

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Оксана Чумаченко

2 сентября 2025

Российский университет дружбы народов, Москва, Россия

Цели и задачи работы

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

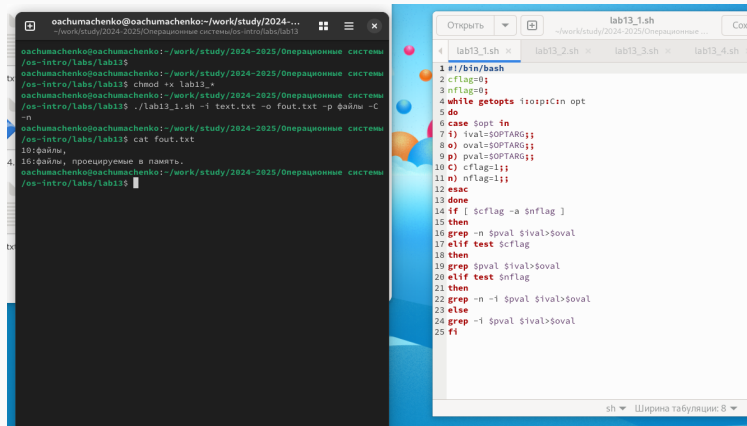
1 Выполнить 4 задания

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; `-p 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 displays two terminal windows side-by-side. The left window shows a user named 'oachumachenko' at a prompt, navigating through a directory structure and executing a script 'lab13_1.sh'. The right window shows the source code of 'lab13_1.sh', which is a shell script that sets flags and uses a while loop to process command-line options.

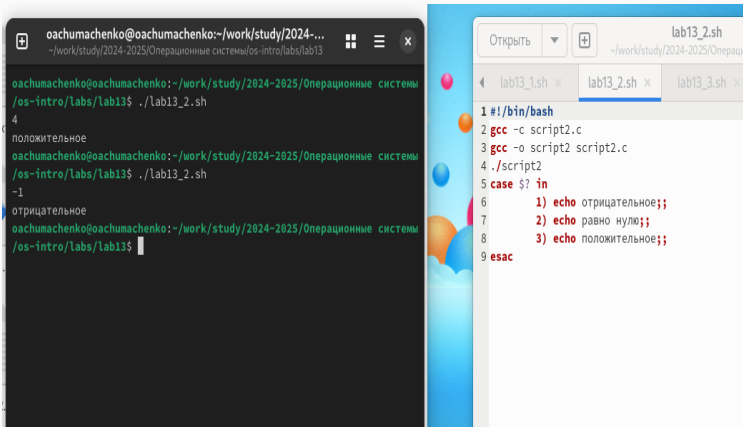
```
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ chmod +x lab13_*  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C  
-n  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ cat fout.txt  
10:файлы,  
16:файлы, проецируемые в память.  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
```

```
#!/bin/bash  
cflag=0;  
nflag=0;  
while getopts i:osp: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



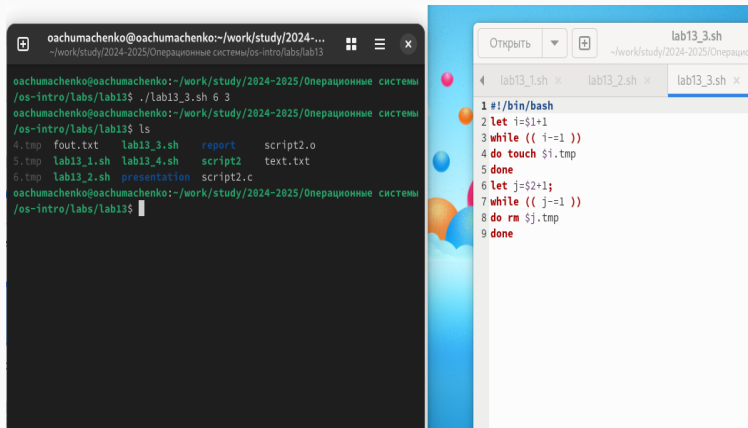
```
oachumachenko@oachumachenko:~/work/study/2024-...  
~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13  
  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы  
/os-intro/labs/lab13$ ./lab13_2.sh  
4  
положительное  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы  
/os-intro/labs/lab13$ ./lab13_2.sh  
-1  
отрицательное  
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы  
/os-intro/labs/lab13$
```

```
lab13_2.sh  
~/work/study/2024-2025/Операц  
  
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 consists of two side-by-side screenshots. The left screenshot shows a terminal window with the following text:

```
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_3.sh 6 3
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ls
4.tmp  fout.txt  lab13_3.sh  report  script2.o
5.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt
6.tmp  lab13_2.sh  presentation  script2.c
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
```

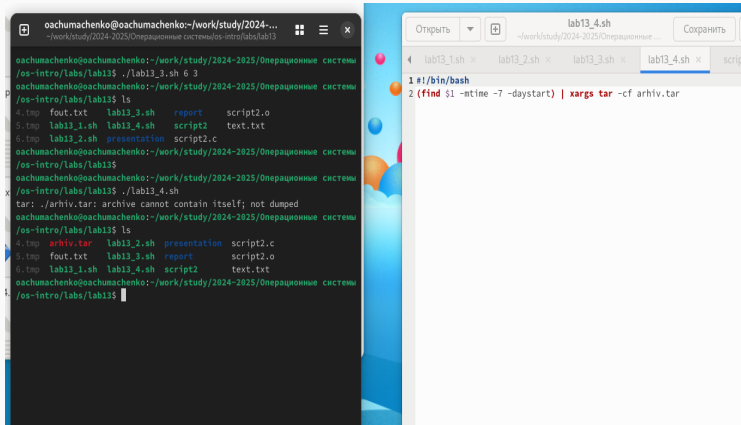
The right screenshot shows a code editor window titled 'lab13_3.sh' with the following script content:

```
1#!/bin/bash
2let i=$1+1
3while (( i--=1 ))
4do touch $i.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



```
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_3.sh 6 3
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ls
4.tmp  fout.txt  lab13_3.sh  report  script2.o
5.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt
6.tmp  lab13_2.sh  presentation  script2.c
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$ ls
4.tmp  arhiv.tar  lab13_2.sh  presentation  script2.c
5.tmp  fout.txt  lab13_3.sh  report  script2.o
6.tmp  lab13_1.sh  lab13_4.sh  script2  text.txt
oachumachenko@oachumachenko:~/work/study/2024-2025/Операционные системы/os-intro/labs/lab13$
```

```
lab13_4.sh
1 #!/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.