

Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Лабораторная работа №11

Старцева А. С.

22 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Старцева Алина Сергеевна
- студент 1 курса, группа НММбд-03-22
- 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


```
#!/bin/bash
iflag=0 oflag=0 pflag=0 cflag=0 eflag=0
while getopts "bcdeop" opt; do
  case $opt in
    b) iflag=1; eval `grep $opt`;;
    c) oflag=1; eval `grep $opt`;;
    e) pflag=1; eval `grep $opt`;;
    o) cflag=1;;
    p) eflag=1;;
    *) echo "illegal option $opt" & exit 1
  esac
done
if ((iflag==0))
then echo "badno or noopt"
else
  if ((eflag==0))
  then echo "badn or noopt"
  else
    if ((cflag==0))
    then if ((pflag==0))
    then if ((oflag==0))
    then grep $opt $1 > /dev/null
    else grep -o $opt $1 > /dev/null
    fi
    else if ((oflag==0))
    then grep -i $opt $1 > /dev/null
    else grep -I -o $opt $1 > /dev/null
    fi
    fi
  else if ((cflag==0))
  then if ((oflag==0))
  then grep $opt $1 > /dev/null
  else grep -o $opt $1 > /dev/null
  fi
  else if ((oflag==0))
  then grep -i $opt $1 > /dev/null
  else grep -I -o $opt $1 > /dev/null
  fi
  fi
fi
```

```
lasterceva@asstarceva ~]$ cat 1.txt
one six
two one
free two
four one
five

lasterceva@asstarceva ~]$ ./lab1_1.sh -i 1.txt -o 2.txt -p one -C -n
lasterceva@asstarceva ~]$ cat 2.txt
1:one six
2:two one
4:four one
```

```
lasterceva@asstarceva ~]$ ./lab1_1.sh -i 1.txt -C -n
заблон не найден
lasterceva@asstarceva ~]$ ./lab1_1.sh -o 2.txt -p one -C -n
забл 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
```

```
[astarceva@asstarceva ~]$ ./lab11_2.sh
Введите число
1
Число больше 0
[astarceva@asstarceva ~]$ ./lab11_2.sh
Введите число
0
Число равно 0
[astarceva@asstarceva ~]$ ./lab11_2.sh
Введите число
-1
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
}
```

```
[astarceva@asstarceva ~]$ chmod +x lab11_3.sh
```

```
[astarceva@asstarceva ~]$ ./lab11_3.sh -c a#.txt 2
[astarceva@asstarceva ~]$ ls
backup      lab11_2     lab11_3.sh  документы  Музыка      Шаблоны
lab09.sh-   lab11_2.c  work       загрузки   Общедоступн
lab11_1.sh  lab11_3.sh  backup     изображения 'Рабочий ста
[astarceva@asstarceva ~]$ ./lab11_3.sh -r a#.txt 2
[astarceva@asstarceva ~]$ ls
backup      lab11_2     lab11_3.sh  документы  Музыка      Шаблоны
lab09.sh-   lab11_2.c  work       загрузки   Общедоступн
lab11_1.sh  lab11_3.sh  backup     изображения 'Рабочий ста
```

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
```

```
[astarceva@asstarceva work]$ ./lab11_4.sh
wrk/
wrk/10/
wrk/10/10_1.sh
wrk/10/10_2.sh
wrk/10/10_3.sh
wrk/10/10_4.sh
wrk/11/
lab11_4.sh
```

```
lab11_4.sh
[astarceva@asstarceva work]$ ls
lab11_4.sh  os  study  work.tar  wrk
[astarceva@asstarceva work]$
```

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` устанавливается в значение этого аргумента. Функция `getopts` также пони-

3. Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования `bash` предоставляет возможность использовать такие управляющие конструкции, как `for`, `case`, `if` и `while`. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути, являются операторами языка программирования `bash`. Поэтому при описании языка программирования `bash` термин оператор будет использоваться наравне с термином команда. Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда `test`, например, создана специально для использования в командных файлах. Единственная функция этой команды заключается в выработке кода завершения

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.