

Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Лабораторная работа №11

Нати Ф. Б.

Российский университет дружбы народов, Москва, Россия

Информация

- Нати Франсиску Бунда
- 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 1:o:p:n 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 ((iflag==0))
then echo "Файл не найден"
else
    if ((iflag==0))
    then echo "Файл не найден"
    exit
    else
        if ((oflag==0))
        then if ((cflag==0))
            then if ((nflag==0))
```


```
[fbnati@fedora ~]$ cat 1.txt
hope
love
evil
shit
[fbnati@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -p hope -C -n
[fbnati@fedora ~]$ cat 2.txt
l:hope
[fbnati@fedora ~]$
```

```
[fbnati@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -p nope -C -n
[fbnati@fedora ~]$ ./lab11_1.sh -i 1.txt -o 2.txt -C -n
Файл не найден
[fbnati@fedora ~]$ ./lab11_1.sh -o 2.txt -p hope -C -n
Файл не найден
[fbnati@fedora ~]$
```

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
```

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

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
}
```

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

```
[fbnati@fedora ~]$ ./lab11_3.sh -c 1.txt 3
[fbnati@fedora ~]$ ls
backup      lab10_3.sh  lab11_2.c  Видео      Музыка
lab09.sh    lab10_4.sh  lab11_2.sh  Документы  Общедоступные
lab10_1.sh  lab11_1.sh  lab11_3.sh  Загрузки   'Рабочий стол'
lab10_2.sh  lab11_2     work        Изображения  Шаблоны
[fbnati@fedora ~]$ ./lab11_3.sh -r 1.txt 3
[fbnati@fedora ~]$ ls
backup      lab10_3.sh  lab11_2.c  Видео      Музыка
lab09.sh    lab10_4.sh  lab11_2.sh  Документы  Общедоступные
lab10_1.sh  lab11_1.sh  lab11_3.sh  Загрузки   'Рабочий стол'
lab10_2.sh  lab11_2     work        Изображения  Шаблоны
```

Четвёртая программа

```
Открыть ▾ lab11_4.sh
#1/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
```

```
[fbnati@fedora ~]$ ./lab11_4.sh
.bash_history
work/
work/study/
work/study/2022-2023/
work/study/2022-2023/Архитектура компьютера/
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/branch
/
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/hooks
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/hooks
plypatch-msg.sample
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/hooks
mail-msg.sample
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/hooks
monitor-watchman.sample
work/study/2022-2023/Архитектура компьютера/study_2022-2023_arh-pc/.git/hooks
st-update.sample
```

```
[fbnati@fedora ~]$ ls
backup      lab10_4.sh  lab11_2.sh  Документы  'Рабочий стол'
fbnati.tar  lab10_4.sh  lab11_3.sh  Загрузки   папки
lab09.sh    lab11_1.sh  lab11_4.sh  Избранная
lab10_1.sh  lab11_2     work        Музыка
lab10_2.sh  lab11_2.c  video       Общедоступные
```

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

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 опера-

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.