

Лабораторная работа 12

Программирование в командном процессоре ОС UNIX. Ветвления и
циклы

Акопян Изабелла Арменовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	13
5	Библиография	14

Список таблиц

Список иллюстраций

3.1	Командный файл	7
3.2	Текстовый файл	8
3.3	Вызов	8
3.4	Командный файл	9
3.5	Код на Си	9
3.6	Вызов	9
3.7	Командный файл	10
3.8	Вызов	10
3.9	Командный файл	11
3.10	Вызов	11
3.11	ls	11
3.12	Архив	12

1 Цель работы

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

2 Задание

- написать 4 командных файла.
- написать программу на Си.
- работа с файлом .txt.

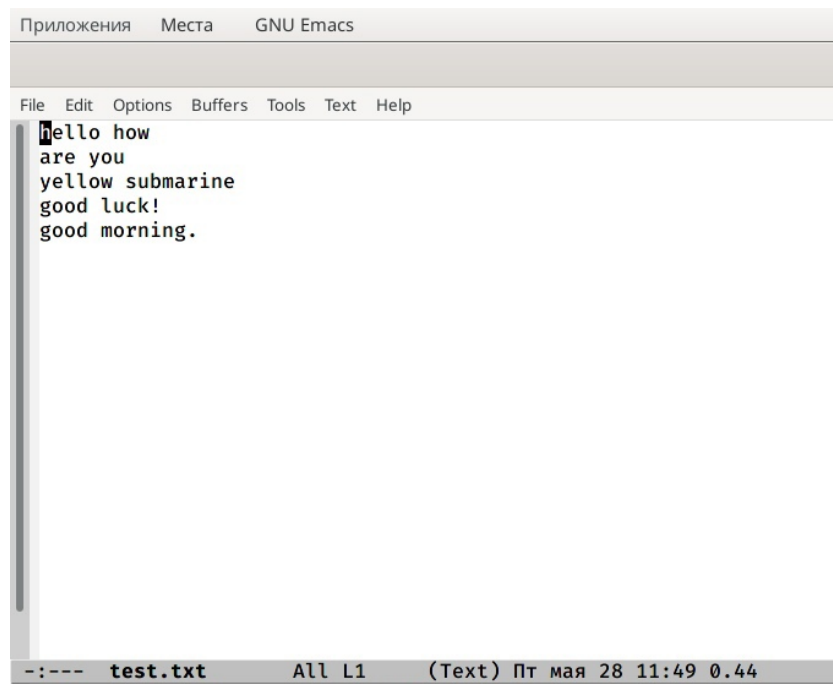


Рис. 3.2: Текстовый файл

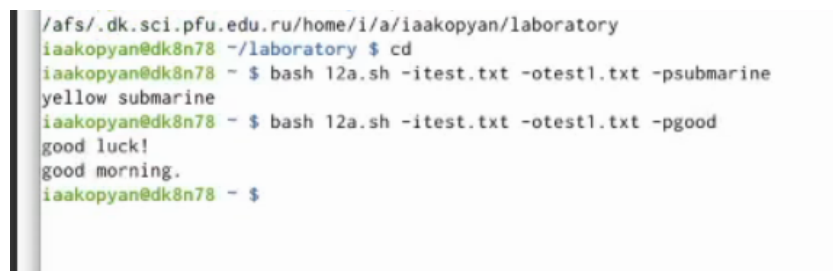


Рис. 3.3: Вызов

Написала на языке Си программу (рис. 5), 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 (рис. 4) вызывает эту программу и, проанализировав с помощью команды `$?`, выдал сообщение о том, какое число было введено. (рис. 6)

Написать программу на Си мне помог сайт. ссылка 2

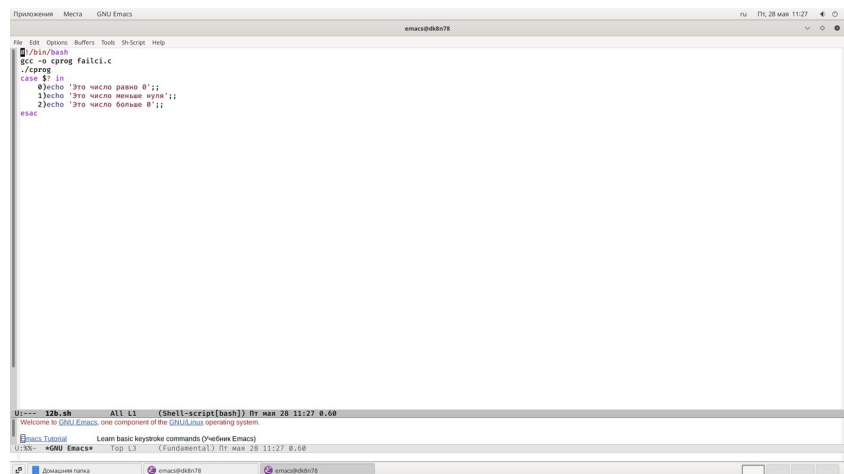


Рис. 3.4: Командный файл

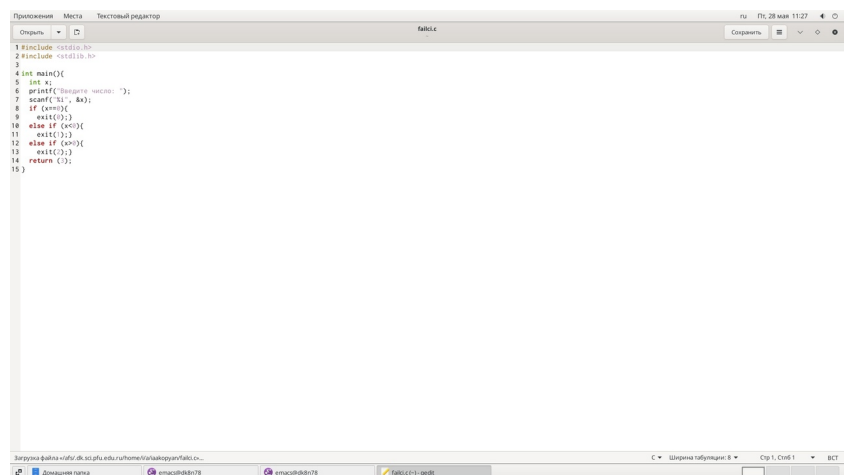
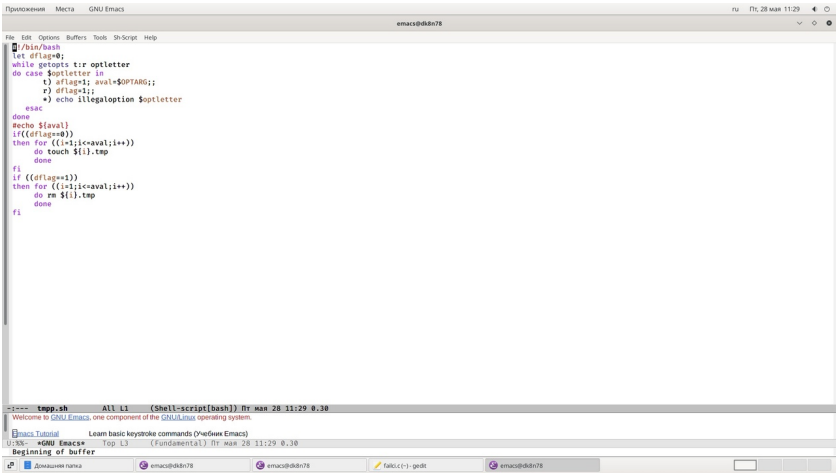


Рис. 3.5: Код на Си



Рис. 3.6: Вызов

Написала командный файл (рис. 7), 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). (рис. 8)



```
#!/bin/bash
let dflag=0
while getopts t:r optletter
do case $optletter in
  t) dflag=1; avail=$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
```

Рис. 3.7: Командный файл



```
iaakopyan@dk8n78 ~ $ bash tmpp.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 tmpp.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

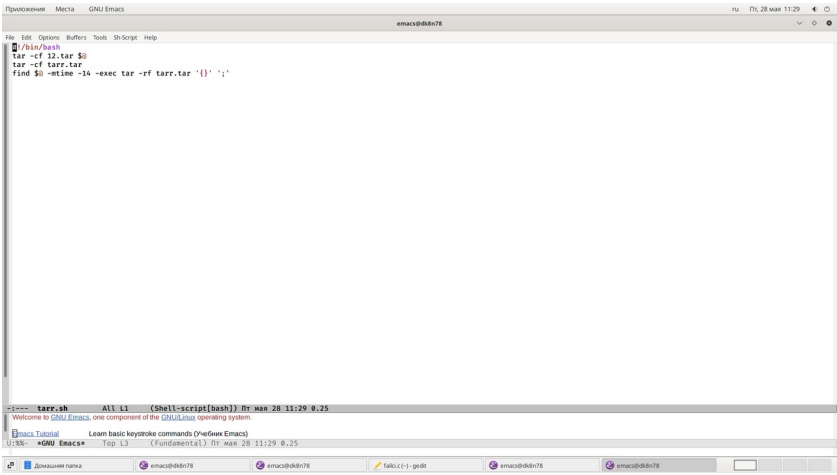
iaakopyan@dk8n78 ~ $ bash tmpp.sh -t3 -r
iaakopyan@dk8n78 ~ $ ls
0      12b.sh~  absd      addition.txt  asdfg.asm  failci.c
0.txt  321      absd.asm  asd           asdfg.o    failci.c~
1      abc1     absd.lst  asd~          conf.txt   file.txt
12b.sh abcv     absd.map  asdfg         cprog      GNUstep

iaakopyan@dk8n78 ~ $
```

Рис. 3.8: Вызов

Написала командный файл (рис. 9), который с помощью команды tar запаковыв-

вает 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). (рис. 10)



```
#!/bin/bash
tar -cf 12.tar $0
tar -cf tarr.tar
find $0 -mtime -14 -exec tar -cf tarr.tar '{}' ';'

tarr.sh
```

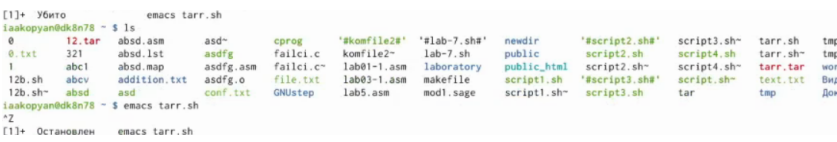
Рис. 3.9: Командный файл



```
12b.sh  abcv  addition.txt  asdfg.o  file.txt  lab03-1.asm  makef
12b.sh~  absd  asd  conf.txt  GNUstep  lab5.asm  mod1.
iaakopyan@dk8n78 ~ $ emacs tarr.sh
^Z
[1]+  Остановлен  emacs tarr.sh
iaakopyan@dk8n78 ~ $ bash tarr.sh лаб11
```

Рис. 3.10: Вызов

Результат (рис. 11) (рис. 12):



```
[1]+ Убито  emacs tarr.sh
iaakopyan@dk8n78 ~ $ ls
0.txt  12.tar  absd.asm  asd~  cprog  'komfile28'  '#lab-7.sha'  newdir  '#script2.sh#'  script3.sh~  tarr.sh  tmp
0.txt  321  absd.lst  asdfg  failci.c  lab01-1.asm  laboratory  public_html  script2.sh  script4.sh  tarr.tar  tmp
1  abcl  absd.map  asdfg.asm  failci.c~  lab01-1.asm  laboratory  public_html  script2.sh  script4.sh  tarr.tar  tmp
12b.sh  abcv  addition.txt  asdfg.o  file.txt  lab03-1.asm  makefile  script1.sh  '#script3.sh#'  script.sh~  text.txt  tmp  док
12b.sh~  absd  asd  conf.txt  GNUstep  lab5.asm  mod1.sage  script1.sh~  script3.sh  tar  tmp  док
iaakopyan@dk8n78 ~ $ emacs tarr.sh
^Z
[1]+ Остановлен  emacs tarr.sh
```

Рис. 3.11: ls

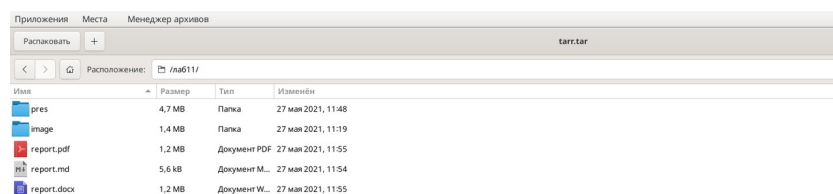


Рис. 3.12: Архив

4 Выводы

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

5 Библиография

ссылка 2 ссылка 1