

Лабораторная работа № 13

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Разанацуа Сара Естэлл

Цель работы

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

- Создаю файл с разрешением на исполнение, а затем ищет в указанном файле нужные строки, определяемые ключом -p

```
serazanacua@serazanacua:~$ chmod +x 111.sh
serazanacua@serazanacua:~$ bash 111.sh -p улит -i input.txt -o output.txt -c -n
grep: input.txt: Нет такого файла или каталога
serazanacua@serazanacua:~$ touch input.txt
serazanacua@serazanacua:~$ touch output.txt
serazanacua@serazanacua:~$ bash 111.sh -p улит -i input.txt -o output.txt -c -n
serazanacua@serazanacua:~$
```

Рис. 1: Создание файла



```
#!/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;;
```

- Результат работы программы в файле output.txt.

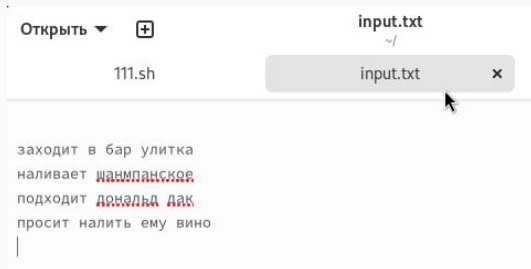
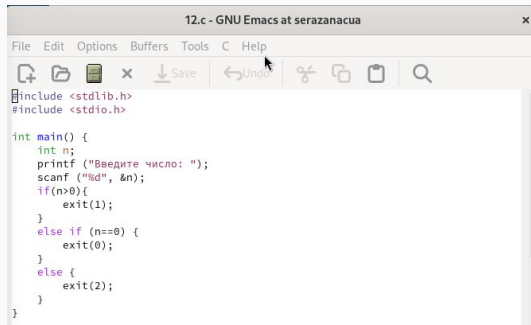


Рис. 3: Результат работы программы



- Создаю исполняемый файл для второй программы, также создаю файл 12.c для программы на Си. Пишу программу 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



```
12.c - GNU Emacs at serazanacua
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Cut, Copy, Paste, Find]
#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);
    }
}
```

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



```
#!/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
```

Рис. 6: Код программы

- Создаю исполняемый файл для третьей программы. 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)



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

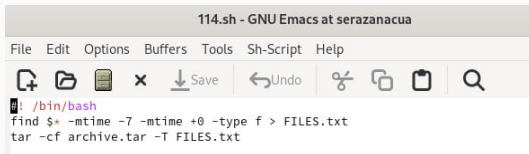
Рис. 7: Код программы

- Проверяю, что программа создала файлы и удалила их при соответствующих запросах.

```
serazanacua@serazanacua:~$ bash 113.sh 4
serazanacua@serazanacua:~$ ls
111.sh      Desktop      monthly      text.txt
112.sh      dotfiles     newdir       work
113.sh      Downloads    output.txt   Видео
12.c        file.txt     pandoc-3.1.12.2 Доклады
1.tmp       git-extended pandoc-crossref Документы
2.tmp       git@github.com:Sarahestelle pandoc-crossref.1 Загрузки
3.tmp       https:       prog1.sh     Изображения
4.tmp       input.txt   prog2.sh     Музыка
abc1        'lab07.sh#' prog3.sh     Общедоступные
backup      lab07.sh    prog4.sh     'Рабочий стол'
bin         lab07.sh~   reports      Шаблоны
conf.txt    LICENSE     ski.places
cprog       may         ski.places
serazanacua@serazanacua:~$
```

Рис. 8: Результат работы программы

- Создаю исполняемый файл для четвертой программы. Это 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`).



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

Рис. 9: Код программы

- Проверяю работу программы.

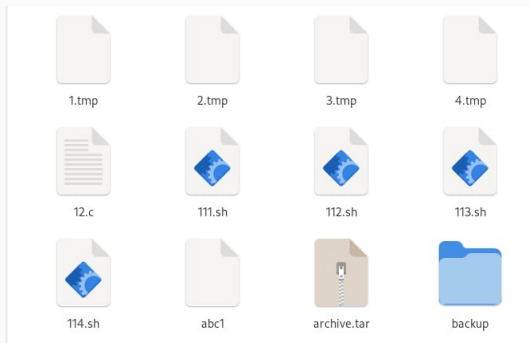


Рис. 10: Результат работы программы

Выводы

- 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.

1. Командная строка Windows [Электронный ресурс]. URL:
<https://foxford.ru/wiki/informatika/komandnaya-stroka-windows>.

Спасибо за внимания