

Операционные системы

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Чернятин Артём Андреевич

2025-12-10

1. Цели и задачи работы

2. Процесс выполнения лабораторной работы

3. Выводы по проделанной работе

1. Цели и задачи работы

1.1 Цель лабораторной работы

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

1.2 Задачи лабораторной работы

1 Выполнить 4 задания

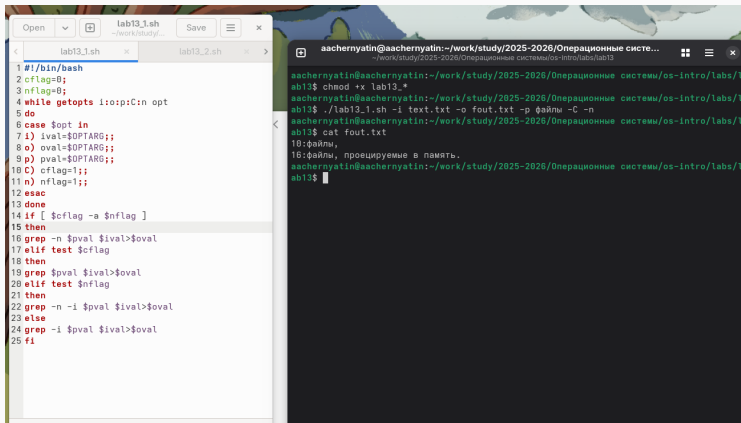
2. Процесс выполнения лабораторной работы

2.1 Выполнение 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

2.2 Выполнение работы



The image shows two terminal windows side-by-side. The left window, titled 'lab13_1.sh', displays a shell script with 25 lines of code. The script starts with a shebang, sets flags, and uses a while loop to process command-line options. It then uses a case statement to handle different options, setting variables and flags. The right window, titled 'aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13', shows the execution of the script. It runs 'chmod +x lab13.*', then './lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n'. The output shows the script processing the options and printing the file 'файлы'.

```
1 #!/bin/bash
2 cflag=0;
3 nflag=0;
4 while getopts i:op:C:n opt
5 do
6 case $opt in
7 i) ival=$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 $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $nflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi
```

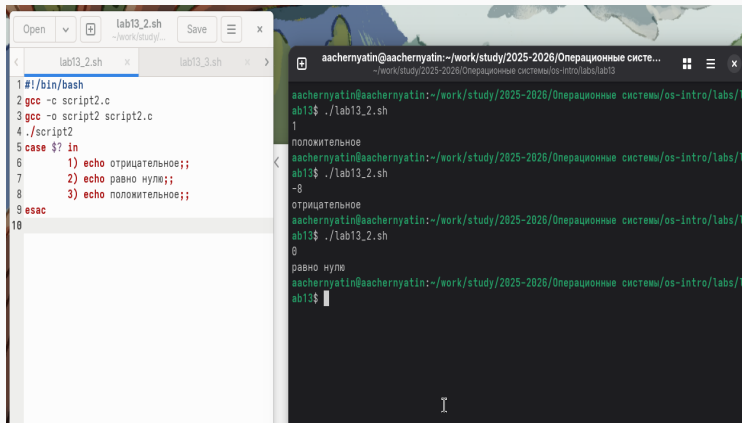
```
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13$ chmod +x lab13.*
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13$ ./lab13_1.sh -i text.txt -o fout.txt -p файлы -C -n
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13$ cat fout.txt
10:файлы,
16:файлы, проецируемые в память.
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13$
```

Рисунок 1: Задание 1

2.3 Выполнение работы

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

2.4 Выполнение работы



The image shows a code editor on the left and a terminal window on the right. The code editor displays a shell script named `lab13_2.sh` with the following content:

```
1#!/bin/bash
2gcc -c script2.c
3gcc -o script2 script2.c
4./script2
5case $? in
6    1) echo отрицательное;;
7    2) echo равно нулю;;
8    3) echo положительное;;
9esac
10
```

The terminal window shows the execution of the script. The user runs `./lab13_2.sh`, and the output is:

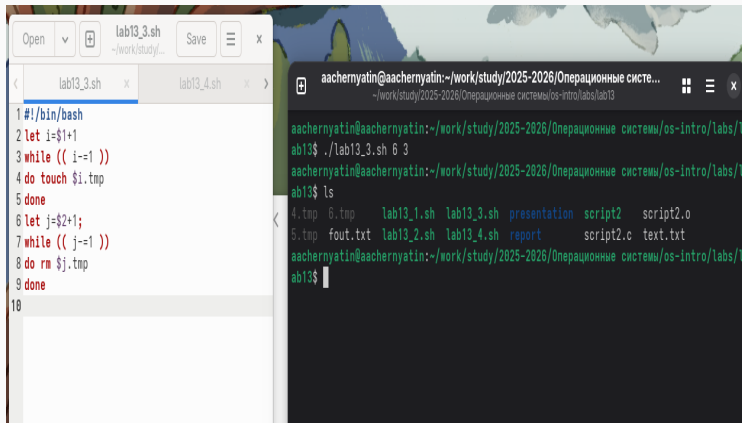
```
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные систе...
ab13$ ./lab13_2.sh
1
положительное
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные систе...
ab13$ ./lab13_2.sh
-8
отрицательное
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные систе...
ab13$ ./lab13_2.sh
0
равно нулю
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные систе...
ab13$
```

Рисунок 2: Задание 2

2.5 Выполнение работы

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

2.6 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 `lab13_3.sh` with the following 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
10
```

The terminal window shows the execution of the script. The user runs `./lab13_3.sh 6 3`, and the output shows the files created and removed:

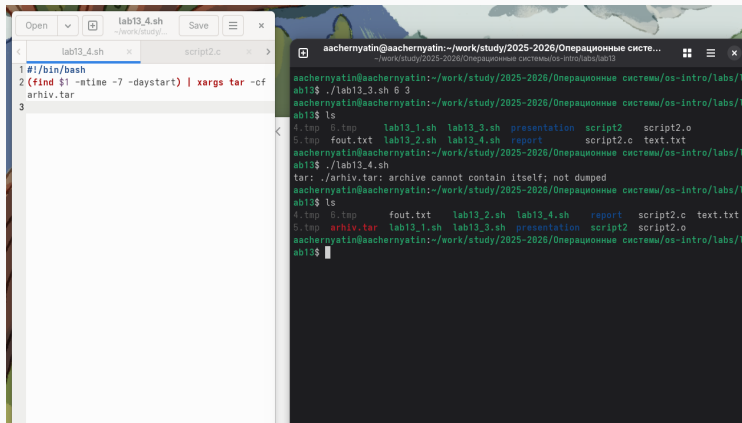
```
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные систе...
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$ ./lab13_3.sh 6 3
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$ ls
4.tmp 6.tmp  lab13_1.sh lab13_3.sh presentation script2  script2.o
5.tmp fout.txt lab13_2.sh lab13_4.sh report      script2.c text.txt
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$
```

Рисунок 3: Задание 3

2.7 Выполнение работы

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.

2.8 Выполнение работы



The image shows a code editor window on the left and a terminal window on the right. The code editor has tabs for 'lab13_4.sh' and 'script2.c'. The 'lab13_4.sh' tab is active, showing a script with three lines: a shebang, a find command, and a tar command. The terminal window shows the execution of the script, including directory listings and the output of the tar command.

```
1 #!/bin/bash
2 (find $1 -mtime -7 -daystart) | xargs tar -cf
  arhiv.tar
3
```

```
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные систе...
~/work/study/2025-2026/Операционные системы/os-intro/labs/lab13

aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$ ./lab13_3.sh 6 3
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$ ls
4.tmp  6.tmp      lab13_1.sh  lab13_3.sh  presentation  script2      script2.o
5.tmp  fout.txt  lab13_2.sh  lab13_4.sh  report        script2.c    text.txt
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$ ./lab13_4.sh
tar: ./arhiv.tar: archive cannot contain itself; not dumped
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$ ls
4.tmp  6.tmp      fout.txt    lab13_2.sh  lab13_4.sh  report      script2.c  text.txt
5.tmp  arhiv.tar  lab13_1.sh  lab13_3.sh  presentation  script2     script2.o
aachernyatin@aachernyatin:~/work/study/2025-2026/Операционные системы/os-intro/labs/l
ab13$
```

Рисунок 4: Задание 4

3. Выводы по проделанной работе

3.1 Вывод

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