

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Плугин Никита¹

18 апреля, 2024, Москва, Россия

¹Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

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

Задачи лабораторной работы

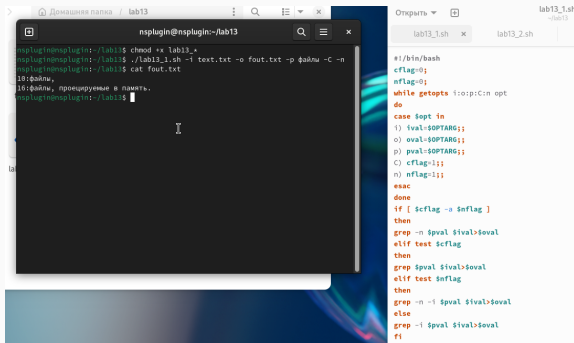
1 Выполнить 4 задания

Процесс выполнения лабораторной работы

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; `-r шаблон` — указать шаблон для поиска; `-C` — различать большие и малые буквы; `-n` — выдавать номера строк;

а затем ищет в указанном файле нужные строки

Выполнение работы



The image shows a terminal window on the left and a code editor on the right. The terminal window, titled 'nsplugin@nsplugin:~/lab13', shows the execution of a shell script 'lab13.sh'. The script creates a file 'fout.txt' and prints its contents. The code editor, titled 'lab13_1.sh', shows the source code of the script.

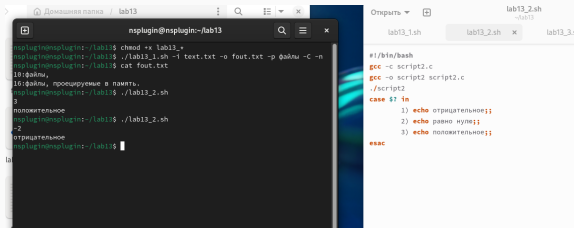
```
nsplugin@nsplugin:~/lab13$ chmod +x lab13.*
nsplugin@nsplugin:~/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файл -C -n
nsplugin@nsplugin:~/lab13$ cat fout.txt
10:файл,
10:файл, процириуемые в память,
nsplugin@nsplugin:~/lab13$
```

```
#!/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
elif test $cflag
then
grep $pval $ival>$oval
elif test $nflag
then
grep -n -i $pval $ival>$oval
else
grep -i $pval $ival>$oval
fi
```

Рис. 1: 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

Выполнение работы



The image shows a terminal window on the left and a file editor on the right. The terminal window, titled 'nspugin@nspugin:~/lab13', shows the execution of a C program 'cheod' with arguments '+x lab13_+'. The program outputs '10:файлы, просцируение в память.' and then runs './lab13_2.sh'. The script outputs 'положительное' and '-2', and then 'отрицательное'. The file editor, titled 'lab13_2.sh', shows the script's source code, which includes a case statement for checking if a number is negative, zero, or positive.

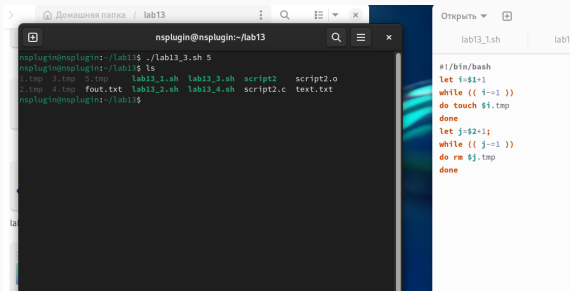
```
nspugin@nspugin:~/lab13$ cheod +x lab13_+
nspugin@nspugin:~/lab13$ ./lab13_1.sh -1 text.txt -o fout.txt -p файлы -C -n
nspugin@nspugin:~/lab13$ cat fout.txt
10:файлы,
10:файлы, просцируение в память.
nspugin@nspugin:~/lab13$ ./lab13_2.sh
положительное
nspugin@nspugin:~/lab13$ ./lab13_2.sh
-2
отрицательное
nspugin@nspugin:~/lab13$
```

```
#!/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
```

Рис. 2: Задание 2

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

Выполнение работы



The image shows a terminal window and a file explorer. The terminal window, titled 'nsplugin@nsplugin:~/lab13', displays the following commands and output:

```
nsplugin@nsplugin:~/lab13$ ./lab13_3.sh 5
nsplugin@nsplugin:~/lab13$ ls
1.tmp  3.tmp  5.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
2.tmp  4.tmp  fout.txt lab13_2.sh  lab13_4.sh  script2.c  text.txt
nsplugin@nsplugin:~/lab13$
```

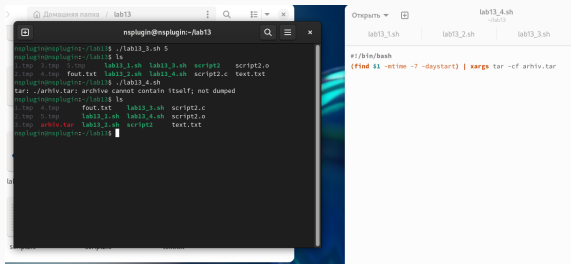
The file explorer, titled 'Домашняя папка / lab13', shows two files: 'lab13_1.sh' and 'lab13'. The 'lab13_1.sh' file is selected, and its contents are displayed in a preview pane:

```
#!/bin/bash
let i=$1+1
while (( i-->0 ))
do touch $i.tmp
done
let j=$2+1
while (( j-->0 ))
do rm $j.tmp
done
```

Рис. 3: 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.

Выполнение работы



The image shows a terminal window and a file manager. The terminal window, titled 'nsplugin@nsplugin:~/lab13', displays the execution of a script 'lab13_3.sh' with 5 arguments. The script lists files in the current directory, which include 'fout.txt', 'lab13_1.sh', 'lab13_2.sh', 'lab13_3.sh', 'script2.c', 'script2.o', and 'text.txt'. It then attempts to create a tar archive named 'arhiv.tar' but fails with the message 'tar: ./arhiv.tar: archive cannot contain itself; not dumped'. The file manager, titled 'lab13_4.sh', shows a directory listing with 'lab13_1.sh', 'lab13_2.sh', and 'lab13_3.sh'. The terminal output is as follows:

```
nsplugin@nsplugin:~/lab13$ ./lab13_3.sh 5
nsplugin@nsplugin:~/lab13$ ls
1.tmp  3.tmp  5.tmp  lab13_1.sh  lab13_3.sh  script2  script2.o
2.tmp  4.tmp  fout.txt  lab13_2.sh  lab13_4.sh  script2.c  text.txt
nsplugin@nsplugin:~/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
nsplugin@nsplugin:~/lab13$ ls
1.tmp  4.tmp  fout.txt  lab13_1.sh  lab13_3.sh  script2.c
2.tmp  5.tmp  lab13_2.sh  lab13_4.sh  script2.o
3.tmp  arhiv.tar  lab13_1.sh  script2  text.txt
nsplugin@nsplugin:~/lab13$
```

The file manager shows the following content:

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

Рис. 4: Задание 4

Выводы по проделанной работе

В данной работе мы изучили основы программирования в оболочке ОС UNIX и писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.