Лабораторная работа № 12

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Маметкадыров Ынтымак

Цель работы

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

Выполнение лабораторной работы

1. Написали на языке Си программу (рис. 1), 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. 2) вызывает эту программу и, проанализировав с помощью команды $?, выдает сообщение о том, какое число было введено (рис. 3).

```c
#include <stdio.h>
#include <stdlib.h>

int main(){
  int n;
  printf("Введите число\n");
  scanf("%i", &n);
  if (n>0)
    {
      printf("Введенное число больше нуля\n", n);
      exit(n);
    }
  if (n<0)
    {
      printf("Введенное число меньше нуля\n");
      exit(n);
    }
  if (n==0)
    {
      printf("Введенное число равно нулю\n");
      exit(0);
    }
  return 0;
}
```

Рис. 1. Листинг программы compare.c

```
gcc compare.c
./a.out
let b=$?
if [ ${b} -gt 0 ]
then echo "Введенное число ${b}"
fi
if [ 0 -gt ${b} ]
then echo "Введенное число ${b}"
fi
if [ ${b} -eq 0 ]
then echo "Введенное число ${b}"
fi
```

Рис. 2. Листинг командного файла compare.sh

```
[itmametkadihrov@itmametkadihrov ~]$ ./compare.sh
Введите число
4
Введенное число больше нуля
Введенное число 4
[itmametkadihrov@itmametkadihrov ~]$
```

Рис. 3. Работа командного файла compare.sh

2. Написали командный файл files.sh (рис. 4), создающий указанное число файлов, пронумерованных последовательно от 1 до N. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Командный файл также умеет удалять все созданные им файлы, для этого нужно при запросе об удалении нажать 1, иначе 0. Передали в качестве параметра командному файлу число 5, в результате создались файлы 1.tmp, 2.tmp, 3.tmp, 4.tmp, 5.tmp (рис. 5). Попробовали удалить все созданные файлы (рис. 6).

```
n=$1
a=0
while [ ${a} -ne ${n} ]
do
(( a++ ))
touch ${a}.tmp
done
echo "Если хотите удалить созданные файлы нажмите 1, иначе 0"
read ans
if [ $ans -eq 1 ]
then b=0
while [ ${b} -ne ${n} ]
do
(( b++ ))
rm ${b}.tmp
done
fi
```

Рис. 4. Командный файл files.sh

```
[itmametkadihrov@itmametkadihrov ~]$ ./files.sh 5
Если хотите удалить созданные файлы нажмите 1, иначе 0
0
[itmametkadihrov@itmametkadihrov ~]$ ls
1.sh                                    compare.sh  pa
1.sh~                                   #count.sh#  #p
1.tmp                                   count.sh    pr
2.tmp                                   count.sh~   pr
3.tmp                                   files.sh    SC
4.tmp                                   files.sh~   sr
5.tmp                                   get.sh      ta
```

Рис. 5. Создание файлов

```
[itmametkadihrov@itmametkadihrov ~]$ ./files.sh 5
Если хотите удалить созданные файлы нажмите 1, иначе 0
1
[itmametkadihrov@itmametkadihrov ~]$ ls
1.sh                                    count.sh~       #print.sh#
1.sh~                                   files.sh        print.sh
academic-laboratory-report-template     files.sh~       print.sh~
academic-presentation-markdown-template get.sh          SCR
a.out                                   hello.sh        snap
a.txt                                   hello.sh~       tar.sh
backup                                  iftex.sty       tar.sh~
#backup.sh#                             lab08           work
backup.sh                               lab10.sh~       Видео
b.txt                                   lab.sh          Документы
cat.sh                                  lab.sh~         Загрузки
cat.sh~                                 ls.sh           Изображения
compare.c                               ls.sh~          Музыка
compare.sh                              out.tar         Общедоступные
#count.sh#                              pandoc-2.5      Рабочий стол
count.sh                                pandoc-crossref Шаблоны
```

Рис. 6. Удаление файлов

3. Написали tar -cvf out.tar work в командный файл tar.sh, который с помощью команды tar запаковывает в архив все файлы в указанной директории work (рис. 7).

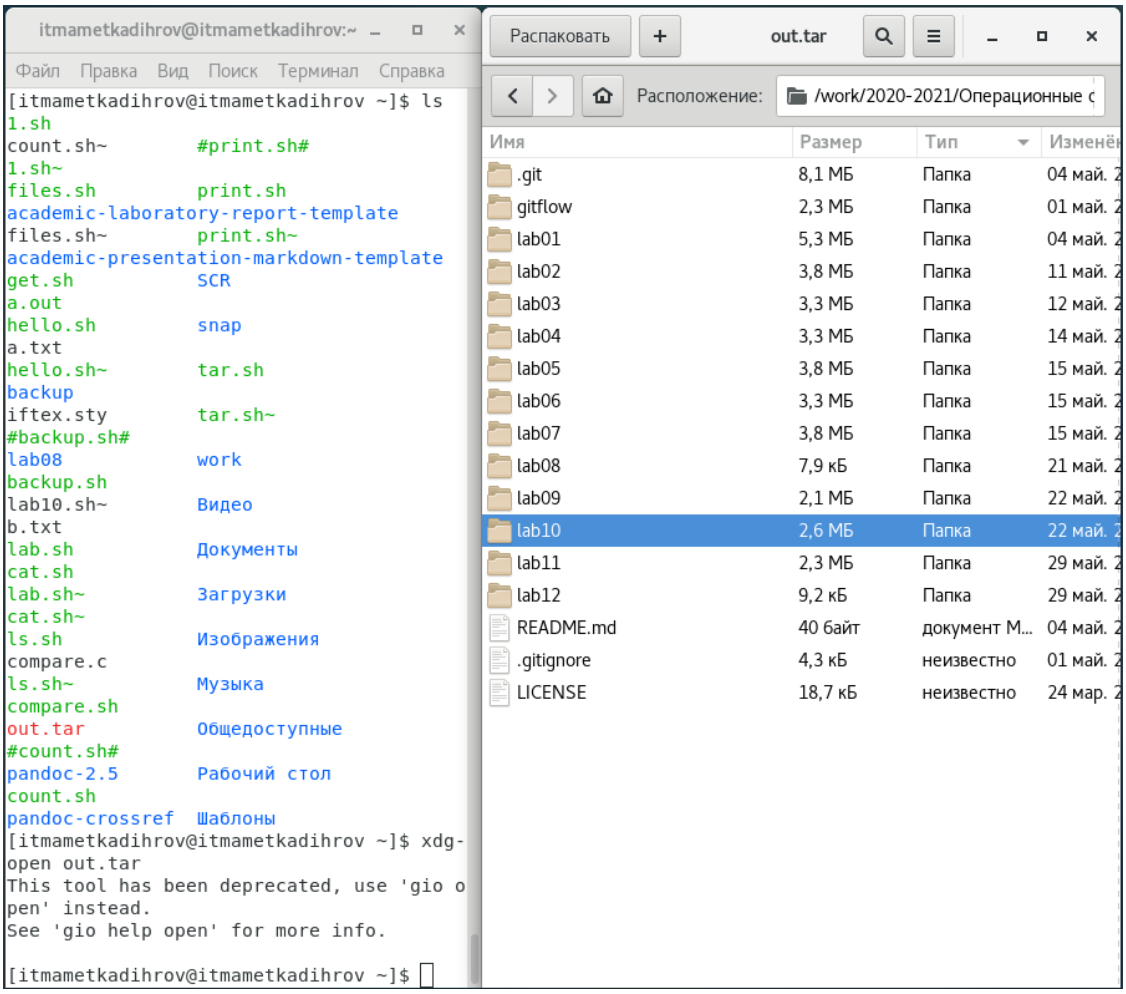

Рис. 7. Результат работы командного файла tar.sh

4. Модифицировали его так (рис. 8), чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (рис. 9).

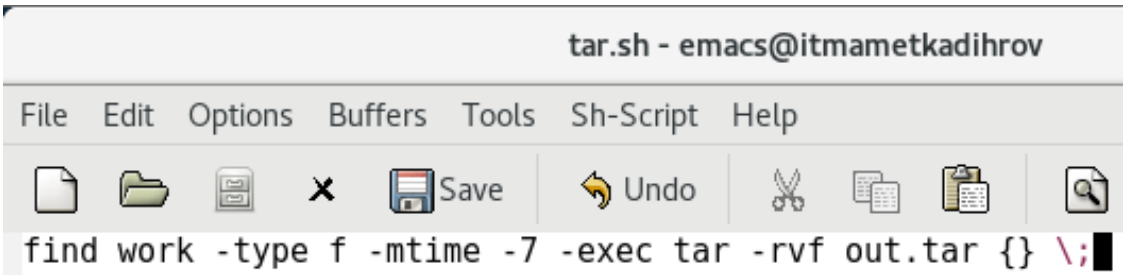

Рис. 8. Модифицированный командный файл tar.sh

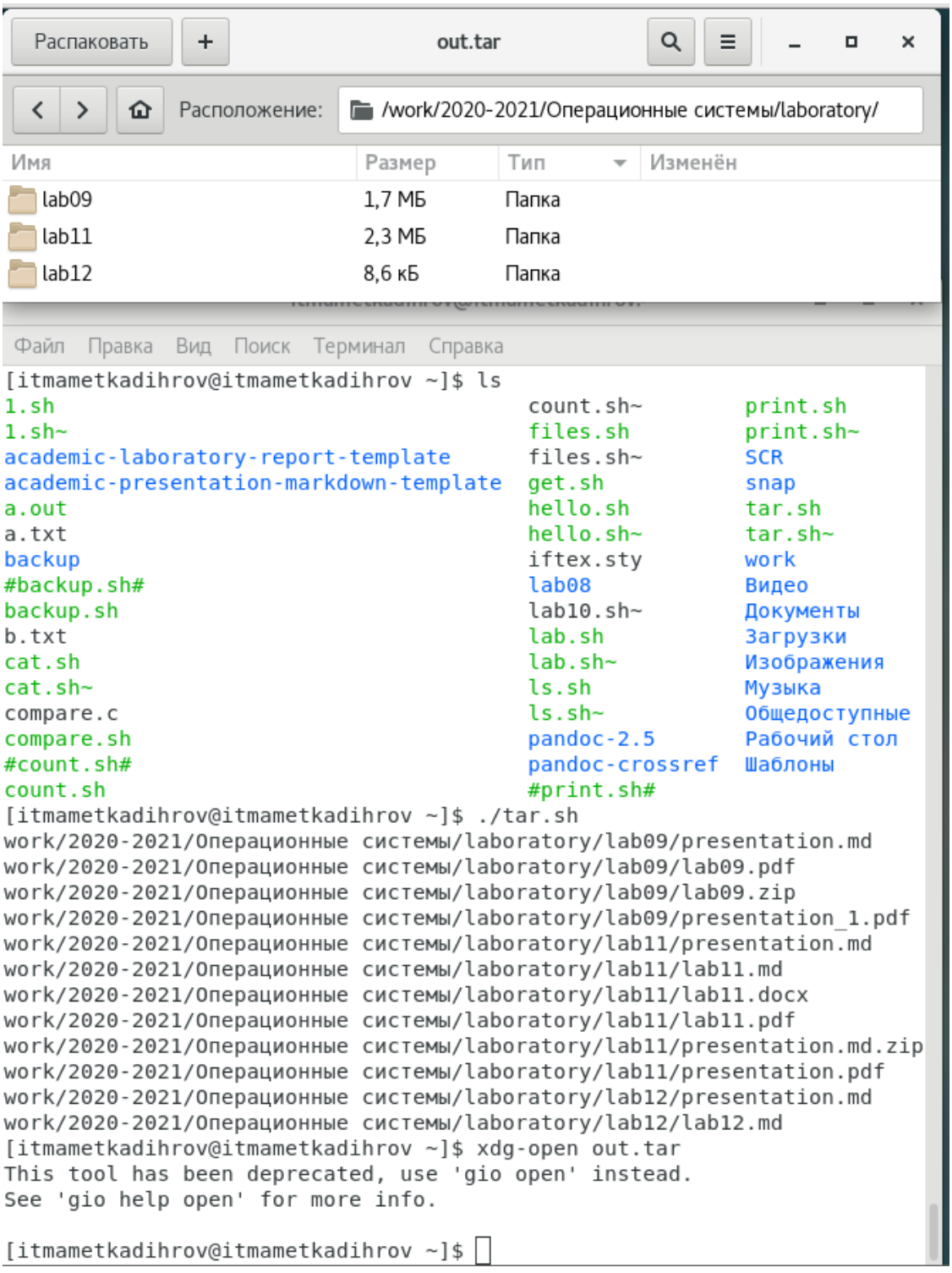

Рис. 9. Результат работы командного файла tar.sh

Вывод

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

Ответы на контрольные вопросы

1. getopts - осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных.

2. Такие символы, как ’ < > * ? | " &, являются метасимволами и имеют для командного процессора специальный смысл.

3. Управляющие операторы, как for, case, if и while.

4. Команды прерывания break и continue.

5. Используются для возвращения кода завершения. true всегда возвращает код завершения, равный нулю (т.е. истина), а команда false всегда возвращает код завершения, не равный нулю (т. е. ложь).

6. Возвращается истину, если существует файл.

7. При замене в операторе цикла while служебного слова while на until условие, при выполнении которого осуществляется выход из цикла, меняется на противоположное. В остальном оператор цикла while и оператор цикла until идентичны.