

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Касканте Родригес Альберто¹

25 декабря, 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

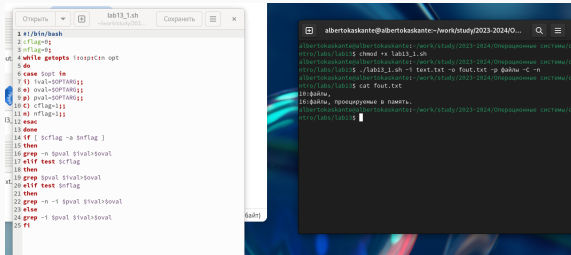
1 Выполнить 4 задания

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 consists of two side-by-side screenshots. The left screenshot shows a text editor window titled 'lab13_1.sh' with a Bash script. The script defines variables for flags, sets a loop, and uses a case statement to handle different flags. The right screenshot shows a terminal window where the script is being executed. The user runs 'chmod +x lab13_1.sh' and then './lab13_1.sh -f text.txt -o fout.txt -p qinw -C -n'. The terminal output shows the script's execution, including file creation and flag processing.

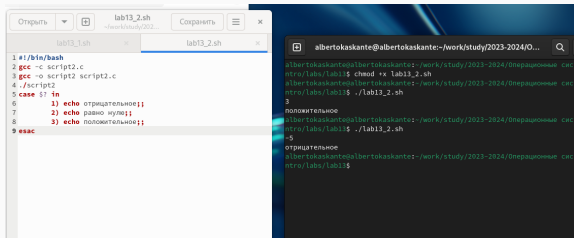
```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts :fospn opt
5 do
6 case $opt in
7 f) sval=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 *)
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $sval $oval>$oval
17 elif test $cflag
18 then
19 grep $pval $oval>$oval
20 elif test $nflag
21 then
22 grep -n -f $pval $oval>$oval
23 else
24 grep -f $pval $oval>$oval
25 fi
```

```
albertokskante@albertokskante: /work/study/2023-2024/Операционные системы$
nsro/labs/lab13$ chmod +x lab13_1.sh
albertokskante@albertokskante: /work/study/2023-2024/Операционные системы$
nsro/labs/lab13$ ./lab13_1.sh -f text.txt -o fout.txt -p qinw -C -n
albertokskante@albertokskante: /work/study/2023-2024/Операционные системы$
nsro/labs/lab13$ cat fout.txt
10:qinw,
10:qinw, прокурорские в поезде.
albertokskante@albertokskante: /work/study/2023-2024/Операционные системы$
nsro/labs/lab13$
```

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 code editor window on the left and a terminal window on the right. The code editor displays a shell script named `script2.c` with the following content:

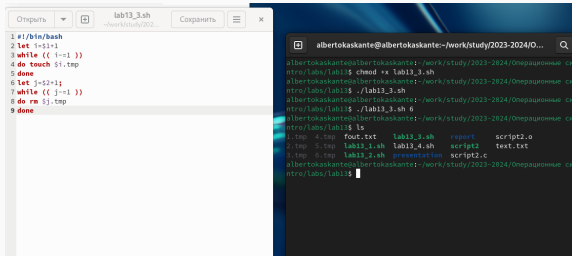
```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $1 in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 esac
```

The terminal window shows the execution of the script. The user runs `chmod +x lab13_2.sh` and then `./lab13_2.sh`. The script outputs "отрицательное" (negative) for the first case, "равно нулю" (equal to zero) for the second case, and "положительное" (positive) for the third case.

Рис. 2: Задание 2

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

Выполнение работы



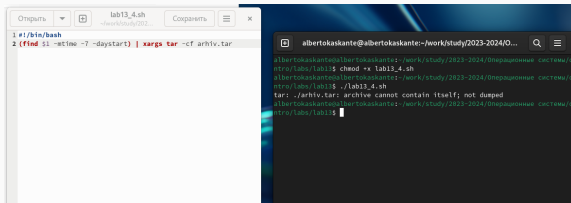
```
1 #!/bin/bash
2 let i=i+1
3 while (( i<=1 ))
4 do touch $i.tmp
5 done
6 let j=2+1;
7 while (( j=1 ))
8 do rm $j.tmp
9 done
```

```
albertokaskante@albertokaskante:~/work/study/2023-2024/O...
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные см
ntro/labs/lab13$ chmod +x lab13_3.sh
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные см
ntro/labs/lab13$ ./lab13_3.sh
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные см
ntro/labs/lab13$ ./lab13_3.sh 6
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные см
ntro/labs/lab13$ ls
1.tmp 4.tmp  fout.txt      lab13_3.sh  report    script2.o
2.tmp 5.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt
3.tmp 6.tmp  lab13_2.sh  presentation script2.c
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные см
ntro/labs/lab13$
```

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 displays two terminal windows side-by-side. The left window, titled 'lab13_4.sh', shows a sequence of commands: a shell prompt, a 'find' command to locate files older than 7 days, and an 'xargs tar' command to create an archive. The right window, titled 'albertokaskante@albertokaskante:~/work/study/2023-2024/O...', shows the execution of 'chmod +x lab13_4.sh', the execution of the script, and the resulting error message from 'tar'.

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

albertokaskante@albertokaskante:~/work/study/2023-2024/O...
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные системы/О
ntro/labs/lab13$ chmod +x lab13_4.sh
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные систе
ntro/labs/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
albertokaskante@albertokaskante:~/work/study/2023-2024/Операционные систе
ntro/labs/lab13$
```

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.