

Российский университет дружбы народов

Факультет физико-математических и естественных наук

Лабораторная работа № 11. Программирование в командном процессоре ОС UNIX. Ветвления и циклы

- Имя : исса гадир
- Студенческий билет : 1032218267
- Группа : нфибд-01-21

Цель работы

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

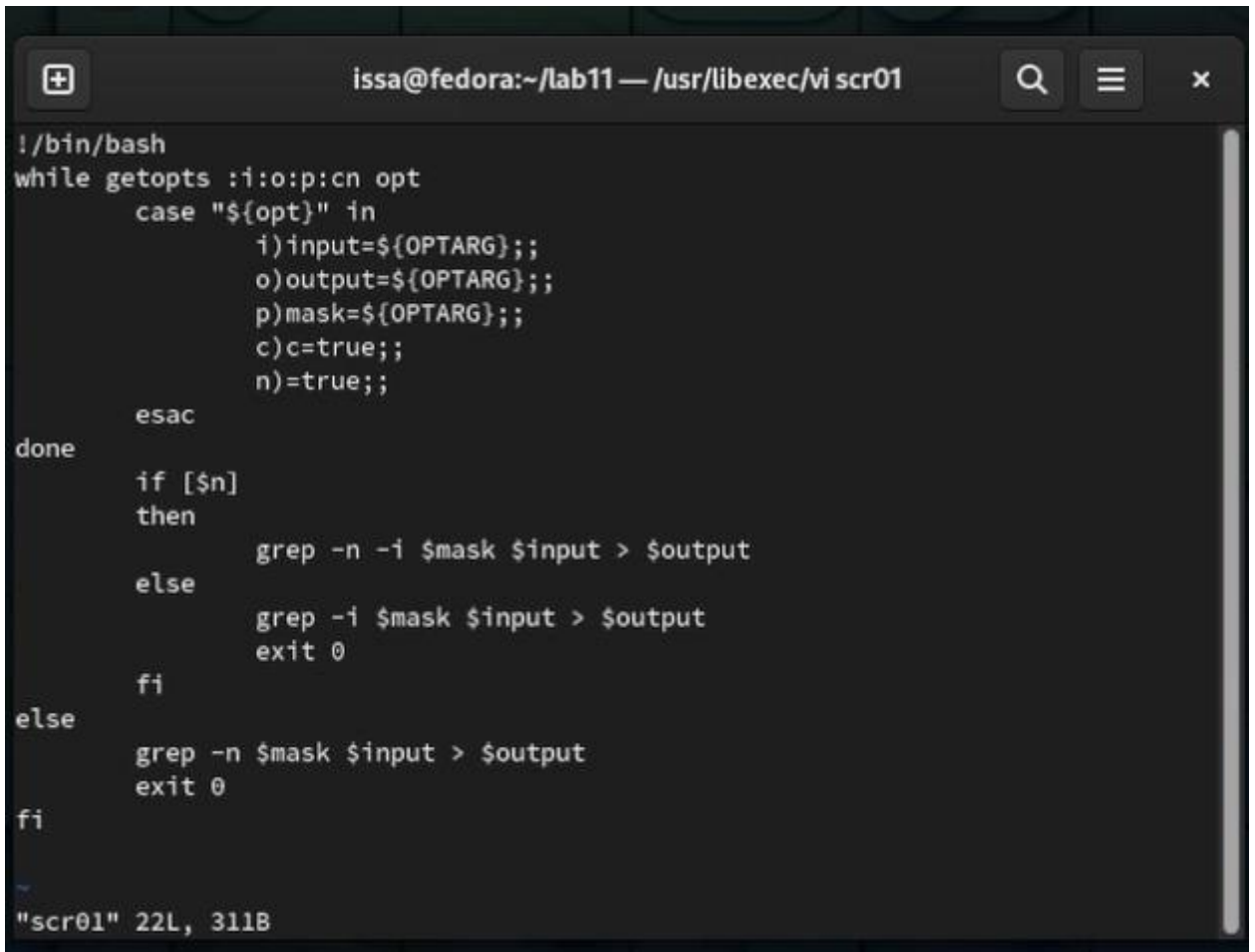
Последовательность выполнения работы :

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

```
issa@fedora:~/lab11

[issa@fedora ~]$ ls
abcl      etc      lab07.sh  monthly  Public    tutorial
australia feathers lab07.sh~ Music     reports   Videos
conf.txt  file.txt letters   my_os     scr        work
Desktop   gitflow  may       PhoneInfo ski.places
Documents gitflow-installer.sh memos     Pictures  Templates
Downloads include  wisk      play      text1.txt

[issa@fedora ~]$ mkdir lab11
[issa@fedora ~]$ cd lab11/
[issa@fedora lab11]$ touch scr01
[issa@fedora lab11]$
```



```
! /bin/bash
while getopts :i:o:p:cn opt
do
    case "${opt}" in
        i) input=${OPTARG};;
        o) output=${OPTARG};;
        p) mask=${OPTARG};;
        c) c=true;;
        n) n=true;;
    esac
done
if [ $n ]
then
    grep -n -i $mask $input > $output
else
    grep -i $mask $input > $output
    exit 0
fi
else
    grep -n $mask $input > $output
    exit 0
fi

"scr01" 22L, 311B
```

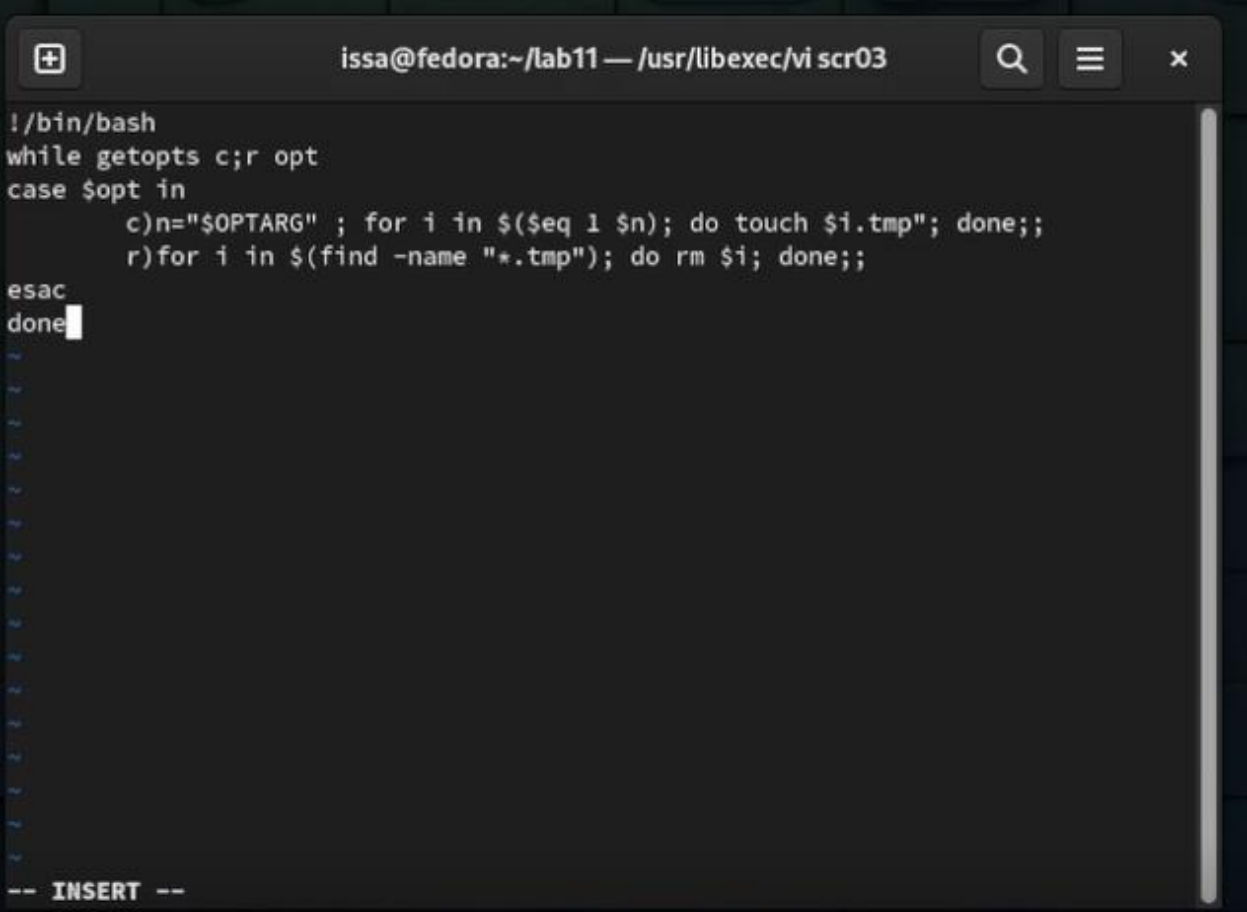
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Команд- ный файл должен вызывать эту программу и, проанализировав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

```
issa@fedora:~/lab11 — /usr/libexec/vi compare.cpp
#include <iostream>
using namespace std;
int main(int argc, char *argv[]) {
    if (atoi(argv[1])) {
        exit(1);
    }
    else if (atoi(argv[1] == 0 )) {
        exit(2);
    }
    else {
        esxit(3);
    }
}
return 0;

-- VISUAL BLOCK --
```

```
issa@fedora:~/lab11 — /usr/libexec/vi
#!/bin/bash
cc=g++
EXEC=compare
src=compare.cpp
if [ "$SRC" -nt "EXEC" ]
then
    echo "REBUILDING $ EXEC ..."
    $CC -o $EXEC $SRC
fi
./EXEC $1
ec=$?
if [ "$ec" == "1" ]
then
    echo "input > 0"
fi
if [ "$ec" == "2" ]
then
    echo "input == 0"
fi
if [ "$ec" == "3" ]
then
    echo "input < 0"
fi
E32: No file name
```

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (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).



```
issa@fedora:~/lab11 — /usr/libexec/vi scr03
#!/bin/bash
while getopts c;r opt
case $opt in
    c)n="$OPTARG" ; for i in $(seq 1 $n); do touch $i.tmp; done;;
    r)for i in $(find -name "*.tmp"); do rm $i; done;;
esac
done
```

-- INSERT --

4. Написать 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).

4. Какие операторы используются для прерывания цикла ?

Break

5. Для чего нужны команды false и true ?

строковый : True для истины , False для лжи ; числовой (встроенные типы int и float) : 1 для истины , 0 для лжи .

6. Что означает строка `if test - f man \s / Si .\ s ,` ,
встреченная в командном файле ?

Проверьте , существует ли также руководство

7. Объясните различия между конструкциями while и until?

While : Выполнение начинается при выполнении условий Until : Остановить выполнение , если условия выполнены

Вывод:

*Оболочки Я изучил основы программирования операционной системы UNIX .
Узнал , как использовать логические структуры управления и ЦИКЛЫ ДЛЯ
написания более сложных командных файлов .*