

Лабораторная работа №11

Цель работы

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

Задание

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,

Задание

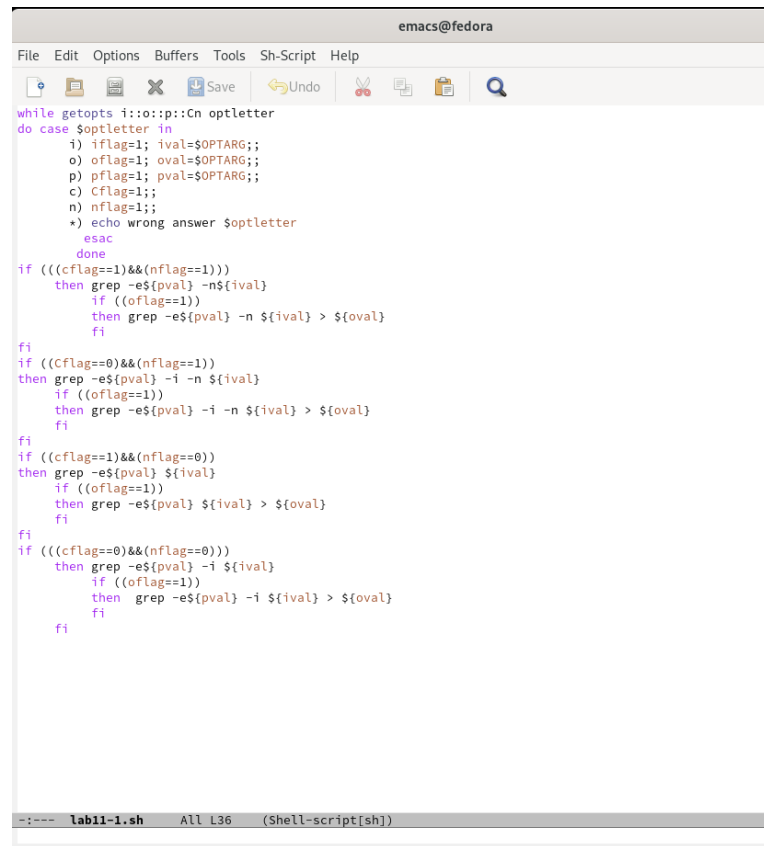
2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции `exit(n)`, передавая информацию в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав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

Задание

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).

Выполнение лабораторной работы

1. Используя команды `getopts` и `grep`, я написала командный файл, анализирующий командную строку с приведенными в задании ключами (рис.1-2).

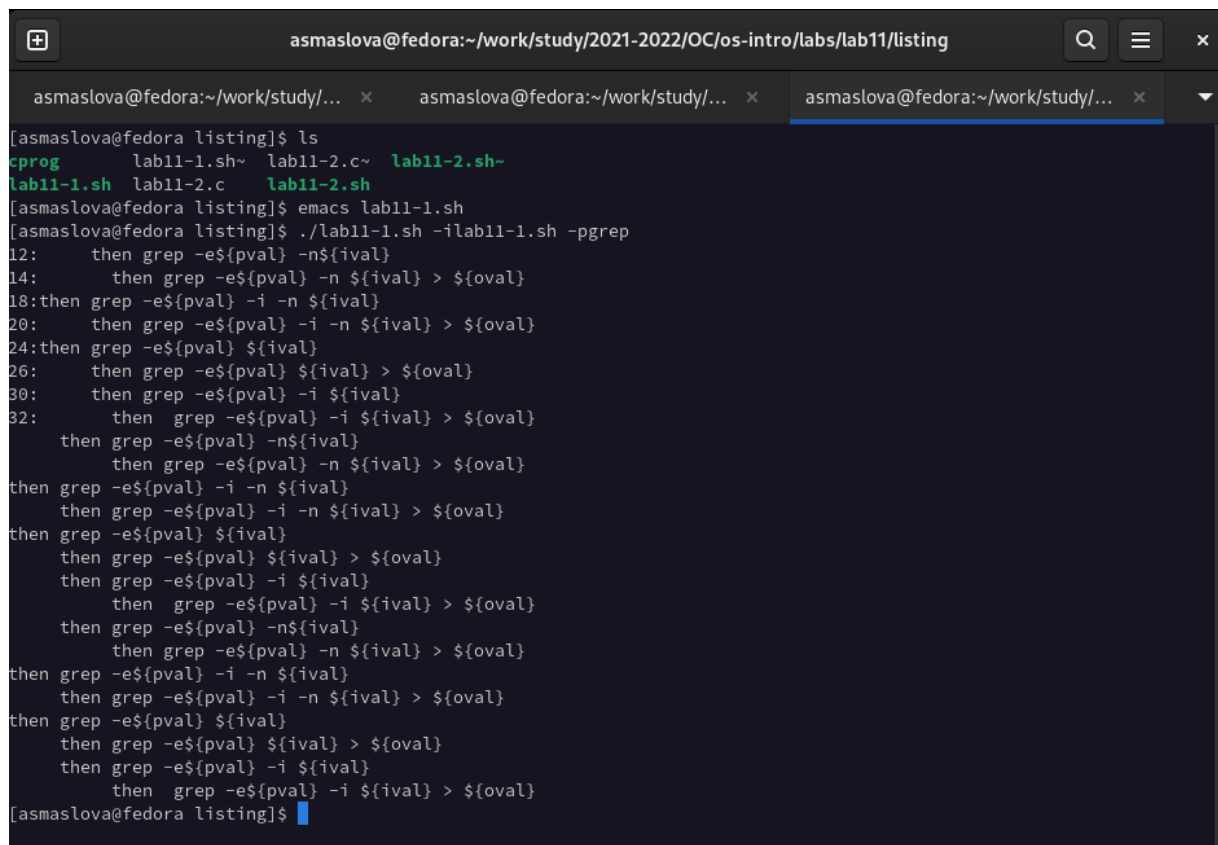


```
emacs@fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Redo, Find]

while getopts i::o::p::c:n optletter
do case $optletter in
  i) iflag=1; ival=$OPTARG;;
  o) oflag=1; oval=$OPTARG;;
  p) pflag=1; pval=$OPTARG;;
  c) Cflag=1;;
  n) nflag=1;;
  *) echo wrong answer $optletter
     esac
done
if (((cflag==1)&&(nflag==1)))
then grep -e${pval} -n${ival}
     if ((oflag==1))
     then grep -e${pval} -n ${ival} > ${oval}
     fi
fi
if ((cflag==0)&&(nflag==1))
then grep -e${pval} -i -n ${ival}
     if ((oflag==1))
     then grep -e${pval} -i -n ${ival} > ${oval}
     fi
fi
if ((cflag==1)&&(nflag==0))
then grep -e${pval} ${ival}
     if ((oflag==1))
     then grep -e${pval} ${ival} > ${oval}
     fi
fi
if (((cflag==0)&&(nflag==0)))
then grep -e${pval} -i ${ival}
     if ((oflag==1))
     then grep -e${pval} -i ${ival} > ${oval}
     fi
fi

-:--- lab11-1.sh All L36 ($Shell-script[sh])
```

рис.1 Код программы 1



```

[asmaslova@fedora listing]$ ls
cprog      lab11-1.sh~  lab11-2.c~  lab11-2.sh~
lab11-1.sh  lab11-2.c   lab11-2.sh

[asmaslova@fedora listing]$ emacs lab11-1.sh
[asmaslova@fedora listing]$ ./lab11-1.sh -ilab11-1.sh -pgrep
12:      then grep -e${pval} -n${ival}
14:      then grep -e${pval} -n ${ival} > ${oval}
18:then grep -e${pval} -i -n ${ival}
20:      then grep -e${pval} -i -n ${ival} > ${oval}
24:then grep -e${pval} ${ival}
26:      then grep -e${pval} ${ival} > ${oval}
30:      then grep -e${pval} -i ${ival}
32:      then grep -e${pval} -i ${ival} > ${oval}
      then grep -e${pval} -n${ival}
      then grep -e${pval} -n ${ival} > ${oval}
then grep -e${pval} -i -n ${ival}
      then grep -e${pval} -i -n ${ival} > ${oval}
then grep -e${pval} ${ival}
      then grep -e${pval} ${ival} > ${oval}
      then grep -e${pval} -i ${ival}
      then grep -e${pval} -i ${ival} > ${oval}
      then grep -e${pval} -n${ival}
      then grep -e${pval} -n ${ival} > ${oval}
then grep -e${pval} -i -n ${ival}
      then grep -e${pval} -i -n ${ival} > ${oval}
then grep -e${pval} ${ival}
      then grep -e${pval} ${ival} > ${oval}
      then grep -e${pval} -i ${ival}
      then grep -e${pval} -i ${ival} > ${oval}
[asmaslova@fedora listing]$

```

рис.2 Пример работы программы

2. На языке С я написала программу,
определяющую, больше, меньше или равно
нулю введенное пользователем число (рис.3-5).

The image shows a screenshot of the Emacs editor window titled "emacs@fedora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "C", and "Help". The toolbar contains icons for opening files, saving, undo, redo, and search. The main text area displays a C++ program:

```
#include <iostream>
int main(){
    int x;
    std::cin>>x;
    if (x==0) exit(0);
    else if(x<0) exit(1);
    else if(x>0) exit(2);
    return 3;
}
```

Below the code, the Emacs startup screen is visible. It shows the file name "lab11-2.c", the buffer "All L8", and the mode "(C/*l Abbrev)". The startup screen includes a welcome message, a list of links for learning Emacs, and the GNU Emacs version and copyright information.

--:--- lab11-2.c All L8 (C/*l Abbrev)
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial	Learn basic keystroke commands
Emacs Guided Tour	Overview of Emacs features at gnu.org
View Emacs Manual	View the Emacs manual using Info
Absence of Warranty	GNU Emacs comes with <i>ABSOLUTELY NO WARRANTY</i>
Copying Conditions	Conditions for redistributing and changing Emacs
Ordering Manuals	Purchasing printed copies of manuals

To quit a partially entered command, type Control-g.

This is GNU Emacs 27.2 (build 1, x86_64-redhat-linux-gnu, GTK+ Version 3.24.30, cairo version 1.17.4)
of 2021-08-07
Copyright (C) 2021 Free Software Foundation, Inc.

[Dismiss this startup screen](#) ☐ Never show it again.

U:%%- *GNU Emacs* All L8 (Fundamental)
(No changes need to be saved)

рис.3 Код программы 2.1

The image shows a screenshot of the Emacs editor window titled "emacs@fedora". The window has a menu bar with "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". Below the menu bar is a toolbar with icons for file operations and editing. The main text area contains a C++ program:

```
g++ -o cprog lab11-2.c
./cprog
case $? in
0)echo "number is equal to zero";;
1) echo "number is smaller than zero";;
2) echo "number is bigger than zero";;
esac
```

Below the code editor is a status bar showing the current buffer as "lab11-2.sh", the major mode as "All L4", and the shell as "(Shell-script[sh])". Below the status bar is a startup screen with the following text:

Welcome to [GNU Emacs](#), one component of the [GNU/Linux](#) operating system.

Emacs Tutorial	Learn basic keystroke commands
Emacs Guided Tour	Overview of Emacs features at gnu.org
View Emacs Manual	View the Emacs manual using Info
Absence of Warranty	GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions	Conditions for redistributing and changing Emacs
Ordering Manuals	Purchasing printed copies of manuals

To quit a partially entered command, type Control-g.

This is GNU Emacs 27.2 (build 1, x86_64-redhat-linux-gnu, GTK+ Version 3.24.30, cairo version 1.17.4)
of 2021-08-07
Copyright (C) 2021 Free Software Foundation, Inc.

[Dismiss this startup screen](#) ☐ Never show it again.

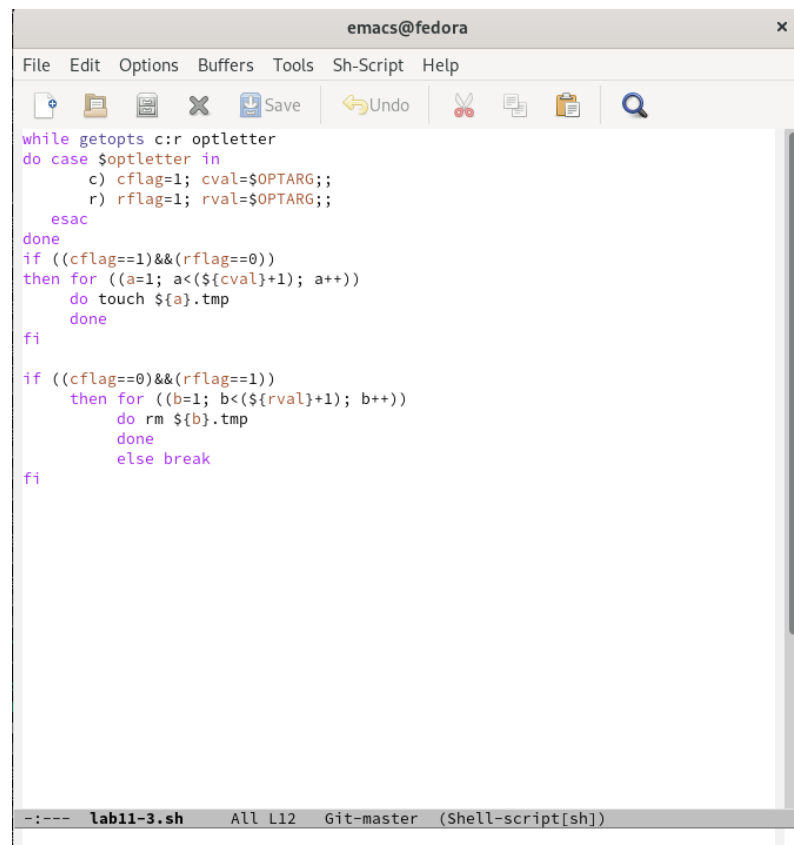
The status bar at the bottom shows "U:%%- *GNU Emacs* All L3 (Fundamental)".

рис.4 Код программы 2.2

```
[asmaslova@fedora listing]$ emacs lab11-2.c &  
[2] 7920  
[asmaslova@fedora listing]$ ./lab11-2.sh  
5  
number is bigger than zero  
[asmaslova@fedora listing]$ 0  
bash: 0: command not found...  
[asmaslova@fedora listing]$ ./lab11-2.sh  
0  
number is equal to zero  
[asmaslova@fedora listing]$ ./lab11-2.sh  
-1513  
number is smaller than zero  
[asmaslova@fedora listing]$
```

рис.5 Пример работы программы

3. Далее я написала командный файл, создающий указанное пользователем количество файлов, пронумерованных последовательно от 1 до указанного числа (рис.6-7).



```
emacs@fedora
File Edit Options Buffers Tools Sh-Script Help
Save Undo
while getopts c:r optletter
do case $optletter in
  c) cflag=1; cval=$OPTARG;;
  r) rflag=1; rval=$OPTARG;;
  esac
done
if ((cflag==1)&&(rflag==0))
then for ((a=1; a<(${cval}+1); a++))
do touch ${a}.tmp
done
fi
if ((cflag==0)&&(rflag==1))
then for ((b=1; b<(${rval}+1); b++))
do rm ${b}.tmp
done
else break
fi
fi

-:--- lab11-3.sh All L12 Git-master (Shell-script[sh])
```

рис.6 Код программы 3

```
[asmaslova@fedora listing]$ ./lab11-3.sh -c10
[asmaslova@fedora listing]$ ls
10.tmp  5.tmp  chern.sh      lab11-2.c      lab11-3.sh~
1.tmp   6.tmp  chern.sh~     lab11-2.c~
2.tmp   7.tmp  cprog        lab11-2.sh
3.tmp   8.tmp  lab11-1.sh   lab11-2.sh~
4.tmp   9.tmp  lab11-1.sh~   lab11-3.sh
[asmaslova@fedora listing]$ ./lab11-3.sh -r10
./lab11-3.sh: illegal option -r
```

рис.7 Пример работы программы

Выводы

- В результате выполнения лабораторной работе я получила навыки написания более сложных командных файлов с использованием логических управляющих конструкций и циклов.

Список литературы