Лабораторная работа №11.Программирование в командном процессоре ОС UNIX. Ветвления и циклы.

Операционые системы

Кочарян Никита Робертович

Содержание

1 Цель работы **5**

2 Задания **6**

3 Выполнение лабораторной работы **7**

4 Контрольные вопросы **10**

5 Ответы на контрольные вопросы **11**

6 Вывод **13**

Список иллюстраций

Список таблиц

1 Цель работы

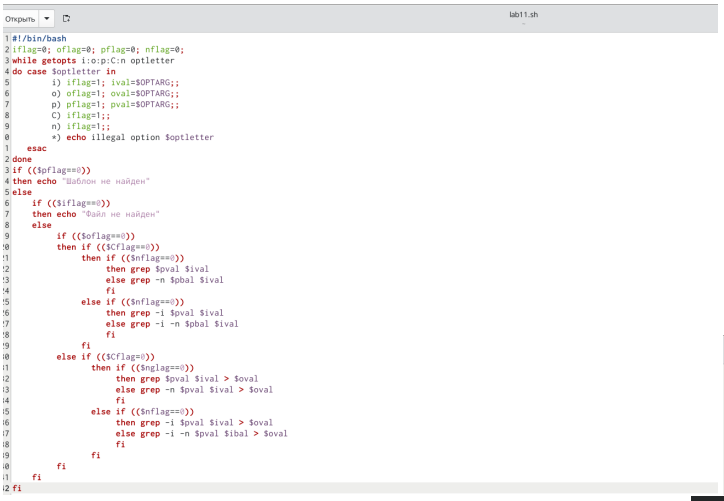
Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

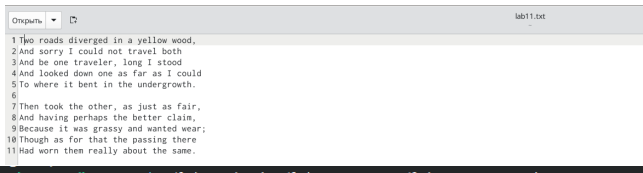
2 Задания

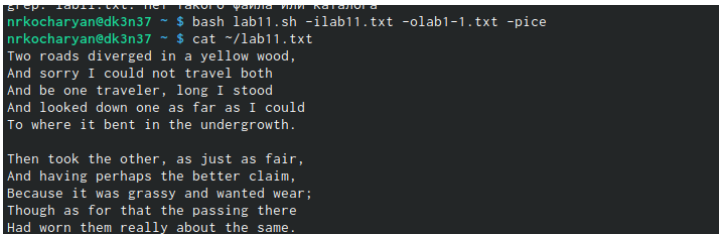
1. Используя команды getopts grep, написать командный файл, который анализирует командную строку с ключами: – -iinputfile — прочитать данные из указанного файла; – -ooutputfile — вывести данные в указанный файл; – -pшаблон — указать шаблон для поиска; – -C — различать большие и малые буквы; – -n — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом -p.
2. Написать 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 в о коде завершения в оболочку. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды $?, выдать сообщение о том, какое число было введено.
3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ⊠ (например 1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же ко- мандный файл должен уметь удалять все созданные им файлы (если они существуют).
4. Написать 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 (использовать команду find).

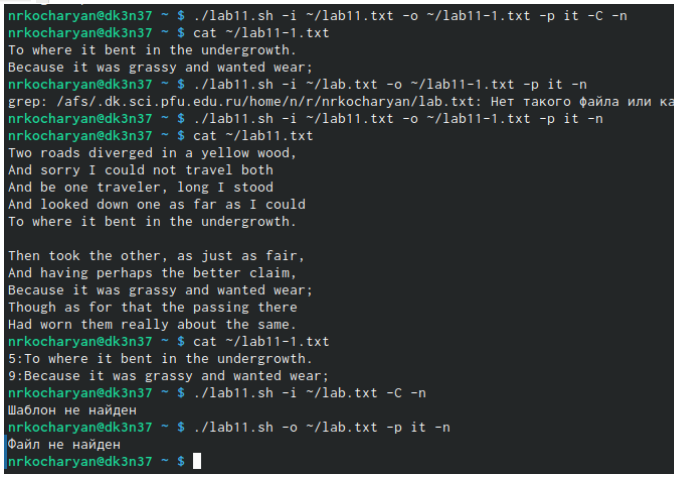
3 Выполнение лабораторной работы

1. Используя команды getopts grep, пишу 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.









2. Пишу 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 в о коде завершения в оболочку. Командный файл должен вызывать эту программу и, проанализировав с помощью команды $?,выдать сообщение о том, какое число было введено.

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до ☒ (например 1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

Открыть prog2.sh Сохранить

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

```
nrkocharyan@dk3n37 ~ $ chmod +x prog2.sh
nrkocharyan@dk3n37 ~ $ ./prog2.sh -c a#.txt 3
./prog2.sh: строка 7: синтаксическая ошибка рядом с неожиданным маркером «dot»
./prog2.sh: строка 7: ` for ((i=1; i<=num; i++)) dot'
nrkocharyan@dk3n37 ~ $ ./prog2.sh -c a#.txt 3
./prog2.sh: строка 9: [: отсутствует символ «]»
./prog2.sh: строка 9: [: отсутствует символ «]»
./prog2.sh: строка 9: [: отсутствует символ «]»
nrkocharyan@dk3n37 ~ $ ./prog2.sh -c a#.txt 3
./prog2.sh: строка 8: [: отсутствует символ «]»
./prog2.sh: строка 8: [: отсутствует символ «]»
./prog2.sh: строка 8: [: отсутствует символ «]»
nrkocharyan@dk3n37 ~ $ ./prog2.sh -c a#.txt 3
./prog2.sh: строка 9: [: отсутствует символ «]»
./prog2.sh: строка 9: [: отсутствует символ «]»
./prog2.sh: строка 9: [: отсутствует символ «]»
nrkocharyan@dk3n37 ~ $ ./prog2.sh -c a#.txt 3
./prog2.sh: строка 9: [: отсутствует символ «]»
./prog2.sh: строка 9: [: отсутствует символ «]»
./prog2.sh: строка 9: [: отсутствует символ «]»
nrkocharyan@dk3n37 ~ $ ./prog2.sh -c a#.txt 3
nrkocharyan@dk3n37 ~ $ ls
 backup      lab07.sh~     lab11.txt     prog.c        script2.sh~   tmp      Документы     Общедоступные
 bin         lab11-1.txt   lab11-1.txt   prog.sh       script3.sh~   TP       Загрузки      'Поляков c++'
 file.sh~    lab11.sh      prog          public        script4.sh~   work     Изображения   'Рабочий стол'
 GNUstep     lab1-1.txt    prog2.sh      public_html   script.sh~    Видео    Музыка        Шаблоны
nrkocharyan@dk3n37 ~ $ ./prog2.sh -r a#.txt3
./prog2.sh: строка 7: ((: i<=: синтаксическая ошибка: ожидается операнд (неверный маркер «<=»)
nrkocharyan@dk3n37 ~ $ ./prog2.sh -r a#.txt 2
nrkocharyan@dk3n37 ~ $ ls
 backup      lab07.sh~     lab11.txt     prog.c        script2.sh~   tmp      Документы     Общедоступные
 bin         lab11-1.txt   lab11-1.txt   prog.sh       script3.sh~   TP       Загрузки      'Поляков c++'
 file.sh~    lab11.sh      prog          public        script4.sh~   work     Изображения   'Рабочий стол'
 GNUstep     lab1-1.txt    prog2.sh      public_html   script.sh~    Видео    Музыка        Шаблоны
nrkocharyan@dk3n37 ~ $ ./prog2.sh -r a#.txt 3
nrkocharyan@dk3n37 ~ $ ls
 backup      lab07.sh~     lab11.txt     prog.c        script2.sh~   tmp      Документы     Общедоступные
 bin         lab11-1.txt   lab11-1.txt   prog.sh       script3.sh~   TP       Загрузки      'Поляков c++'
 file.sh~    lab11.sh      prog          public        script4.sh~   work     Изображения   'Рабочий стол'
 GNUstep     lab1-1.txt    prog2.sh      public_html   script.sh~    Видео    Музыка        Шаблоны
nrkocharyan@dk3n37 ~ $
```

4. Написать 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 (использовать команду find)

Открыть prog3.sh Сохранить

```
#!/bin/bash

files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```

```
nrkocharyan@dk3n37 ~ $ touch prog 3.sh
nrkocharyan@dk3n37 ~ $ gedit prog3.sh
nrkocharyan@dk3n37 ~ $ chmod +x prog3.sh
nrkocharyan@dk3n37 ~ $ ls -l
итого 61
-rw-r--r-- 1 nrkocharyan studsci     0 апр 20 14:38  3.sh
drwxr-xr-x 2 nrkocharyan studsci  2048 апр 13 13:33  backup
drwxr-xr-x 2 nrkocharyan studsci  2048 мар 16 13:07  bin
-rwxr-xr-x 1 nrkocharyan studsci     0 апр 13 13:41  file.sh~
drwxr-xr-x 3 nrkocharyan studsci  2048 мар  2 11:41  GNUstep
-rw-r--r-- 1 nrkocharyan studsci     0 апр  6 14:28  lab07.sh~
-rw-r--r-- 1 nrkocharyan studsci    80 апр 20 14:14  lab11-1.txt
-rwxr-xr-x 1 nrkocharyan studsci  1210 апр 20 14:11  lab11.sh
-rw-r--r-- 1 nrkocharyan studsci     0 апр 20 14:11  lab1-1.txt
-rw-r--r-- 1 nrkocharyan studsci   369 апр 20 14:03  lab11.txt
-rw-r--r-- 1 nrkocharyan studsci     0 апр 20 14:11  labll-1.txt
-rwxr-xr-x 1 nrkocharyan studsci 15624 апр 20 14:38  prog
-rwxr-xr-x 1 nrkocharyan studsci   232 апр 20 14:38  prog2.sh
-rwxr-xr-x 1 nrkocharyan studsci   210 апр 20 14:42  prog3.sh
-rw-r--r-- 1 nrkocharyan studsci   189 апр 20 14:20  prog.c
-rwxr-xr-x 1 nrkocharyan studsci   181 апр 20 14:25  prog.sh
drwxr-xr-x 3 nrkocharyan root     2048 сен  2  2022  public
lrwxr-xr-x 1 nrkocharyan root       18 мар  3 00:05  public_html -> public/public_htm
-rwxr-xr-x 1 nrkocharyan studsci    44 апр 13 13:37  script2.sh~
-rwxr-xr-x 1 nrkocharyan studsci   234 апр 13 12:49  script3.sh~
-rwxr-xr-x 1 nrkocharyan studsci   144 апр 13 13:49  script4.sh~
-rwxr-xr-x 1 nrkocharyan studsci    83 апр 13 13:30  script.sh~
drwxr-xr-x 2 nrkocharyan studsci  2048 мар  2 11:40  tmp
drwxr-xr-x 2 nrkocharyan studsci  2048 апр 20 11:46  TP
drwxr-xr-x 6 nrkocharyan studsci  2048 мар 30 12:11  work
drwxr-xr-x 2 nrkocharyan studsci  2048 сен 15  2022  Видео
drwxr-xr-x 2 nrkocharyan studsci  2048 сен 15  2022  Документы
drwxr-xr-x 2 nrkocharyan studsci  2048 апр 20 10:40  Загрузки
drwxr-xr-x 3 nrkocharyan studsci  2048 сен 19  2022  Изображения
drwxr-xr-x 2 nrkocharyan studsci  2048 сен 15  2022  Музыка
drwxr-xr-x 2 nrkocharyan studsci  2048 сен 15  2022  Общедоступные
drwxr-xr-x 3 nrkocharyan studsci  2048 мар 23 10:47 'Поляков c++'
drwxr-xr-x 2 nrkocharyan studsci  2048 мар 16 12:34 'Рабочий стол'
drwxr-xr-x 2 nrkocharyan studsci  2048 сен 15  2022  Шаблоны
nrkocharyan@dk3n37 ~ $ sudo ~/work/prog3.sh

Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

    №1) Уважайте частную жизнь других.
    №2) Думайте, прежде что-то вводить.
    №3) С большой властью приходит большая ответственность.

Пароль:
Попробуйте ещё раз.
Пароль:
Попробуйте ещё раз.
```

4 Контрольные вопросы

1. Каково предназначение команды getopts?
2. Какое отношение метасимволы имеют к генерации имён файлов?
3. Какие операторы управления действиями вы знаете?
4. Какие операторы используются для прерывания цикла?
5. Для чего нужны команды false и true?
6. Что означает строка if test -f man$s$/i.$s, встреченная в командном файле?
7. Объясните различия между конструкциями while и until

5 Ответы на 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.

6 Вывод

В ходе выполнения данной лабораторной работы я научился программировать в командом процессоре ОС UNIX, узнал что такое ветвления и циклы.