

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №12

Дисциплина: Операционные системы

Студент: Акопян Изабелла Арменовна

Группа: НБИбд-01-2020

МОСКВА

2021 г.

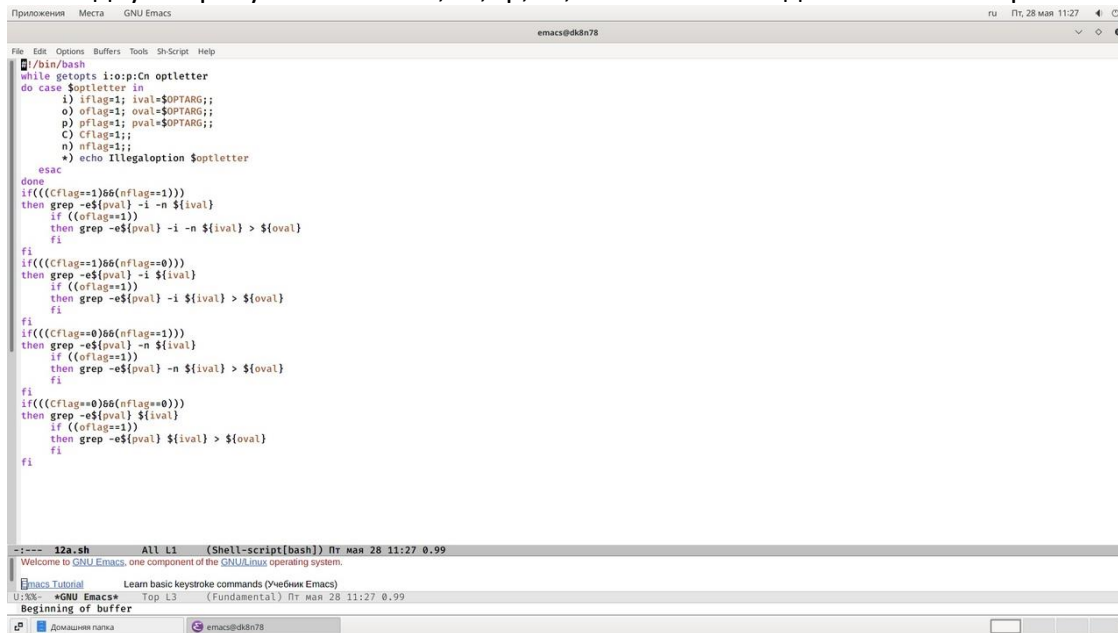
Цель:

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

Ход работы:

Ознакомилась в материалами лабораторной работы ([ссылка 1](#)) и приступила к выполнению заданий.

Используя команды getoptс grep, написала командный файл, который анализирует командную строку с ключами -i, -o, -p, -C, -n. Сначала создала текстовый файл.



```
#!/bin/bash
while getopts i:op:Cn 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 Illegaloption $optletter
  esac
done
if(((Cflag==1)&&(nflag==1)))
then grep -e${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} -i ${ival}
  if ((oflag==1))
  then grep -e${pval} -i ${ival} > ${oval}
  fi
fi
if(((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
  if ((oflag==1))
  then grep -e${pval} -n ${ival} > ${oval}
  fi
fi
if(((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
  if ((oflag==1))
  then grep -e${pval} ${ival} > ${oval}
  fi
fi
fi
```

The screenshot shows a GNU Emacs editor window with a menu bar at the top containing 'Приложения', 'Места', and 'GNU Emacs'. Below the menu bar is a toolbar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Text', and 'Help'. The main text area contains a C program with the following code:

```
hello how  
are you  
yellow submarine  
good luck!  
good morning.
```

Below the text area is a status bar showing ' -:--- test.txt All L1 (Text) Пт мая 28 11:49 0.44'. Below the status bar is a terminal window showing the execution of the program. The terminal output is as follows:

```
/afs/.dk.sci.pfu.edu.ru/home/i/a/iaakopyan/laboratory  
iaakopyan@dk8n78 ~/laboratory $ cd  
iaakopyan@dk8n78 - $ bash 12a.sh -itest.txt -otest1.txt -psubmarine  
yellow submarine  
iaakopyan@dk8n78 - $ bash 12a.sh -itest.txt -otest1.txt -pgood  
good luck!  
good morning.  
iaakopyan@dk8n78 - $
```

Написала 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 о коде завершения в оболочку. Командный файл вызывает эту программу и, проанализировав с помощью команды `$?`, выдал сообщение о том, какое число было введено.

Написать программу на Си мне помог сайт. [ссылка 2](#)

The screenshot displays a Linux desktop environment. At the top, there is a window titled "GNU Emacs" with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a status bar showing "emacs@dk8n78". The main editing area contains a C program named `failci.c` that uses `scanf` to read an integer and `if` statements to check if it is equal to, greater than, or less than zero, printing the results in Russian. Below the editor, a terminal window shows the user `iaakopyan@dk8n78` running `emacs failci.c`, `emacs 12b.sh`, and `bash 12b.sh`. The terminal output shows the program prompting for a number and responding accordingly: "Введите число: 1", "Это число больше 0", "Введите число: 0", "Это число равно 0", "Введите число: -1", "Это число меньше нуля". At the bottom of the screen, a taskbar shows the task manager with entries for a PDF file, the terminal, and OBS Studio.

```
File Edit Options Buffers Tools Sh-Script Help
emacs@dk8n78

./bin/bash
$C -o cprog failci.c
./cprog
case $? in
0)echo 'Это число равно 0';;
1)echo 'Это число меньше нуля';;
2)echo 'Это число больше 0';;
esac

U:~ 12b.sh All L1 (Shell-script[bash]) Пт мая 28 11:27 0.60
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
Learn basic keystroke commands (Руководство Emacs)
U:~*GNU Emacs* Top L3 (Fundamental) Пт мая 28 11:27 0.60

Приложения Места GNU Emacs
emacs@dk8n78

1#include <stdio.h>
2#include <stdlib.h>
3
4int main(){
5    int x;
6    printf("Введите число: ");
7    scanf("%i", &x);
8    if (x==0){
9        exit(0);
10    } else if (x<0){
11        exit(1);
12    } else if (x>0){
13        exit(2);
14    } return (0);
15}

Приложения Места Текстовый редактор
failci.c
Сохранить

Загрузка файла «/afs/dk.sci.pfu.edu.ru/home/ia/iaakopyan/failci.c»
C Ширина табуляции: 8 Стр 1, Стоб 1 ВСТ
emacs@dk8n78

iaakopyan@dk8n78 - $ emacs failci.c
iaakopyan@dk8n78 - $ emacs 12b.sh
iaakopyan@dk8n78 - $ bash 12b.sh
Введите число: 1
Это число больше 0
iaakopyan@dk8n78 - $ bash 12b.sh
Введите число: 0
Это число равно 0
iaakopyan@dk8n78 - $ bash 12b.sh
Введите число: -1
Это число меньше нуля
iaakopyan@dk8n78 - $
```

Написала командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые было необходимо создать, передавалось в аргументы командной строки. Этот же командный файл умеет удалять все созданные им файлы (если они существуют).

```
Приложения Места GNU Emacs
emacs@dk8n78
File Edit Options Buffers Tools Sh-Script Help
~/bin/bash
let dflag=0;
while getopts t:r optletter
do case $optletter in
t) aflag=1; aval=$OPTARG;;
r) dflag=1;;
*) echo illegaloption $optletter
esac
done
#echo $[aval]
if((dflag==0))
then for ((i=1;i<=aval;i++))
do touch ${i}.tmp
done
fi
if((dflag==1))
then for ((i=1;i<=aval;i++))
do rm ${i}.tmp
done
fi
fi

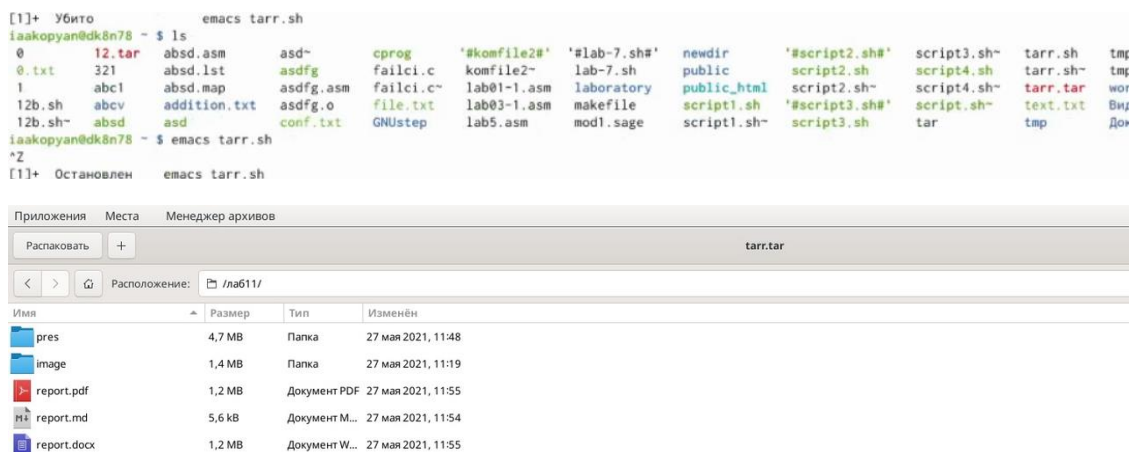
--tmp.sh All L1 (Shell-script[bash]) Пт мая 28 11:29 0.30
Welcome to GNU Emacs, one component of the GNU/Linux operating system.
Learn basic keystroke commands (Чтение Emacs)
U: 50% - GNU Emacs* Top L3 (Fundamental) Пт мая 28 11:29 0.30
Beginning of buffer
Домашняя папка emacs@dk8n78 emacs@dk8n78 failci(-)-gedit emacs@dk8n78
```

```
iaakopyan@dk8n78 ~ $ bash tmp.sh -t3
iaakopyan@dk8n78 ~ $ ls
0      1.tmp  abcv      addition.txt  asdfg.o      file.txt
0.txt  2.tmp  absd      asd           conf.txt     GNUstep
1      321    absd.asm  asd~          cprog        '#komfile2#'
12b.sh 3.tmp  absd.lst  asdfg         failci.c     komfile2~
12b.sh~ abc1   absd.map  asdfg.asm    failci.c~    lab01-1.asm
iaakopyan@dk8n78 ~ $ bash tmp.sh -t3 r
iaakopyan@dk8n78 ~ $ ls
0      1.tmp  abcv      addition.txt  asdfg.o      file.txt
0.txt  2.tmp  absd      asd           conf.txt     GNUstep
1      321    absd.asm  asd~          cprog        '#komfile2#'
12b.sh 3.tmp  absd.lst  asdfg         failci.c     komfile2~
12b.sh~ abc1   absd.map  asdfg.asm    failci.c~    lab01-1.asm
iaakopyan@dk8n78 ~ $ bash tmp.sh -t3 -r
iaakopyan@dk8n78 ~ $ ls
0      12b.sh~ absd      addition.txt  asdfg.asm    failci.c
0.txt  321    absd.asm  asd~          asdfg.o      failci.c~
1      abc1   absd.lst  asd~          conf.txt     file.txt
12b.sh abc1   absd.map  asdfg         cprog        GNUstep
iaakopyan@dk8n78 ~ $
```

Написала командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовала команду find).



Результат:



Вывод:

Я успешно изучила основы программирования в оболочке ОС UNIX. Научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Контрольные вопросы:

1. Каково предназначение команды getopt?

Весьма необходимой при программировании является команда getopt, которая осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

2. Какое отношение метасимволы имеют к генерации имён файлов?

* произвольная (возможно пустая) последовательность символов;

? один произвольный символ;

[...] любой из символов, указанных в скобках перечислением и/или с указанием диапазона;

cat f* выдаст все файлы каталога, начинающиеся с "f";

cat *f* выдаст все файлы, содержащие "f";

cat program.? выдаст файлы данного каталога с однобуквенными расширениями, скажем "program.c" и "program.o", но не выдаст "program.com";

cat [a-d]* выдаст файлы, которые начинаются с "a", "b", "c", "d". Аналогичный эффект дадут и команды "cat [abcd]*" и "cat [bdac]*".

3. Какие операторы управления действиями вы знаете?

Операторы && и || .

4. Какие операторы используются для прерывания цикла?

Оператор break завершает выполнение ближайшего включающего цикла или условного оператора, в котором он отображается.

5. Для чего нужны команды false и true?

Следующие две команды ОС UNIX используются только совместно с управляющими конструкциями языка программирования bash: это команда true, которая всегда возвращает код завершения, равный нулю (т.е. истина), и команда false, которая всегда возвращает код завершения, не равный нулю (т. е. ложь).

6. Что означает строка if test -f man\$s/\$i.\$s, встреченная в командном файле?

Введенная строка означает условие существования файла man\$s/\$i.\$s

7. Объясните различия между конструкциями while и until.

Цикл While выполняется до тех пор, пока указанное в нем условие истинно, а когда указанное условие становится ложным - цикл завершается. Цикл Until наоборот выполняется до тех пор, пока указанное в нем условие ложно.