

Лабораторная работа №11

Операционные системы

Бембо жозе лумингу

21 марта 2024

Российский университет дружбы народов, Москва, Россия

- Бембо лумингу жозе
- студент из группы НКАбд-01-23
- Факультет физико-математических и естественных наук
- Российский университет дружбы народов
- 1032229273@rudn.ru
- <https://github.com/zlbembo1>

Цель данной лабораторной работы - изучить основы программирования в оболочке ОС UNIX, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-iinputfile` — прочитать данные из указанного файла;
 - `-ooutputfile` — вывести данные в указанный файл;
 - `-ршаблон` — указать шаблон для поиска;
 - `-C` — различать большие и малые буквы;
 - `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.

```
#!/bin/bash

while getopts i:o:p:cn 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 ! test $cflag
```

```
then
```

```
    cf=-i
```

```
fi
```

```
if test $nflag
```

```
then
```

```
    nf=-n
```

```
fi
```

```
grep $cf $nf $pval $ival >> $oval
```

2. Программа на языке Си, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

```
#include <stdlib.h>
#include <stdio.h>

int main () {
    int n;
    printf ("Введите число: ");
    scanf ("%d", &n);
    if(n>0){
        exit(1);}
    else if (n==0) {
        exit(0);}
    else {
        exit(2);}
```



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

```
gcc -o cprog 12.c
```

```
./cprog
```

```
case $? in
```

```
0) echo "Число равно нулю";;
```

```
1) echo "Число больше нуля";;
```

```
2) echo "Число меньше нуля";;
```

```
esac
```

3. Командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `%d` (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

```
#!/bin/bash
for((i=1; i<=$*; i++))
do
if test -f "$i".tmp
then rm "$i".tmp
else touch "$i.tmp"
fi
done
```

4. Командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`).

```
#!/bin/bash  
find $* -mtime -7 -mtime +0 -type f > FILES.txt  
tar -cf archive.tar -T FILES.txt
```

При выполнении данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.