

Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Лабораторная работа №11

Паулу А. Ж.

Российский университет дружбы народов, Москва, Россия

Информация

- Паулу Антонью Жоау
- студент 1 курса, группа НММбд-02-22
- 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

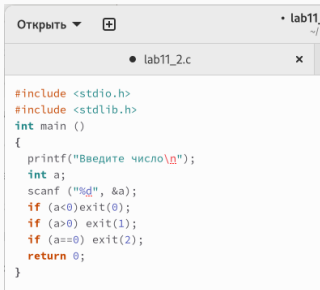
```
Открыть ▾  • lab11_1.sh  ⓘ  ⌵  ✕

#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
while getopts :i:opcn 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 $optletter"
       ;;
esac
done
if (($flag==0))
then echo "Вашим не найдем"
exit
else
    if (($flag==0))
    then echo "Ваш не найдем"
    exit
    else
        if (($flag==0))
        then if (($cflag==0))
            then if (($flag==0))
                then grep $sval $sval
                else grep -n $sval $sval
                fi
            else if (($flag==0))
                then grep -i $sval $sval
                else grep -i -n $sval $sval
                fi
            ..
        fi
    fi
fi
```

```
[azpaulu@fedora ~]$ cat 1.txt
1
2
3
4
5
6
7
[azpaulu@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -p 3 -C -n
[azpaulu@fedora ~]$ cat 2.txt
3:3
[azpaulu@fedora ~]$
```

```
1
[azpaulu@fedora ~]$ ./lab11_1.sh -i 1.txt -o -p 3 -C -n
Вашим не найдем
[azpaulu@fedora ~]$ ./lab11_1.sh -o 2.txt -p 3 -C -n
Файл не найден
[azpaulu@fedora ~]$
```

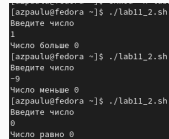
Вторая программа



```
#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;
}
```




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



```
[azrael@fedora ~]$ ./lab11_2.sh
Введите число
1
Число больше 0
[azrael@fedora ~]$ ./lab11_2.sh
Введите число
-9
Число меньше 0
[azrael@fedora ~]$ ./lab11_2.sh
Введите число
0
Число равно 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
}
```

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

```
[azpaul@fedora ~]$ ./lab11_3.sh -c 1.txt 5
[azpaul@fedora ~]$ ls
backup      lab10_3.sh-  lab11_2.c  Документы  'Рабочий стол'
lab09.sh    lab10_4.sh-  lab11_3.sh  Загрузки   Избранное
lab10_1.sh  lab10_4.sh-  lab11_3.sh  Избранное
lab10_2.sh  lab11_1.sh   work       Музыка
lab10_3.sh  lab11_2      home       Общедоступные
[azpaul@fedora ~]$ ./lab11_3.sh -r 1.txt 5
[azpaul@fedora ~]$ ls
backup      lab10_3.sh-  lab11_2.c  Документы  'Рабочий стол'
lab09.sh    lab10_4.sh-  lab11_3.sh  Загрузки   Избранное
lab10_1.sh  lab10_4.sh-  lab11_3.sh  Избранное
lab10_2.sh  lab11_1.sh   work       Музыка
lab10_3.sh  lab11_2      home       Общедоступные
```

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
```

```
azpaulu@fedora ~]$ ./lab11_4.sh
.cache/
.cache/mesa_shader_cache/
.cache/mesa_shader_cache/index
.cache/mesa_shader_cache/45/
.cache/mesa_shader_cache/45/087bb9d067b72e4ae4410872807a4333b3d011
.cache/mesa_shader_cache/a5/
.cache/mesa_shader_cache/a5/0e6f48985dfbdc0389e2f2e183b3a846b6aaba
.cache/mesa_shader_cache/a5/7e22bca5650285dd9e4527949759cc483a2c8e
.cache/mesa_shader_cache/a5/f11f0e003779cf3a7cfff9f3e20dda9dcbf82ed
.cache/mesa_shader_cache/58/
.cache/mesa_shader_cache/58/016bb235fd37eeb3d521b9985fff9edcc72611
.cache/mesa_shader_cache/58/7de943071c254478ddcdef5a3de32fc4c549db
.cache/mesa_shader_cache/ff/
.cache/mesa_shader_cache/ff/5ddb2f79d46c5b4eba6c6afb2b3bdd48f69748
.cache/mesa_shader_cache/8f/
.cache/mesa_shader_cache/8f/61378b62853479f9c39e43668d434fb5f6b74c
.cache/mesa_shader_cache/b6/
.cache/mesa_shader_cache/b6/372c0597390ad489841891b52cdacd39cdd99
.cache/mesa_shader_cache/b6/453b42ee3ef41a713530085660f214997ac9f8
```

```
azpaulu@fedora ~]$ ls
azpaulu.tar  lab10_3.sh  lab11_2  work  Музыка
Backup      lab10_3.sh  lab11_2.c  Видео  Объекты
lab09.sh    lab10_4.sh  lab11_3.sh  Документы  'Рабочий стол'
lab10_1.sh  lab10_4.sh  lab11_3.sh  Зеркала    Шаблоны
lab10_2.sh  lab11_1.sh  lab11_4.sh  Избранная
```

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 –

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

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.