

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Агоссоу Вигнон Тримегистре Разиел НФИбд-05-22¹

7 сентября, 2023, 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

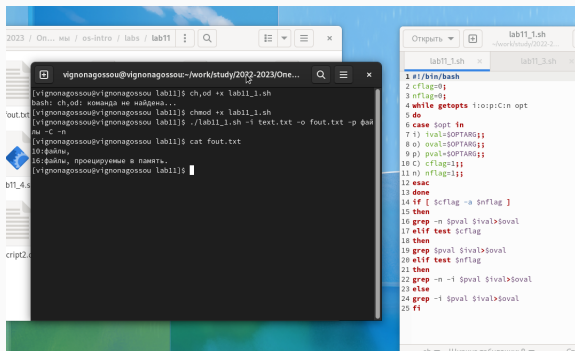
1 Выполнить 4 задания

Процесс выполнения лабораторной работы

1. Используя команды `getopts` `grep` напомним 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 and a code editor side-by-side. The terminal window, titled 'vignonagossou@vignonagossou:~/work/study/2022-2023/One...', displays the execution of a shell script named 'lab11_1.sh'. The user runs 'ch,od +x lab11_1.sh', then 'bash: ch,od: команда не найдена...', followed by 'chmod +x lab11_1.sh'. Then they run './lab11_1.sh -i text.txt -o fout.txt -p 0' and finally 'cat fout.txt', which outputs '10:файл, прощуренные в память.' and '16:файл, прощуренные в память.'.

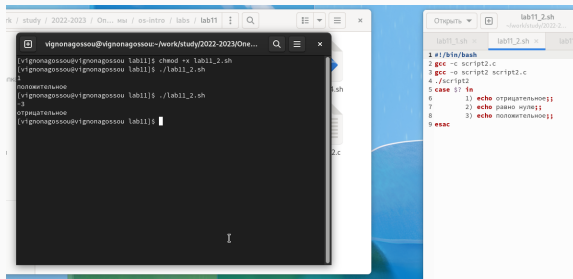
The code editor, titled 'lab11_1.sh', shows the following script:

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:o:p:C:n opt
5 do
6 case $opt in
7 i) tval=$OPTARG;;
8 o) oval=$OPTARG;;
9 p) pval=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $tval>$oval
17 elif test $cflag
18 then
19 grep $pval $tval>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $tval>$oval
23 else
24 grep -i $pval $tval>$oval
25 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 and a code editor. The terminal window, titled 'vignonagossou@vignonagossou:~/work/study/2022-2023/One...', displays the following commands and output:

```
[vignonagossou@vignonagossou lab11]$ chmod +x lab11_2.sh
[vignonagossou@vignonagossou lab11]$ ./lab11_2.sh
положительное
1
отрицательное
[vignonagossou@vignonagossou lab11]$
```

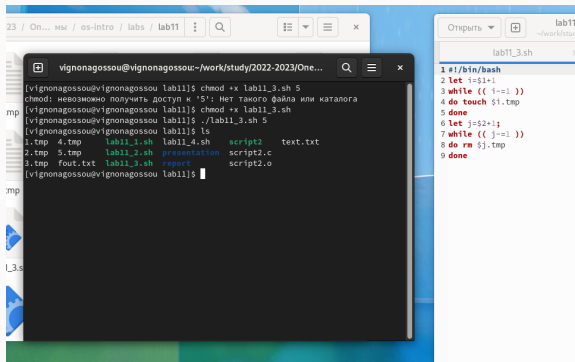
The code editor, titled 'lab11_2.sh', shows the script content:

```
1 #!/bin/bash
2 gcc -c script2.c
3 gcc -o script2 script2.c
4 ./script2
5 case $? in
6     1) echo отрицательное;;
7     2) echo равно нулю;;
8     3) echo положительное;;
9 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 editor side-by-side. The terminal window, titled 'vignonagossou@vignonagossou: ~/work/study/2022-2023/One...', displays the execution of a shell script 'lab11_3.sh' with permissions '5'. It shows the script's output, which lists files in a directory and then executes a loop that touches a file and increments a counter. The file editor, titled 'lab11_3.sh', shows the source code of the script, which includes a shebang, variable initialization, a while loop, and a touch command.

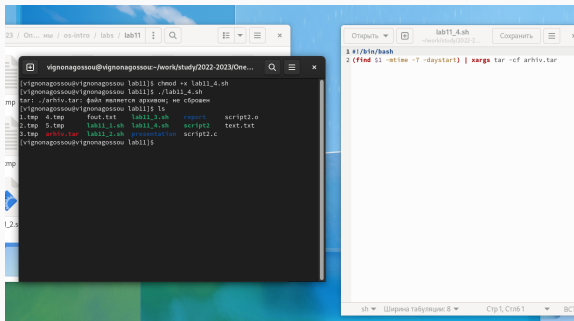
```
[vignonagossou@vignonagossou lab11]$ chmod +x lab11_3.sh 5
chmod: невозможно получить доступ к '5': Нет такого файла или каталога
[vignonagossou@vignonagossou lab11]$ chmod +x lab11_3.sh
[vignonagossou@vignonagossou lab11]$ ./lab11_3.sh 5
[vignonagossou@vignonagossou lab11]$ ls
1.tmp 4.tmp lab11_1.sh lab11_4.sh script2 text.txt
2.tmp 5.tmp lab11_2.sh presentation script2.c
3.tmp fout.txt lab11_3.sh report script2.o
[vignonagossou@vignonagossou lab11]$
```

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

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 screenshot shows a terminal window with the following commands and output:

```
vignonagossou@vignonagossou:~/work/study/2022-2023/One...  
[vignonagossou@vignonagossou lab11]$ chmod +x lab11_4.sh  
[vignonagossou@vignonagossou lab11]$ ./lab11_4.sh  
tar: ./arhiv.tar: файл является архивом, но cpioadm  
[vignonagossou@vignonagossou lab11]$ ls  
1.tmp 4.tmp fout.txt lab11_3.sh report script2.o  
2.tmp 5.tmp lab11_1.sh lab11_4.sh script2 text.txt  
3.tmp arhiv.tar lab11_2.sh presentation script2.c  
[vignonagossou@vignonagossou lab11]$
```

On the right, a smaller window titled 'lab11_4.sh' shows the script's content:

```
#!/bin/bash  
2 (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.