

Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Лабораторная работа №11

Азимов М.

22 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Азимов Миразим
- студент 1 курса, группа НММбд-01-22
- Российский университет дружбы народов



Вводная часть

- Командный процессор ОС UNIX
- Командные файлы

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

- Ознакомиться с теоретическим материалом.
- Выполнить упражнения.
- Ответить на контрольные вопросы.

Выполнение лабораторной работы №11

Первая программа

```
Открыть ▾ lab11_1.sh
```

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
while getopts :i:op:cn optletter
do case $optletter in
  i) iflag=1; sval=SOFTARG;;
  o) oflag=1; sval=SOFTARG;;
  p) pflag=1; sval=SOFTARG;;
  c) cflag=1;;
  n) nflag=1;;
  *) echo "illegal option $optletter"
     exit
esac
done
if (($pflag==0))
then echo "набор не найден"
else
  if (($cflag==0))
  then echo "набор не найден"
  exit
  else
    if (($nflag==0))
    then if (($cflag==0))
        then grep $sval $sval
        else grep -n $sval $sval
        fi
      else if (($cflag==0))
        then grep -i $sval $sval
        else grep -i -n $sval $sval
        fi
      fi
    else if (($cflag==0))
    then if (($cflag==0))
        then grep $sval $sval > $sval
        else grep -n $sval $sval > $sval
        fi
      else if (($cflag==0))
        then grep -i $sval $sval > $sval
        else grep -i -n $sval $sval > $sval
        fi
    fi
  fi
fi
```

```
[mazimov@fedora ~]$ cat 1.txt
cat dog
fox dog
cat fox
fox
cat
dog
[mazimov@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -p cat -C -n
[mazimov@fedora ~]$ cat 2.txt
1:cat dog
3:cat fox
5:cat
```

```
[mazimov@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -C -n
шаблон не найден
[mazimov@fedora ~]$ ./lab11_1.sh -o 2.txt -p cat -C -n
файл не найден
```

Вторая программа

Открыть ▾ ⊕

lab11_2.c

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    printf("Введите число\n");
    int a;
    scanf ("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```


Открыть ▾ ⊕

lab11_2.sh

```
#!/bin/bash
gcc lab11_2.c -o lab11_2
./lab11_2
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число больше 0";;
    2) echo "Число равно 0"
esac
```

```
[mazimov@fedora ~]$ ./lab11_2.sh
Введите число
0
Число равно 0
[mazimov@fedora ~]$ ./lab11_2.sh
Введите число
1
Число больше 0
[mazimov@fedora ~]$ ./lab11_2.sh
Введите число
-2
Число меньше 0
```

Третья программа

Открыть  lab11_3.sh

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

```
[mazimov@fedora ~]$ chmod +x lab11_3.sh
```

```
[mazimov@fedora ~]$ ./lab11_3.sh -c abc.txt 3
[mazimov@fedora ~]$ ls
lab11_1.sh  lab11_2.sh  Видео  Изображения  'Рабочий стол'
lab11_2    lab11_3.sh  Документы  Музыка  Шаблоны
lab11_2.c  work       Загрузки  Общедоступный
[mazimov@fedora ~]$ ./lab11_3.sh -r abc.txt 3
[mazimov@fedora ~]$ ls
lab11_1.sh  lab11_2.sh  Видео  Изображения  'Рабочий стол'
lab11_2    lab11_3.sh  Документы  Музыка  Шаблоны
lab11_2.c  work       Загрузки  Общедоступный
```

Четвёртая программа

Открыть ▾ + lab11_4.sh
~/

```
#!/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
```

```
[mazimov@fedora ~]$ touch lab11_4.sh
[mazimov@fedora ~]$ chmod +x lab11_4.sh
```

```
[mazimov@fedora ~]$ ./lab11_4.sh
.vboxclient-clipboard.pid
.vboxclient-seamless.pid
.vboxclient-draganddrop.pid
lab11_1.sh
lab11_2.c
lab11_2.sh
lab11_2
lab11_3.sh
lab11_4.sh
```

1. Команда `getopts` осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных. Синтаксис команды следующий: `getopts option-string variable [arg ...]` Флаги – это опции командной строки, обычно помеченные знаком минус; Например, для команды `ls` флагом может являться `-F`. Строка опций `option-string` – это список возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за символом, обозначающим этот флаг, должно следовать двоеточие. Соответствующей переменной присваивается буква данной опции. Если команда `getopts` может распознать аргумент, то она возвращает истину. Принято включать `getopts` в цикл `while` и анализировать введенные данные с помощью оператора `case`. Функция `getopts` включает две специальные переменные среды – `OPTARG` и `OPTIND`. Если ожидается дополнительное значение, то `OPTARG`

5. Следующие две команды ОС UNIX используются только совместно с управляющими конструкциями языка программирования `bash`: это команда `true`, которая всегда возвращает код завершения, равный нулю (т.е. истина), и команда `false`, которая всегда возвращает код завершения, не равный нулю (т.е. ложь). Примеры бесконечных циклов: `while true do echo hello andy done` `until false do echo hello mike done`

6. Строка `if test -f mans/i.s, mans/i.s` и является ли этот файл обычным файлом. Если данный файл является каталогом, то команда вернет нулевое значение (ложь).

7. Выполнение оператора цикла `while` сводится к тому, что сначала выполняется последовательность команд (операторов), которую задаёт список-команд в строке, содержащей служебное слово `while`, а затем, если последняя выпол-

Результаты

В ходе выполнения лабораторной работы были изучены основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.