Лабораторная работа 11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Ерёмин Даниил

Содержание

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 Выполнение лабораторной работы

1) Используя команды гетопс и грепс я написал первый командный файл,который анализирует командную строку с несколькими ключами,а затем в указанном файле ищет нужные строки,определяемые также ключом и выводит их в указазанный файл, после этого добавляю права на выполнение файла и выполняю его,указав необходимые опции и аргументы (рис. -2.5)

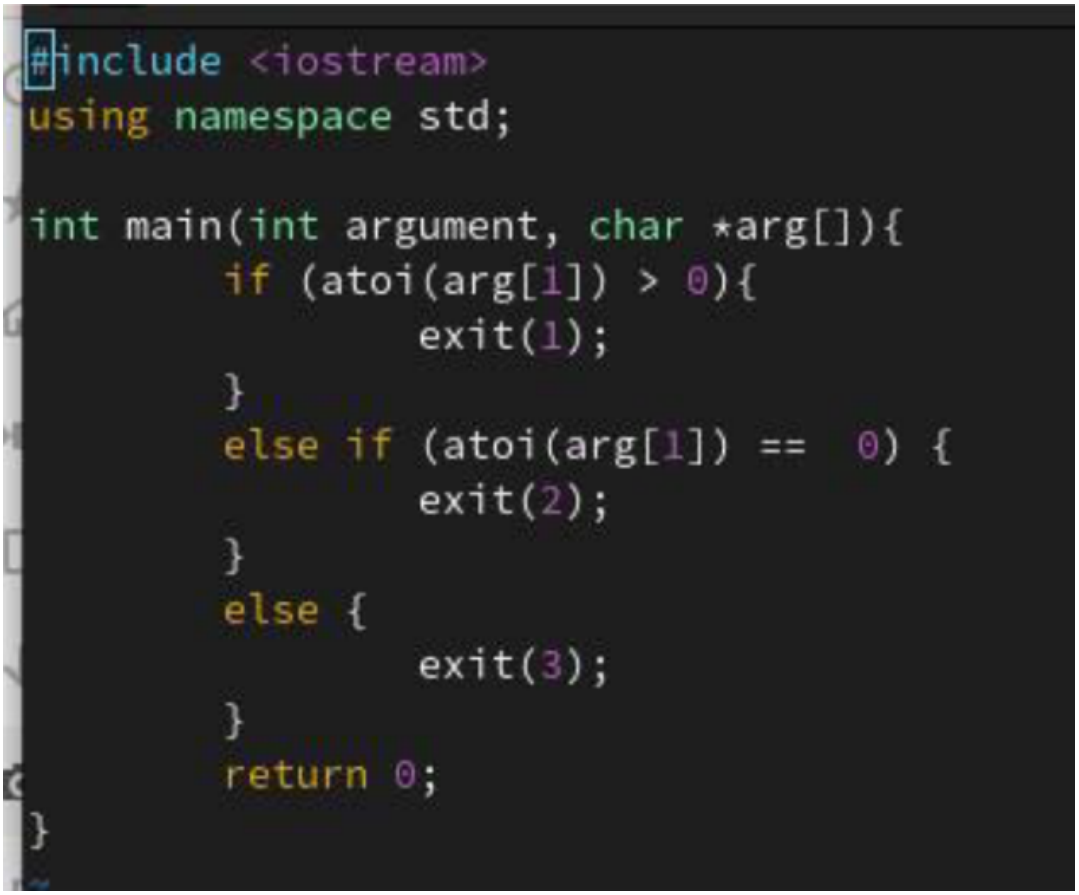
```
while getopts "i:o:p:c:n" opt
do
case $opt in
    i)inputfile="$OPTARG";;
    o)outputfile="$OPTARG";;
    p)shablon="OPTARG";;
    c)registr="";;
    n)number="":;;
esac
done

grep -n "$shablon" "$inputfile" > "$outputfile"

~
~
```

Рис. 2.1: первый командный файл

2) На языке программирования C## я написал вспомогательную программу,которая вводит число и определяет его относительно нуля. (рис. -2.5)



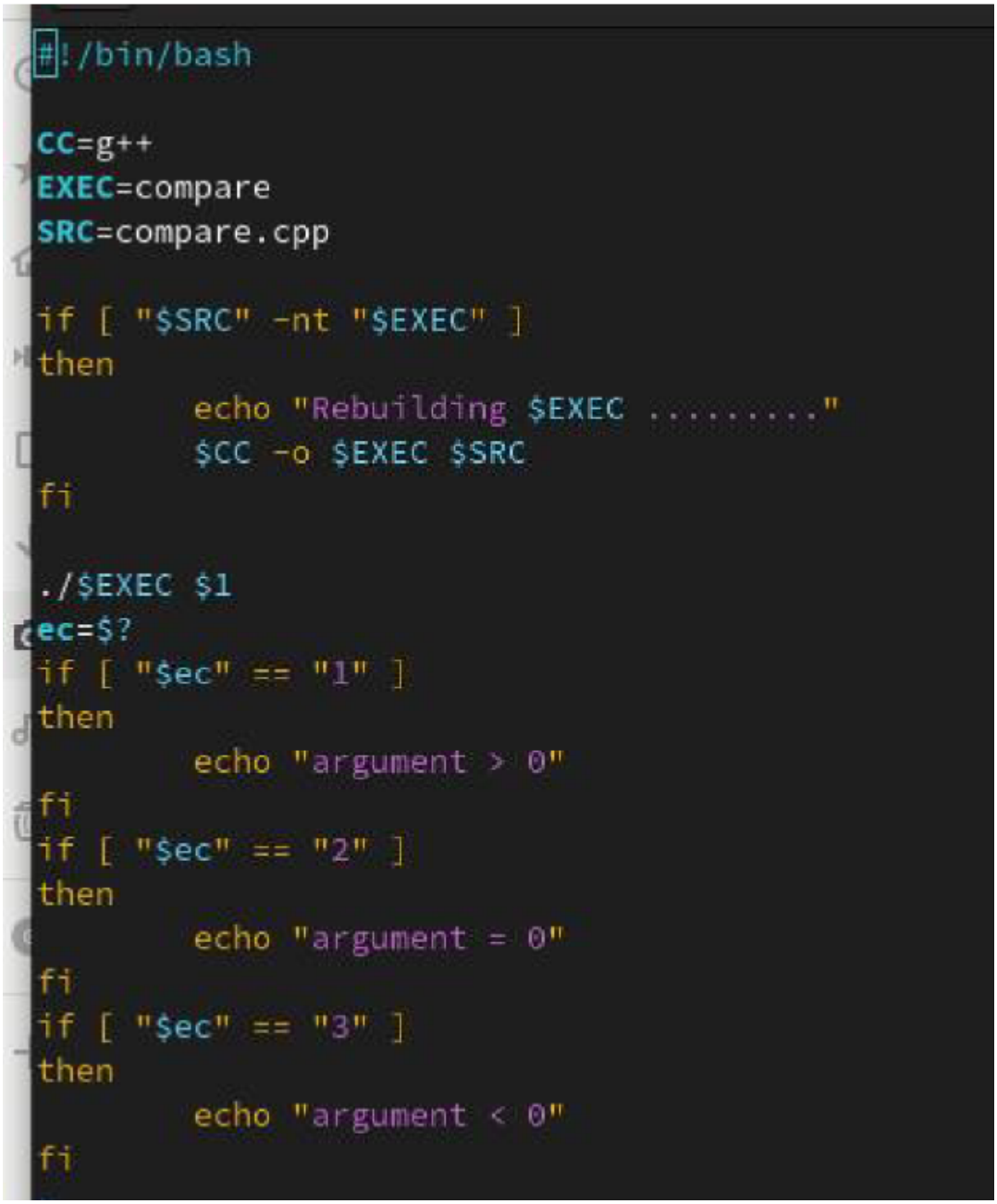
```
#include <iostream>
using namespace std;

int main(int argument, char *arg[]){
        if (atoi(arg[1]) > 0){
                exit(1);
        }
        else if (atoi(arg[1]) ==  0) {
                exit(2);
        }
        else {
                exit(3);
        }
        return 0;
}
```

Рис. 2.2: вторая программа

3) Написал командный файл,который вызывает эту программу и выдает сообщение о том,какое число было введено, после чего я добавил права на выполнение файла и выполнил его,указав необходимые опции и аргументы.

(рис. -2.5)



```
#!/bin/bash

CC=g++
EXEC=compare
SRC=compare.cpp

if [ "$SRC" -nt "$EXEC" ]
then
        echo "Rebuilding $EXEC ........."
        $CC -o $EXEC $SRC
fi

./$EXEC $1
ec=$?
if [ "$ec" == "1" ]
then
        echo "argument > 0"
fi
if [ "$ec" == "2" ]
then
        echo "argument = 0"
fi
if [ "$ec" == "3" ]
then
        echo "argument < 0"
fi
```

Рис. 2.3: третья программа

4) Создал командный файл, кооторый создает n файлов последовательно пронумерованных, где n задается как аргумент командной строки, после чего я добавил права на выполнение файла и выполнил его,указав необходимые опции и аргументы. (рис. -2.5)

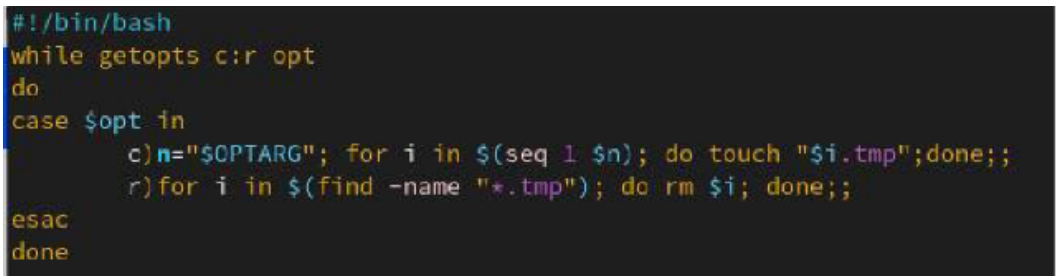


Рис. 2.4: четвертая программа

5) Создал командный файл,который с помощью команды tar запаковывает в архив все файлы в указанной директории,модифицировал его так,чтобы он запаковывал только те файлы,которые изменялись менее недели тому назад, после чего я добавил права на выполнение файла и выполнил его,указав необходимые опции и аргументы.

(рис. -2.5)

Рис. 2.5: пятая программа

Контрольные вопросы: Каково предназначение команды getopts?

Команда getopts осуществляет синтаксический анализ командной строки, выделяя флаги, и используется для объявления переменных. Синтаксис команды следующий: getopts option-string variable [arg ...]

Флаги – это опции командной строки, обычно помеченные знаком минус; Например, -F является флагом для команды ls -F. Иногда эти флаги имеют аргумен-

ты, связанные с ними. Программы интерпретируют эти флаги, соответствующим образом изменяя свое поведение. Строка опций option-string — это список возможных букв и чисел соответствующего флага. Если ожидается, что некоторый флаг будет сопровождаться некоторым аргументом, то за этой буквой должно следовать двоеточие. Соответствующей переменной присваивается буква данной опции. Если команда getopts может распознать аргумент, она возвращает истину. Принято включать getopts в цикл while и анализировать введенные данные с помощью оператора case.

Какое отношение метасимволы имеют к генерации имён файлов?

При перечислении имен файлов текущего каталога можно использовать следующие символы:

соответствует произвольной, в том числе и пустой строке;

? — соответствует любому одному символу;

c1-c1 — соответствует любому символу, лексикографически на ходящемуся между симво

echo * — выведет имена всех файлов текущего каталога, что представляет собой прос

ls .c — выведет все файлы с последними двумя символами, равными .c;

echo prog.? — выдаст все файлы, состоящие из пяти или шести символов, первыми пят

a-z — соответствует произвольному имени файла в текущем каталоге, начинающемуся с

Какие операторы управления действиями вы знаете?

Точка с запятой (;) Вы можете разместить две и более команд в одной и той же стро

Амперсанд (&) В том случае, если строка команды оканчивается символом амперсанда

Символ доллара со знаком вопроса (Кодзавершенияпредыдущейкомандысохраняетсявперем

Двойной амперсанд (&&) Командная оболочка будет интерпретировать последовательнос

Двойная вертикальная черта (||) Оператор || представляет логическую операцию "ИЛИ

Комбинирование операторов && и || Вы можете использовать описанные логические опе

Знак фунта (#) Все написанное после символа фунта (#) игнорируется командной обол

Экранирование специальных символов () Символ обратного слэша \ позволяет использо

Какие операторы используются для прерывания цикла?

Для управления ходом выполнения цикла служат команды break и continue [1] и точно соответствуют своим аналогам в других языках программирования. Команда break прерывает исполнение цикла, в то время как continue передает управление в начало цикло, минуя все последующие команды в теле цикла.

Для чего нужны команды false и true?

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.

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

Веденная строка означает условие существования файла man$s/$i.$s

`Объясните различия между конструкциями while и until.`

Разница между циклом while (пока) и until (пока не) – это условие проверки. Пока ВЫПОЛНЯЕТСЯ условие проверки, цикл while будет продолжать работать. Однако цикл until будет выполняться только пока условие ЛОЖНО.

3 Выводы

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