

Лабораторная работа №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Валиева М. Р.

Российский университет дружбы народов, Москва, Россия

Информация

- Валиева Марина Русланбековна
- НБИбд-01-22
- Российский университет дружбы народов

Вводная часть

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

Создание презентации

Элементы презентации

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

1. Используя команды `getopts` `grep` напишем командный файл, который анализирует командную строку с ключами и выполним его: `-i inputfile` — прочитать данные из указанного файла; `-o outputfile` — вывести данные в указанный файл; `-p` шаблон — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки

```
#!/bin/bash
cflag=0;
nflag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nflag=1;;
esac
done
if [ $cflag -a $nflag ]
then
grep -n $pval $ival>$oval
```

```
mrvalieva@dk8n74 ~ $ touch lab11_1.sh
mrvalieva@dk8n74 ~ $ chmod +x lab11_1.sh
mrvalieva@dk8n74 ~ $ ./lab11_1.sh
mrvalieva@dk8n74 ~ $ ./lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n
mrvalieva@dk8n74 ~ $ cat fout.txt
cat: fout.txt: Нет такого файла или каталога
mrvalieva@dk8n74 ~ $ ./lab11_1.sh
mrvalieva@dk8n74 ~ $ ./lab11_1.sh
./lab11_1.sh: строка 16: $oval: неоднозначное перенаправление
mrvalieva@dk8n74 ~ $ ./lab11_1.sh -i text.txt -o fout.txt -p файлы -C -n
grep: text.txt: Нет такого файла или каталога
mrvalieva@dk8n74 ~ $ cat fout.txt
```

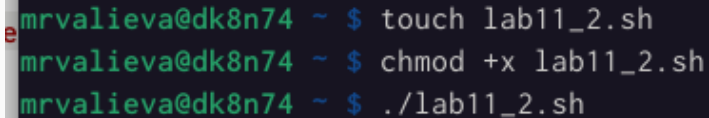
Рис. 1: Задание 1

```
File Edit Options Buffers Tools Sh-Script Outlin
#!/bin/bash
cflag=0;
nglag=0;
while getopts i:o:p:C:n opt
do
case $opt in
i) ival=$OPTARG;;
o) oval=$OPTARG;;
p) pval=$OPTARG;;
C) cflag=1;;
n) nglag=1;;
esac
done
if [ $cflag -a $nglag ]
then
grep -n $pval $ival>$oval
elif test $cflag
then
- 349 lab11_1.sh Shell-script
```

Рис. 2: Задание 1

2. Напишем сначала на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем завершим программу при помощи функции `exit(n)`, передавая информацию о коде завершения в оболочку. Командный файл вызовет эту программу и, проанализировав с помощью команды `$?`, выдаст сообщение о том, какое число было введено

```
#!/bin/bash
gcc -c script2.c
gcc -o script2 script2.c
./script2
case $? in
    1) echo отрицательное;;
    2) echo равно нулю;;
    3) echo положительное;;
esac
```



```
mrvalieva@dk8n74 ~ $ touch lab11_2.sh
mrvalieva@dk8n74 ~ $ chmod +x lab11_2.sh
mrvalieva@dk8n74 ~ $ ./lab11_2.sh
```

Рис. 3: Задание 2

```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac
```

Рис. 4: Задание 2

3. Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N

```
#!/bin/bash
let i=$1+1
while (( i-=1 ))
do touch $i.tmp
done
let j=$2+1;
while (( j-=1 ))
do rm $j.tmp
done
```

```
1 #!/bin/bash
2 let i=$1+1
3 while (( i -=1 ))
4 do touch $i.tmp
5 done
6 let j=$2+1;
7 while (( j -=1 ))
8 do rm $j.tmp
9 done
```

Рис. 5: Задание 3

```

mrvalieva@dk8n74 ~ $ touch lab11_3.sh
mrvalieva@dk8n74 ~ $ chmod +x lab11_3.sh
mrvalieva@dk8n74 ~ $ touch lab11_3.sh
mrvalieva@dk8n74 ~ $ touch lab11_3.sh
mrvalieva@dk8n74 ~ $ ./lab11_3.sh 4
mrvalieva@dk8n74 ~ $ ls
14042.py  bin          lab09.sh~   lab11_2.sh  script.sh   Музыка
14043.py  Desktop     lab10_1.sh  lab11_3.sh  tmp         Общедоступные
1404.py   fout.txt    lab10_2.sh  my_os       work        'Рабочий стол'
1.py      GNUstep     lab10_3.sh  public      Видео       Шаблоны
2.py      hello.asm   lab10_4.sh  public_html Документы
backup    lab09.sh    lab11_1.sh  __pycache__ Изображения

```

Рис. 6: Задание 3

4. Напишем командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад.

```
#!/bin/bash
```

```
(find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
```

```
1 #!/bin/bash
2 (find $1 -mtime -7 -daystart) | xargs tar -cf arhiv.tar
3
```

Рис. 7: Задание 4

```

mrvalieva@dk8n74 ~ $ touch lab11_4.sh
mrvalieva@dk8n74 ~ $ chmod +x lab11_4.sh
mrvalieva@dk8n74 ~ $ ./lab11_4.sh
mrvalieva@dk8n74 ~ $ ./lab11_4.sh

^C
mrvalieva@dk8n74 ~ $ ls
14042.py  bin          lab09.sh~   lab11_2.sh  __pycache__  Изображения
14043.py  Desktop     lab10_1.sh  lab11_3.sh  script.sh     Музыка
1404.py   fout.txt    lab10_2.sh  lab11_4.sh  tmp          Общедоступные
1.py      GNUstep     lab10_3.sh  my_os       work          'Рабочий стол'
2.py      hello.asm   lab10_4.sh  public      Видео         Шаблоны
backup    lab09.sh    lab11_1.sh  public_html  Документы

```

Рис. 8: Задание 4

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

Спасибо за внимание!