

Лабораторная работа 11 Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Хусаинова Динара Айратовна

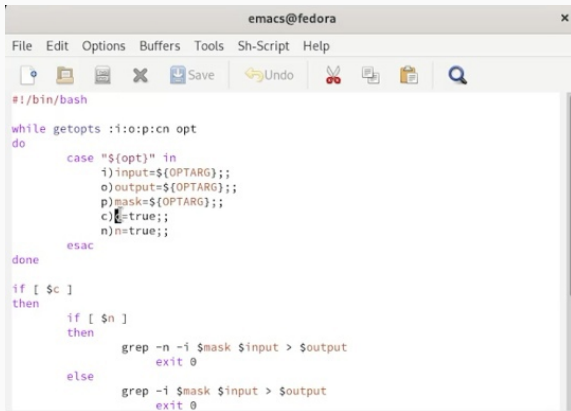
25.05.2022

RUDN

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

Первый командный файл

Используя команды `getopts` `grep`, напомним командный файл, который анализирует командную строку(рис. 1).

The image shows a screenshot of an 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 opening files, saving, undo, redo, and search. The main text area contains a shell script written in a syntax-highlighted format. The script starts with a shebang line, followed by a while loop using getopts to parse command-line options. It then uses a case statement to handle each option, setting variables for input, output, mask, and flags. Finally, it uses an if-then-else structure to call the grep command with the appropriate flags and variables, and exits with a status code.

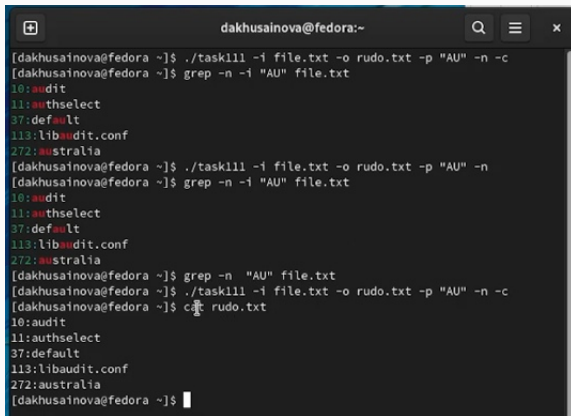
```
#!/bin/bash

while getopts :i:o:p:cn opt
do
    case "${opt}" in
        i) input=${OPTARG};;
        o) output=${OPTARG};;
        p) mask=${OPTARG};;
        c) c=true;;
        n) n=true;;
    esac
done

if [ $c ]
then
    if [ $n ]
    then
        grep -n -i $mask $input > $output
        exit 0
    else
        grep -i $mask $input > $output
        exit 0
    fi
fi
```

Figure 1: Первый файл

Работа файла(рис. 2).

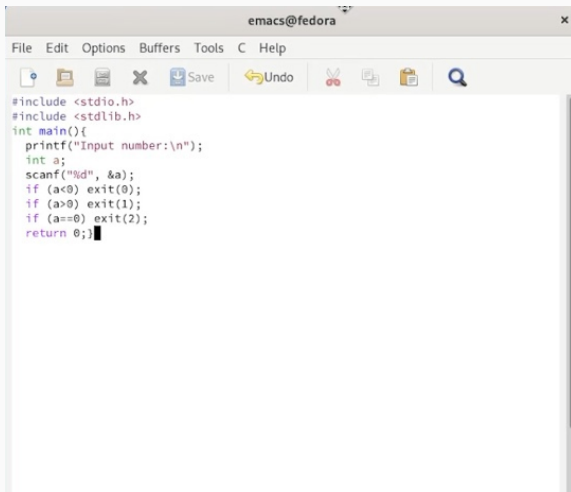


```
dakhusainova@fedora:~  
[dakhusainova@fedora ~]$ ./task111 -i file.txt -o rudo.txt -p "AU" -n -c  
[dakhusainova@fedora ~]$ grep -n -i "AU" file.txt  
10:audit  
11:authselect  
37:default  
113:libaudit.conf  
272:australia  
[dakhusainova@fedora ~]$ ./task111 -i file.txt -o rudo.txt -p "AU" -n  
[dakhusainova@fedora ~]$ grep -n -i "AU" file.txt  
10:audit  
11:authselect  
37:default  
113:libaudit.conf  
272:australia  
[dakhusainova@fedora ~]$ grep -n "AU" file.txt  
[dakhusainova@fedora ~]$ ./task111 -i file.txt -o rudo.txt -p "AU" -n -c  
[dakhusainova@fedora ~]$ cat rudo.txt  
10:audit  
11:authselect  
37:default  
113:libaudit.conf  
272:australia  
[dakhusainova@fedora ~]$
```

Figure 2: Результат

Второй командный файл

Напишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю(рис. 3,4,5).

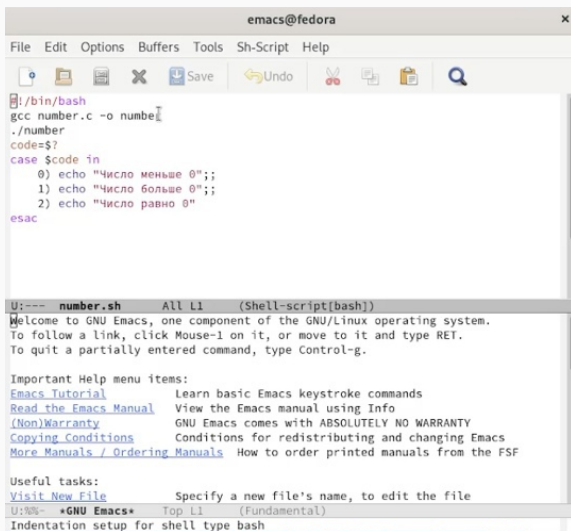


```
emacs@fedora
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Undo, Redo, Copy, Paste, Find]

#include <stdio.h>
#include <stdlib.h>
int main(){
    printf("Input number:\n");
    int a;
    scanf("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;}
```

Figure 3: На языке Си

Второй командный файл



The screenshot shows the Emacs editor window titled 'emacs@fedora'. The menu bar includes File, Edit, Options, Buffers, Tools, Sh-Script, and Help. The toolbar contains icons for opening files, saving, undo, redo, and search. The main text area displays a shell script named 'number.sh' with the following content:

```
#!/bin/bash
gcc number.c -o number
./number
code=$?
case $code in
  0) echo "Число меньше 0";;
  1) echo "Число больше 0";;
  2) echo "Число равно 0";;
esac
```

Below the script, the Emacs help screen is visible, showing the command prompt 'U:--- number.sh All L1 (Shell-script[bash])' and the following text:

Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.

Important Help menu items:

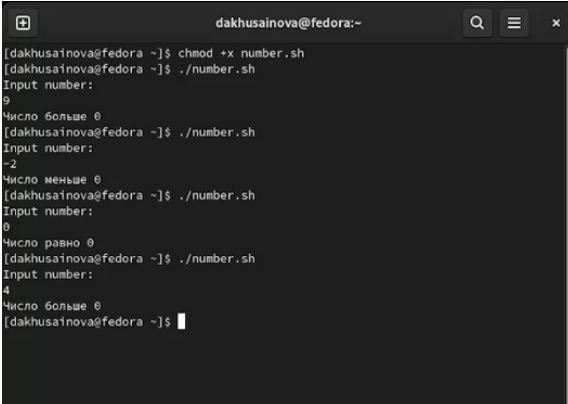
Emacs Tutorial	Learn basic Emacs keystroke commands
Read the Emacs Manual	View the Emacs manual using Info
(Non)Warranty	GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions	Conditions for redistributing and changing Emacs
More Manuals / Ordering Manuals	How to order printed manuals from the FSF

Useful tasks:

Visit New File	Specify a new file's name, to edit the file
--------------------------------	---

At the bottom, the command prompt shows 'U:%%- *GNU Emacs* Top L1 (Fundamental)' and the text 'Indentation setup for shell type bash'.

Figure 4: Содержимое файла

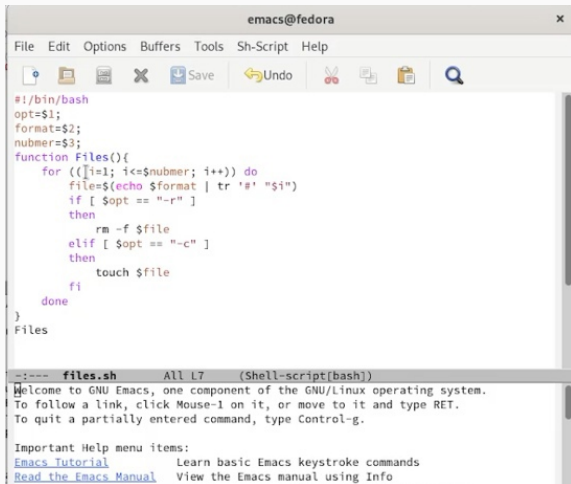


```
dakhusainova@fedora:~  
[dakhusainova@fedora ~]$ chmod +x number.sh  
[dakhusainova@fedora ~]$ ./number.sh  
Input number:  
9  
Число больше 0  
[dakhusainova@fedora ~]$ ./number.sh  
Input number:  
-2  
Число меньше 0  
[dakhusainova@fedora ~]$ ./number.sh  
Input number:  
0  
Число равно 0  
[dakhusainova@fedora ~]$ ./number.sh  
Input number:  
4  
Число больше 0  
[dakhusainova@fedora ~]$
```

Figure 5: Результат

Третий командный файл

Напишем командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до [?] (рис. 6,7).



The screenshot shows the Emacs editor window titled 'emacs@fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for creating a new file, opening a file, saving, undo, redo, and search. The main text area contains a shell script:

```
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function Files(){
  for ((i=1; i<=$number; i++)) do
    file=$(echo $format | tr '#' "$i")
    if [ $opt == "-r" ]
    then
      rm -f $file
    elif [ $opt == "-c" ]
    then
      touch $file
    fi
  done
}
Files
```

The status bar at the bottom shows the file path as '--- files.sh', the buffer as 'All L7', and the mode as '(Shell-script[bash])'. Below the status bar, the Emacs welcome message is displayed:

Welcome to GNU Emacs, one component of the GNU/Linux operating system.
To follow a link, click Mouse-1 on it, or move to it and type RET.
To quit a partially entered command, type Control-g.

Important Help menu items:
[Emacs Tutorial](#) Learn basic Emacs keystroke commands
[Read the Emacs Manual](#) View the Emacs manual using Info

Figure 6: Содержимое файла

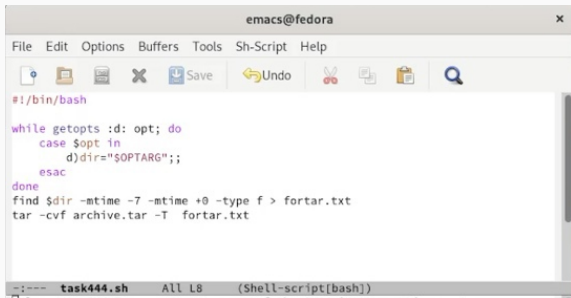

```
[dakhusainova@fedora ~]$ chmod +x files.sh
[dakhusainova@fedora ~]$ ./files.sh -c din#.txt 5
[dakhusainova@fedora ~]$ ls
```

abc1	feathers	lab07.sh~	ski.places	work
australia	file	may	task11	'Без имени 1'
backup	file1.sh	monthly	task111	Видео
backup.sh~	file2.sh	my_os	task222.sh	Документы
bill	file3.sh	non.txt	task2.sh	Загрузки
bin	file4	number	task2.sh~	Изображения
comp.cpp	files.sh	number~	task3.sh	Музыка
conf.txt	files.sh~	number.c	task3.sh~	Общедоступные
dakhusainova	file.txt	number.sh	task4.sh	'Рабочий стол'
din1.txt	go.txt	number.sh~	task4.sh~	Шаблоны
din2.txt	hot.txt	play	temp	
din3.txt	july	reports	text1.txt	
din4.txt	'#lab07.sh#'	rooot.txt	text.cpp	
din5.txt	lab07.sh	rudo.txt	text.txt	

Figure 7: Результат

Четвертый командный файл

Напишем 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(рис. 8,9).



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

#!/bin/bash

while getopts :d: opt; do
    case $opt in
        d)dir="$OPTARG";;
    esac
done
find $dir -mtime -7 -mtime +0 -type f > fortar.txt
tar -cvf archive.tar -T fortar.txt

--:-- task444.sh All L8 (Shell-script[bash])
```

Figure 8: Файл

```
[dakhusainova@fedora ~]$ ls
abc1      file4      my_os      task111     text.txt
[ archive.tar  files.sh  non.txt    task222.sh  worl
australia files.sh~  number     task2.sh~   'Без имени 1'
backup    file.txt   number~    task2.sh~   Видео
backup.sh~ fortar.txt number.c    task3.sh~   Документы
bill      go.txt    number.sh  task3.sh~   Загрузки
bin       hot.txt   number.sh~ task444.sh  Изображения
Catalog1  july      play       task444.sh~ Музыка
dakhusainova '#lab07.sh#' reports    task4.sh~   Общедоступные
file      lab07.sh  rooot.txt  task4.sh~   'Рабочий стол'
file1.sh  lab07.sh~ rudo.txt   temp        Шаблоны
file2.sh  may       ski.places text1.txt
file3.sh  monthly  task11     text.cpp
[dakhusainova@fedora ~]$ mkdir tmp
```

Figure 9: Результат

Мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.