvddenko dvddenko 236 мая 19 15:53 b.sh  
-rwxrwxr-x. 1 dvddenko dvddenko 0 мая 19 15:48 b.sh~  
-rw-rw-r--. 1 dvddenko dvddenko 805 мая 4 16:39 conf.txt  
-rw-rw-r--. 1 dvddenko dvddenko 0 мая 4 15:59 feathers  
-rw-rw-r--. 1 dvddenko dvddenko 2784 мая 4 16:35 fille.txt  
-rwxrwxr-x. 1 dvddenko dvddenko 23832 мая 28 14:35 lll  
-rw-rw-r--. 1 dvddenko dvddenko 179 мая 28 14:39 lll.c  
-rwxrwxr-x. 1 dvddenko dvddenko 111 мая 28 14:30 lll.sh  
-rw-rw-r--. 1 dvddenko dvddenko 111 мая 18 16:51 '#lab07.sh#'  
-rwxrwxr-x. 1 dvddenko dvddenko 98 мая 18 15:43 lab07.sh  
-rw-rw-r--. 1 dvddenko dvddenko 80 мая 28 14:19 lab11-1.txt  
-rwxrwxr-x. 1 dvddenko dvddenko 849 мая 28 14:26 lab11.sh  
-rw-rw-r--. 1 dvddenko dvddenko 711 мая 28 14:26 lab11.txt  
-rwxrwxr-x. 1 dvddenko dvddenko 231 мая 28 14:50 llll.sh  
-rwxrwxr-x. 1 dvddenko dvddenko 0 мая 28 14:54 llllll.sh
```

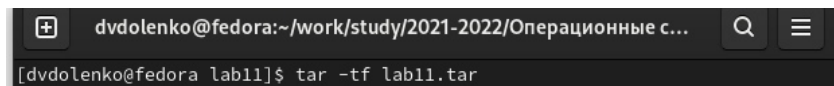
Рис. 2.10: До применения программы

```
[dvdolenko@fedora ~]$ sudo ~/lll111.sh
Изображения/
Изображения/Снимок экрана от 2022-04-27 10-01-33.png
Изображения/Снимок экрана от 2022-04-27 10-01-41.png
Изображения/Снимок экрана от 2022-05-11 08-56-56.png
Изображения/Снимок экрана от 2022-05-11 09-44-51.png
Изображения/Снимок экрана от 2022-05-11 10-16-46.png
Изображения/Снимок экрана от 2022-05-11 10-18-56.png
Изображения/Снимок экрана от 2022-05-11 10-22-03.png
Изображения/Снимок экрана от 2022-05-11 10-23-10.png
Изображения/Снимок экрана от 2022-05-11 10-25-25.png
Изображения/Снимок экрана от 2022-05-11 11-00-06.png
Изображения/Снимок экрана от 2022-05-11 11-08-37.png
Изображения/Снимок экрана от 2022-05-11 12-40-01.png
Изображения/Снимок экрана от 2022-05-11 15-33-42.png
Изображения/Снимок экрана от 2022-05-28 14-22-34.png
Изображения/Снимок экрана от 2022-05-28 14-36-21.png
Изображения/Снимок экрана от 2022-05-28 14-53-20.png
.vboxclient-clipboard.pid
.vboxclient-seamless.pid
.vboxclient-draganddrop.pid
.bash_history
lab11-1.txt
lab11.txt
lab11.sh
lll.sh
lll
lll.c
lll11.sh
VD1.txt
VD2.txt
VD3.txt
VD4.txt
VD5.txt
VD6.txt
VD7.txt
VD8.txt
lll111.sh
[dvdolenko@fedora ~]$
```

Рис. 2.11: После применения программы

```
1 #!/bin/bash
2
3 files=$(find ./ -maxdepth 1 -mtime -7)
4 listing=""
5 for file in "$files" ; do
6     file=$(echo "$file" | cut -c 3-)
7     listing="$listing $file"
8 done
9 dir=$(basename $(pwd))
10 tar -cvf $dir.tar $listing
```

Рис. 2.12: Текст программы



A terminal window with a dark background. The title bar shows the user 'dvdolenko' on a 'fedora' machine, the current directory '~/work/study/2021-2022/Операционные с...', and search and menu icons. The terminal prompt is '[dvdolenko@fedora lab11]\$'. The command 'tar -tf lab11.tar' has been entered and executed.

```
dvdolenko@fedora:~/work/study/2021-2022/Операционные с...
[dvdolenko@fedora lab11]$ tar -tf lab11.tar
```

Рис. 2.13: Команды архивации

3 Вывод

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

4 Ответы на контрольные вопросы:

1. Команда `getopts` является встроенной командой командной оболочки `bash`, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.
2. При генерации имен используют метасимволы:
произвольная (возможно пустая) последовательность символов; `?` один произвольный символ; `[...]` любой из символов, указанных в скобках перечислением и/или с указанием диапазона; `cat f*` выдаст все файлы каталога, начинающиеся с `"f"`; `cat f` выдаст все файлы, содержащие `"f"`; `cat program.?` выдаст файлы данного каталога с однобуквенными расширениями, скажем `"program.c"` и `"program.o"`, но не выдаст `"program.com"`; `cat [a-d]*` выдаст файлы, которые начинаются с `"a"`, `"b"`, `"c"`, `"d"`. Аналогичный эффект дадут и команды `"cat [abcd]"` и `"cat [bdac]"`.
3. Операторы `&&` и `||` являются управляющими операторами. Если в командной строке стоит `command1 && command2`, то `command2` выполняется в том, и только в том случае, если статус выхода из команды `command1` равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид `command1 || command2`, то команда `command2` выполняется тогда, и только тогда, когда статус выхода из команды `command1` отличен от нуля.

4. Оператор `break` завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.
5. Команда `true` всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда `false` всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое другое значение – ложь. Программа `true` – всегда завершается с кодом 0, `false` – всегда завершается с кодом 1.
6. Введенная строка означает условие существования файла `mans/i.$s`
7. Цикл `While` выполняется до тех пор, пока указанное в нем условие истинно. Когда указанное условие становится ложным - цикл завершается. Цикл `Until` выполняется до тех пор, пока указанное в нем условие ложно.