

Лабораторная работа № 13

Баптишта Матеуж Андре ; Нкабв-01-23

Российский университет дружбы народов, Москва, Россия

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

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-iinputfile` — прочитать данные из указанного файла;
 - `-ooutputfile` — вывести данные в указанный файл;
 - `-ршаблон` — указать шаблон для поиска;
 - `-C` — различать большие и малые буквы;
 - `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.

2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено.

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов,которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

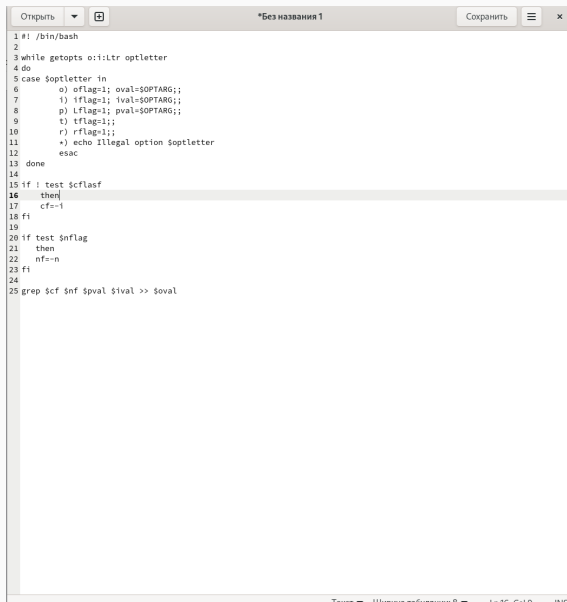
Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: - оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;

- C-оболочка (или csh) — надстройка на оболочкой Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
- оболочка Корна (или ksh) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна;
- BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation).

POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ. Стандарты POSIX разработаны комитетом IEEE (Institute of Electrical and Electronics Engineers) для обеспечения совместимости различных UNIX/Linux-подобных операционных систем и переносимости прикладных программ на уровне исходного кода. POSIX-совместимые оболочки разработаны на базе оболочки Корна. Рассмотрим основные элементы программирования в оболочке `bash`. В других оболочках большинство команд будет совпадать с описанными ниже.

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами:
 - `-iinputfile` — прочитать данные из указанного файла;
 - `-ooutputfile` — вывести данные в указанный файл;
 - `-ршаблон` — указать шаблон для поиска;
 - `-C` — различать большие и малые буквы;
 - `-n` — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом `-р`. (рис. (fig:001?; fig:002?; fig:003?; fig:004?))

Выполнение лабораторной работы



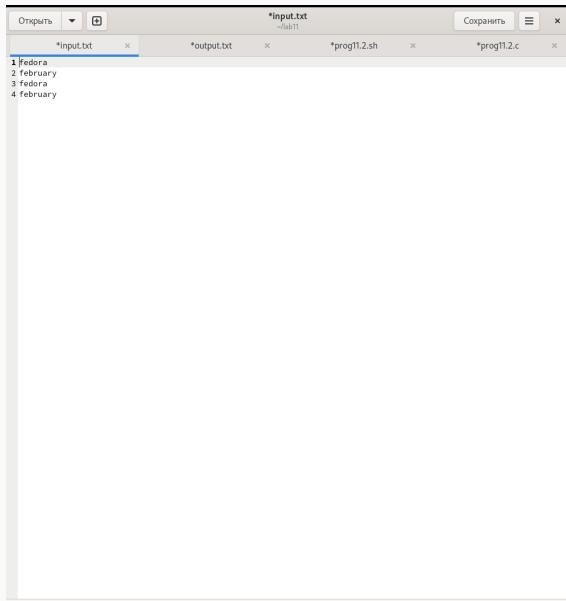
The screenshot shows a text editor window with a title bar containing "Открыть", a file icon, "*Без названия 1", "Сохранить", and window control buttons. The editor contains a shell script with 25 lines of code. Line 16, which begins with "then", is highlighted with a light gray background. The script is as follows:

```
1 #! /bin/bash
2
3 while getopts o:i:ltr optletter
4 do
5 case $optletter in
6   o) oflag=1; oval=$OPTARG;;
7   i) iflag=1; ival=$OPTARG;;
8   p) lflag=1; pval=$OPTARG;;
9   t) tflag=1;;
10  r) rflag=1;;
11  *) echo illegal option $optletter
12     esac
13 done
14
15 if ! test $cflag
16 then
17   cf=-i
18 fi
19
20 if test $nflag
21 then
22   nf=-n
23 fi
24
25 grep $cf $nf $pval $ival >> $oval
```

Выполнение лабораторной работы

```
mabaptishta@fedora:/home/mabaptishta/lab11
mabaptishta@fedora:~$ cd lab11
mabaptishta@fedora:~/lab11$ gedit
mabaptishta@fedora:~/lab11$ gedit input.txt &
[1] 5119
mabaptishta@fedora:~/lab11$ bash prog11.1.sh -p fe -i input.txt -o output.txt -c -n
bash: prog11.1.sh: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$ gedit input.txt &
[2] 5160
mabaptishta@fedora:~/lab11$ gedit output.txt &
[3] 5169
[2] Завершён      gedit input.txt
mabaptishta@fedora:~/lab11$ bash prog11.1.sh -p fe -i input.txt -o output.txt -c -n
bash: prog11.1.sh: Нет такого файла или каталога
[3]+ Завершён      gedit output.txt
mabaptishta@fedora:~/lab11$ gedit prog11.2.sh
mabaptishta@fedora:~/lab11$ gedit prog11.2.c
mabaptishta@fedora:~/lab11$
```

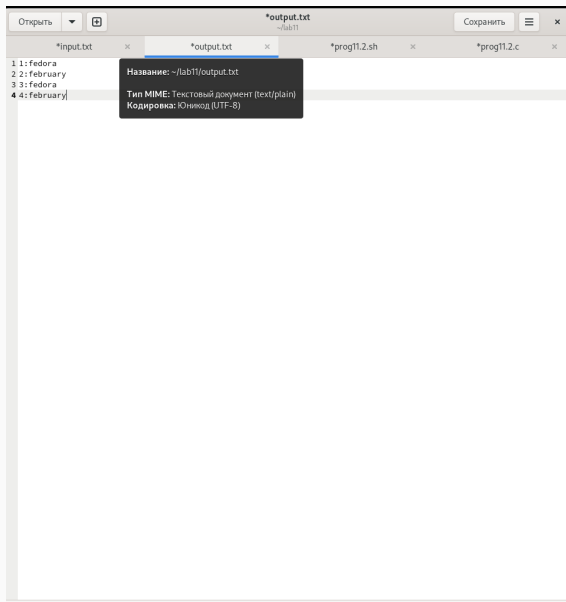
Выполнение лабораторной работы



The screenshot shows a terminal window with a file editor interface. The title bar indicates the current file is `*input.txt` located in the `~/lab11` directory. The interface includes a menu bar with 'Открыть' (Open) and 'Сохранить' (Save) options, and a tab bar showing four open files: `*input.txt`, `*output.txt`, `*prog11.2.sh`, and `*prog11.2.c`. The `*input.txt` tab is active, displaying the following content:

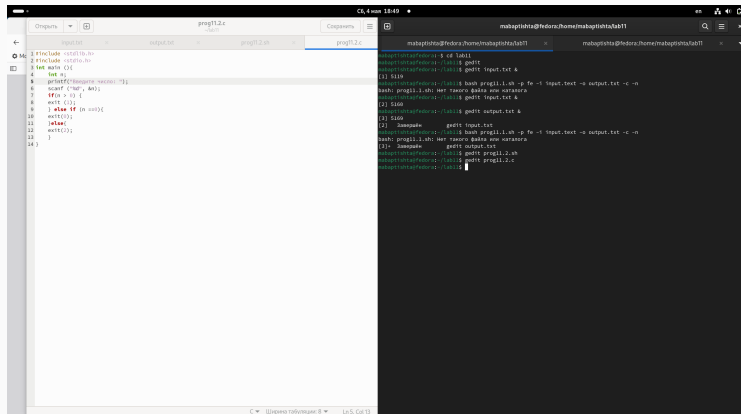
```
1 fedora
2 february
3 fedora
4 february
```

Выполнение лабораторной работы



2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено. (рис. (fig:005?; fig:006?; fig:007?))

Выполнение лабораторной работы

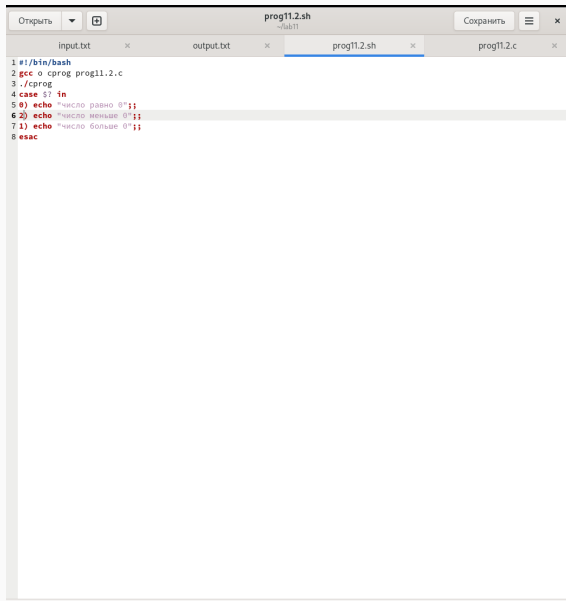


```
prog11.2.c
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main () {
4     int n;
5     printf("Введите значение n: ");
6     scanf ("%d", &n);
7     if(n > 0) {
8         exit (1);
9     } else if (n == 0) {
10        exit(0);
11    } else {
12        exit(2);
13    }
14 }
```

```
prog11.2.sh
1 #!/bin/bash
2 prog11.2.c -o prog11.2
3 prog11.2
```

Рис. 5: Вторая программа

Выполнение лабораторной работы



The screenshot shows a code editor window titled "prog11.2.sh" with a file explorer on the left. The editor contains a shell script with the following lines:

```
1 #!/bin/bash
2 gcc o cprog prog11.2.c
3 ./cprog
4 case $? in
5 0) echo "число равно 0";;
6 2) echo "число меньше 0";;
7 1) echo "число больше 0";;
8 esac
```

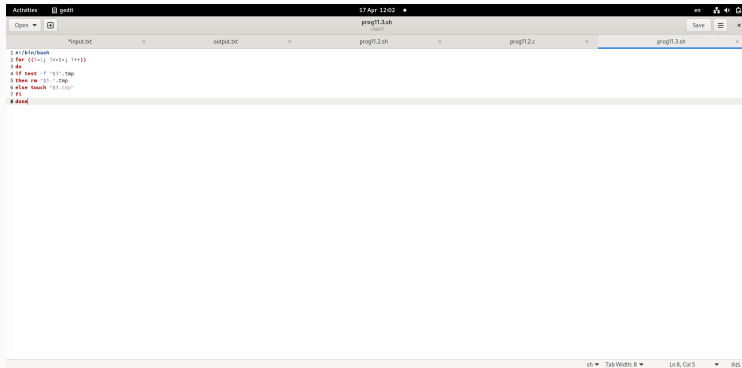
The file explorer on the left shows four files: "input.txt", "output.txt", "prog11.2.sh" (selected), and "prog11.2.c".

Выполнение лабораторной работы

```
mabaptishta@fedora:/home/mabaptishta/lab11
mabaptishta@fedora:/home/mabaptishta/lab11 x mabaptishta@fedora:/home/mabaptishta/lab11 x
mabaptishta@fedora:~/lab11$ bash prog11.2.sh
/usr/bin/ld: невозможно найти о: Нет такого файла или каталога
/usr/bin/ld: невозможно найти cprog: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
prog11.2.sh: строка 3: ./cprog: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$ bash prog11.2.sh
/usr/bin/ld: невозможно найти о: Нет такого файла или каталога
/usr/bin/ld: невозможно найти cprog: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
prog11.2.sh: строка 3: ./cprog: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$
```


3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).
(рис. (fig:008?; fig:009?))

Выполнение лабораторной работы



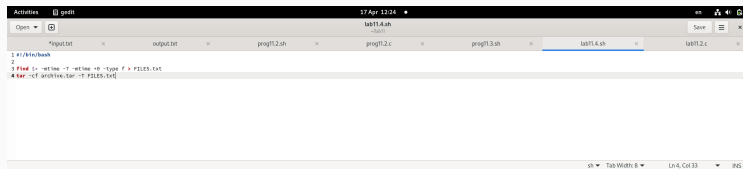
```
1 #!/bin/bash
2 for ((i=1; i<=5; i++))
3 do
4     if test -f "$i".tap
5     then rm "$i".tap
6     else touch "$i".tap
7     fi
8 done
```

Рис. 8: Третья программа

Выполнение лабораторной работы

```
mabaptishta@fedora:/home/mabaptishta/lab11
x mabaptishta@fedora:/home/mabaptishta/lab11 x mabaptishta@fedora:/home/mabaptishta/lab11
x
mabaptishta@fedora:~$ cd lab11
mabaptishta@fedora:~/lab11$ gedit
mabaptishta@fedora:~/lab11$ gedit input.txt &
19
mabaptishta@fedora:~/lab11$ bash prog11.1.sh -p fe -i input.txt -o output.txt -c -n
prog11.1.sh: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$ gedit input.txt &
20
mabaptishta@fedora:~/lab11$ gedit output.txt &
21
Завершён      gedit input.txt
mabaptishta@fedora:~/lab11$ bash prog11.1.sh -p fe -i input.txt -o output.txt -c -n
prog11.1.sh: Нет такого файла или каталога
Завершён      gedit output.txt
mabaptishta@fedora:~/lab11$ gedit prog11.2.sh
mabaptishta@fedora:~/lab11$ gedit prog11.2.c
mabaptishta@fedora:~/lab11$
```

4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду `find`). (рис. (fig:010?; fig:011?; fig:012?))



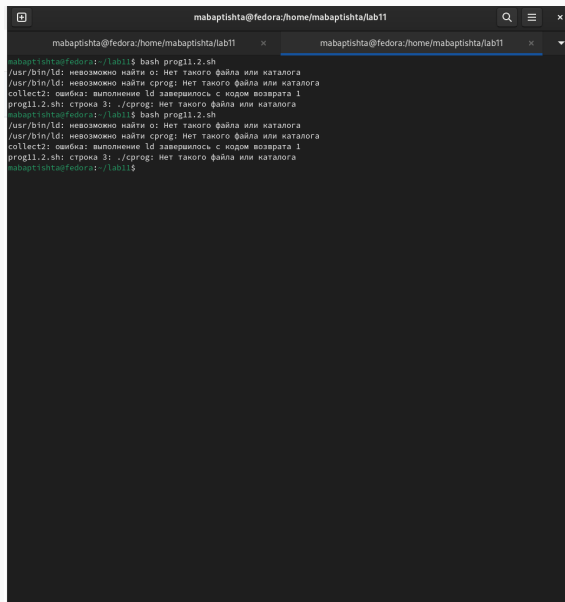
The screenshot shows a terminal window titled "lab11.4.sh" with a menu bar containing "Activities", "edit", and "17 Apr 13:24". The window has several tabs: "input.txt", "output.txt", "prog11.2.sh", "prog11.2.c", "prog11.3.sh", "lab11.4.sh" (active), and "lab11.2.c". The terminal content is as follows:

```
1 #!/bin/bash
2
3 find -s -mtime -7 -mtime +0 -type f > FILES.txt
4 tar -cf archive.tar -T FILES.txt
```

The status bar at the bottom indicates "sh", "Tab Width: 8", "Ln 4, Col 33", and "ihs".

Рис. 10: Четвертая программа

Выполнение лабораторной работы



The image shows a terminal window with two tabs. The active tab is titled 'mabaptishta@fedora:/home/mabaptishta/lab11'. The terminal content shows a user running a script 'prog11.2.sh'. The script contains a loop that calls 'ld' to compile a program. Each iteration results in an error: '/usr/bin/ld: невозможно найти о: Нет такого файла или каталога' and 'collect2: ошибка: выполнение ld завершилось с кодом возврата 1'. The script then calls 'cp' to copy the file, which also fails with the same error. The terminal output is as follows:

```
mabaptishta@fedora:~/lab11$ bash prog11.2.sh
/usr/bin/ld: невозможно найти о: Нет такого файла или каталога
/usr/bin/ld: невозможно найти cprog: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
prog11.2.sh: строка 3: ./cprog: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$ bash prog11.2.sh
/usr/bin/ld: невозможно найти о: Нет такого файла или каталога
/usr/bin/ld: невозможно найти cprog: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
prog11.2.sh: строка 3: ./cprog: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$
```

Выполнение лабораторной работы

```
mabaptishta@fedora:/home/mabaptishta/lab11
mabaptishta@fedora:/home/mabaptishta/lab11 x mabaptishta@fedora:/home/mabaptishta/lab11 x
mabaptishta@fedora:~/lab11$ bash prog11.2.sh
/usr/bin/ld: невозможно найти о: Нет такого файла или каталога
/usr/bin/ld: невозможно найти cprog: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
prog11.2.sh: строка 3: ./cprog: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$ bash prog11.2.sh
/usr/bin/ld: невозможно найти о: Нет такого файла или каталога
/usr/bin/ld: невозможно найти cprog: Нет такого файла или каталога
collect2: ошибка: выполнение ld завершилось с кодом возврата 1
prog11.2.sh: строка 3: ./cprog: Нет такого файла или каталога
mabaptishta@fedora:~/lab11$ FILEA.txt
bash: FILEA.txt: команда не найдена...

^Z
[1]+  Остановлен  FILEA.txt
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$ FILES.txt
bash: FILES.txt: команда не найдена...
mabaptishta@fedora:~/lab11$ gedit FILES.txt
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
mabaptishta@fedora:~/lab11$
```

В процессе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

1. Лабораторная работа № 10. Программирование в командном процессоре ОС UNIX. Командные файлы [Электронный ресурс]. URL: <https://esystem.rudn.ru/>.

спасибо за внимание!