

Презентация №11

Программирование в командном процессоре ОС UNIX. Ветвление и циклы

Гузева Ирина Николаевна

22 марта 2023

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

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

Выполнение лабораторной работы

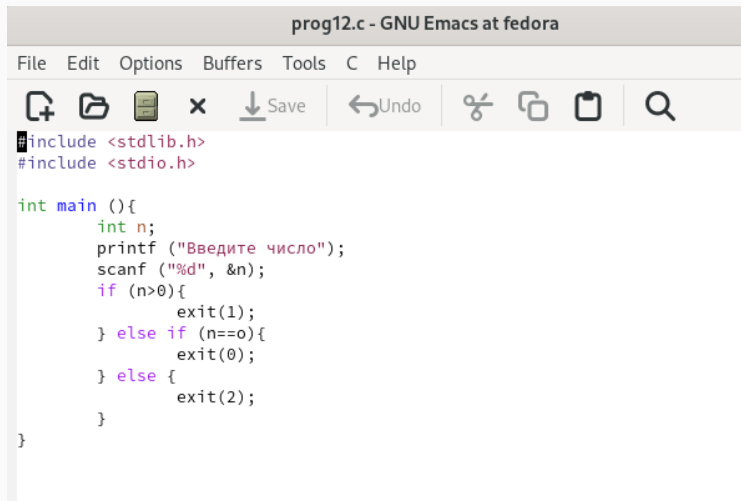
1. Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами: `-i` inputfile — прочитать данные из указанного файла; `-o` outputfile — вывести данные в указанный файл; `-p` шаблон — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n`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

```
prog11.sh      x      input.txt

#!/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
```

2. Можно увидеть, что программа сработала правильно



```
prog12.c - GNU Emacs at fedora
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Search]

#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);
    }
}
```

3. Написала 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. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

```
Открыть ▾ + prog13.sh ~/
#!/bin/bash
gcc -o cprog prog12.c
./cprog
case $? in
0) echo "Число равно 0";;
2) echo "Число меньше 0";;
1) echo "Число больше 0";;
esac
```

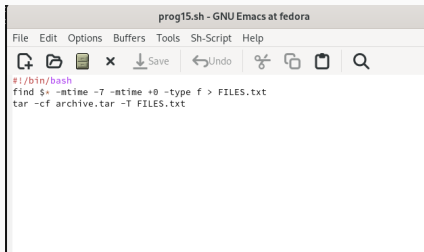
4. Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до n (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если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



```
Открыть ▾ + • prog14.sh ~/  
#!/bin/bash  
for ((i=1; i<=$*; i++))  
do  
if test -f "$i".tmp  
then rm "$i".tmp  
else touch "$i.tmp"  
fi  
done
```

Рис. 4: Программа №3

5. Написала 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)



```
prog15.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Cut, Copy, Paste, Find]
#!/bin/bash
find $* -mtime -7 -mtime +0 -type f > FILES.txt
tar -cf archive.tar -T FILES.txt
```

Рис. 5: Программа №4

В процессе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.