отчёта по лабораторной работе №12

Программирование в командномпроцессоре ОС UNIX. Ветвления и циклы

Джумаев Бегенч

Содержание

List of Tables

List of Figures

1 Цель работы

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

2 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, определяемые ключом-р.

2. Написать на языке Си программу, которая вводит число и определяет, являетсяли оно больше нуля, меньше нуля или равно нулю. Затем программа завершаетсяс помощью функцииexit(n), передавая информацию в о коде завершения воболочку. Командный файл должен вызывать эту программу и, проанализировавс помощью команды$?, выдать сообщение о том, какое число было введено.

3. Написать командный файл, создающий указанное число файлов, пронумерован-ных последовательно от1доN(например1.tmp,2.tmp,3.tmp,4.tmpи т.д.).Число файлов, которые необходимо создать, передаётся в аргументы командойстроки. Этот же командный файл должен уметь удалять все созданные им файлы(если они существуют).

4. Написать командный файл, который с помощью командыtarзапаковывает вархив все файлы в указанной директории. Модифицировать его так, чтобы запа-ковывались только те файлы, которые были изменены менее недели тому назад(использовать командуfind).

3 Выполнение лабораторной работы

1. Используя команды getopts grep, написал командный файл, который анализирует командную строку с ключами: -iinputfile— прочитать данные из указанного файла; -ooutputfile— вывести данные в указанный файл; -pшаблон— указать шаблон для поиска; -C— различать большие и малые буквы; -n— выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом-p.

2. Написал на языке Си программу, которая вводит число и определяет, являетсяли оно больше нуля, меньше нуля или равно нулю. Затем программа завершаетсяс помощью функцииexit(n), передавая информацию в о коде

завершения воболочку. Командный файл должен вызывать эту программу и, проанализировавс помощью команды $?, выдать сообщение о том, какое число было введено.

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

int main(){
    int a;
    printf("input: ");
    scanf("%i", &a);
    if (a==0) exit (0);
    else if (a<0) exit (1);
    else if (a>0) exit (2);
    return (3);
}
```

```
#!/bin/bash
gcc -o cprog lab92.c
./cprog
case $? in
    0)echo 'input number is equal to 0';;
    1)echo 'input number is smaller then 0';;
    2)echo 'input number is bigger then 0';;
esac
```

```
bdzhumaev@dk8n81 ~ $ emacs lab123.sh
bdzhumaev@dk8n81 ~ $ bash lab123.sh
input: 0
input number is equal to 0
bdzhumaev@dk8n81 ~ $ bash lab123.sh
input: 19
input number is bigger then 0
bdzhumaev@dk8n81 ~ $ bash lab123.sh
input: -7
input number is smaller then 0
```

3. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от1до N (например1.tmp,2.tmp,3.tmp,4.tmpи т.д.).Число файлов, которые необходимо создал, передаётся в аргументы командойстроки. Этот же командный файл должен уметь удалять все созданные им файлы

```
#!/bin/bash
let dflag=0;
while getopts a:d optletter
do case $optletter in
       a) aflag=1; aval=$OPTARG;;
       d) dflag=1;;
       *) echo Illegaloptтion $optletter
   esac
done
#echo ${aval}
if ((dflag==0))
then for ((i=1;i<=aval;i++))
     do touch ${i}.txt
     done
fi
if ((dflag==1))
then for ((i=1; i<=aval;i++))
     do rm ${i}.txt
     done
     fi
```

```
bdzhumaev@dk8n81 ~ $ touch lab124.sh
bdzhumaev@dk8n81 ~ $ chmod +x lab124.sh
bdzhumaev@dk8n81 ~ $ emacs lab124.sh
bdzhumaev@dk8n81 ~ $ bash lab124.sh -a4
lab124.sh: строка 5: синтаксическая ошибка рядом с неожиданным маркером «esac»
lab124.sh: строка 5: `   esac'
bdzhumaev@dk8n81 ~ $ emacs lab124.sh
bdzhumaev@dk8n81 ~ $ bash lab124.sh -a4
bdzhumaev@dk8n81 ~ $ ls -l
итого 86
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:17  1.txt
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:17  2.txt
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:17  3.txt
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:17  4.txt
drwxr-xr-x 4 bdzhumaev studsci  2048 мая 15 12:32  AA
-rw-rw-r-- 1 bdzhumaev studsci     0 мая 15 11:35  abc1
-rwxr-xr-x 1 bdzhumaev studsci   704 окт  3  2020  asdfg
-rw-r--r-- 1 bdzhumaev studsci   294 окт 22  2020  asdfg.asm
-rwxr-xr-x 1 bdzhumaev studsci  8072 мая 27 12:52  cprog
-rwxr-xr-x 1 bdzhumaev studsci   236 мая 27 10:27  file.sh
-rwxr-xr-x 1 bdzhumaev studsci     0 мая 27 10:15  file.sh~
drwxr-xr-x 3 bdzhumaev studsci  2048 апр 23 11:02  GNUstep
-rw-r--r-- 1 bdzhumaev studsci   306 мая 20 13:39 '#lab10.sh#'
-rw-r--r-- 1 bdzhumaev studsci    98 мая 20 12:22  lab10.sh
-rw-r--r-- 1 bdzhumaev studsci   326 мая 20 12:07  lab10.sh~
-rwxr-xr-x 1 bdzhumaev studsci   189 мая 27 12:40  lab122.c
-rwxr-xr-x 1 bdzhumaev studsci     0 мая 27 12:34  lab122.c~
-rwxr-xr-x 1 bdzhumaev studsci   191 мая 27 12:51  lab123.sh
-rwxr-xr-x 1 bdzhumaev studsci   190 мая 27 12:50  lab123.sh~
-rwxr-xr-x 1 bdzhumaev studsci   362 мая 27 13:16  lab124.sh
-rwxr-xr-x 1 bdzhumaev studsci   263 мая 27 13:07  lab124.sh~
-rw-r--r-- 1 bdzhumaev studsci   426 мая 27 12:34  lab12.sh
drwxr-xr-x 7 bdzhumaev studsci  2048 мая 27 09:11  laboratory
-rw-r--r-- 1 bdzhumaev studsci   427 окт 23  2020  main.asm
```

```
bdzhumaev@dk8n81 ~ $ emacs lab124.sh
bdzhumaev@dk8n81 ~ $ bash lab124.sh -a4
bdzhumaev@dk8n81 ~ $ ls -l
итого 86
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:29  1.txt
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:29  2.txt
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:29  3.txt
-rw-r--r-- 1 bdzhumaev studsci     0 мая 27 13:29  4.txt
drwxr-xr-x 4 bdzhumaev studsci  2048 мая 15 12:32  AA
-rw-rw-r-- 1 bdzhumaev studsci     0 мая 15 11:35  abc1
-rwxr-xr-x 1 bdzhumaev studsci   704 окт  3  2020  asdfg
-rw-r--r-- 1 bdzhumaev studsci   294 окт 22  2020  asdfg.asm
-rwxr-xr-x 1 bdzhumaev studsci  8072 мая 27 12:52  cprog
-rwxr-xr-x 1 bdzhumaev studsci   236 мая 27 10:27  file.sh
-rwxr-xr-x 1 bdzhumaev studsci     0 мая 27 10:15  file.sh~
drwxr-xr-x 3 bdzhumaev studsci  2048 апр 23 11:02  GNUstep
```

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

```
#!/bin/bash
tar -cf 9.tar $@
tar -cf 9l.tar
find $@ -mtime -7 -exec tar -rf 9l.tar '{}' ';'
```

```
bdzhumaev@dk8n81 ~ $ bash lab125.sh /lab12
tar: Удаляется начальный '/' из имен объектов
tar: /lab12: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: Завершение работы с состоянием неисправности из-за возникших ошибок
tar: Робкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
find: неизвестный предикат «-exe»
bdzhumaev@dk8n81 ~ $ ls
 12.tar      file.sh       lab122.c     lab125.sh    Makefile   program       reports       script.sh~   Загрузки
 AA          file.sh~      lab122.c~    lab125.sh~   may        program.asm   script2.sh    ski.plases   Изображения
 abc1        GNUstep       lab123.sh    lab12.sh     memos      program.lst   script2.sh~   tmp          Музыка
 asdfg      '#lab10.sh#'   lab123.sh~   laboratory   misk       public        script3.sh    work         Общедоступные
 asdfg.asm   lab10.sh      lab124.sh    main.asm     nasm.asm   public_html   script3.sh~   Видео       'Рабочий стол'
 cprog       lab10.sh~     lab124.sh~   main.lst     nasm.lst   README.md     script.sh     Документы    Шаблоны
bdzhumaev@dk8n81 ~ $
```

4 Выводы

Изучал основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

5 Контрольные вопросы

1. Команда getopts является встроенной командой командной оболочки bash, предназначенной для разбора параметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так и без них и этого вполне достаточно для передачи сценариям любых входных данных.

2. При генерации имен используют метасимволы:

- 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. Операторы && и || являются управляющими операторами. Если в командной строке стоит command1 && command2, то command2 выполняется в том, и только в том случае, если статус выхода из команды command1 равен нулю, что говорит об успешном ее завершении. Аналогично, если командная строка имеет вид command1 || command2, то команда command2 выполняется

тогда, и только тогда, когда статус выхода из команды command1 отличен от нуля.

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

5. Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха. Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи. Во всех управляющих конструкциях в качестве логического значения используется код возврата из программы, указанной в качестве условия. Код возврата 0 – истина, любое другое значение – ложь. Программа true – всегда завершается с кодом 0, false – всегда завершается с кодом 1.

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

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