Лабораторная работа №10. Программирование в командном процессоре ОС UNIX. Командные файлы.

Matiukhin Grigorii

Цель работы:

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

Используя команды `getopts`, `grep`, написать командный файл, который анализирует командную строку с ключами:

- `-i` — прочитать данные из указанного файла
- `-o` — вывести данные в указанный файл
- `-p` - шаблон — 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`

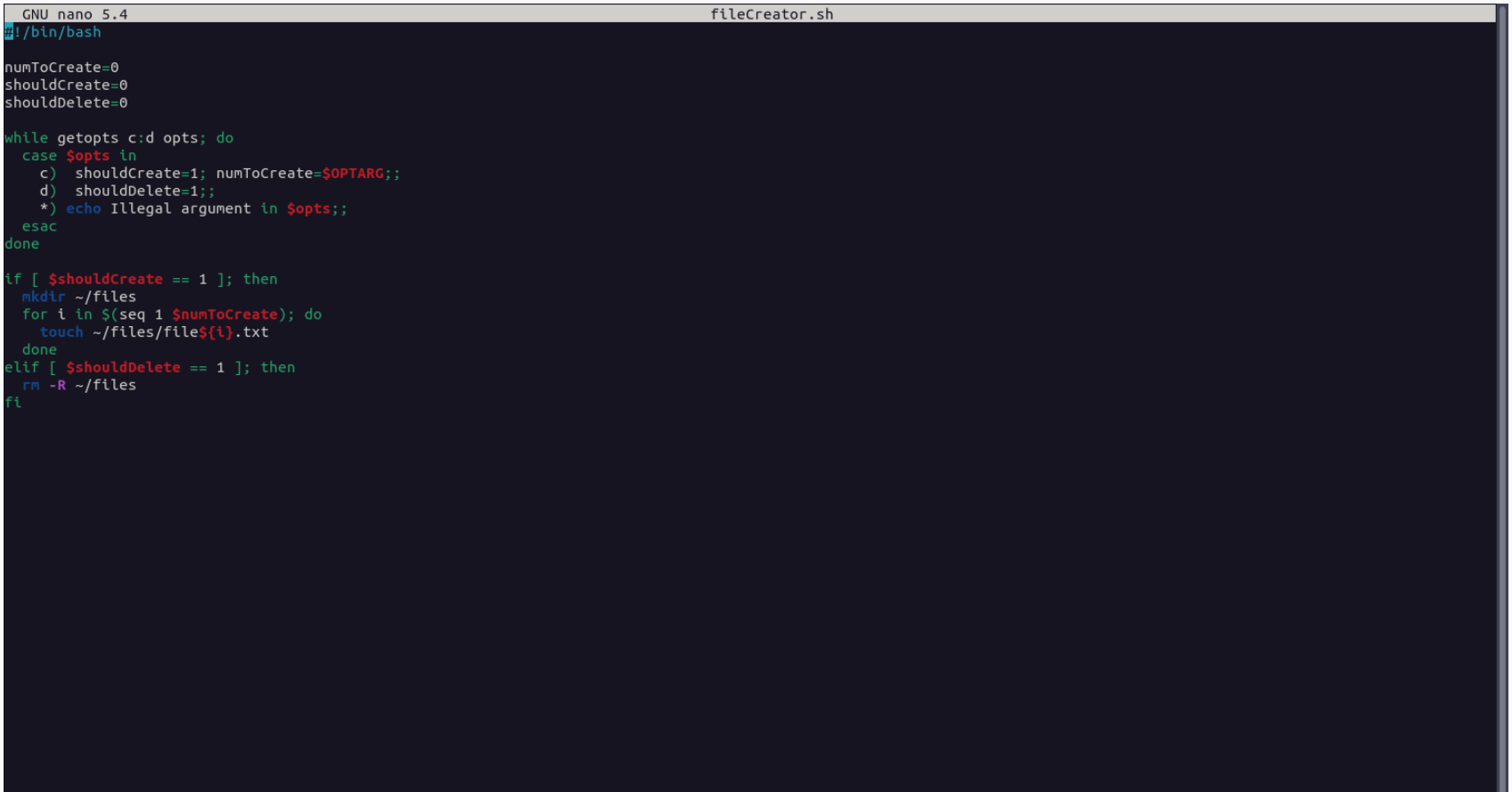
```
 GNU nano 5.4                                  fileAnalyzer.sh *
#!/bin/bash

inputFile=""
outputFile=""
pattern=""
isCaseSensitive=0
printLineNumbers=0

while getopts i:o:p:Cn optletters; do
  case $optletters in
    i)  inputFile="$OPTARG";;
    o)  outputFile="$OPTARG";;
    p)  pattern="$OPTARG";;
    C)  isCaseSensitive=1;;
    n)  printLineNumbers=1;;
    *) echo Illegal option $optletters; exit;;
  esac
done

if [ -z "$inputFile" ] || [ -z "$outputFile" ] || [ -z "$pattern" ]; then
  echo inputFile, outputFile and pattern are mandatory arguments
  exit
fi

if [ $isCaseSensitive == 1 ] && [ $printLineNumbers == 1 ]; then
  grep $pattern -n $inputFile > $outputFile
elif [ $isCaseSensitive == 1 ] && [ $printLineNumbers == 0 ]; then
  grep $pattern $inputFile > $outputFile
elif [ $isCaseSensitive == 0 ] && [ $printLineNumbers == 1 ]; then
  grep $pattern -i -n $inputFile > $outputFile
else
  grep $pattern -i $inputFile > $outputFile
fi
```

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

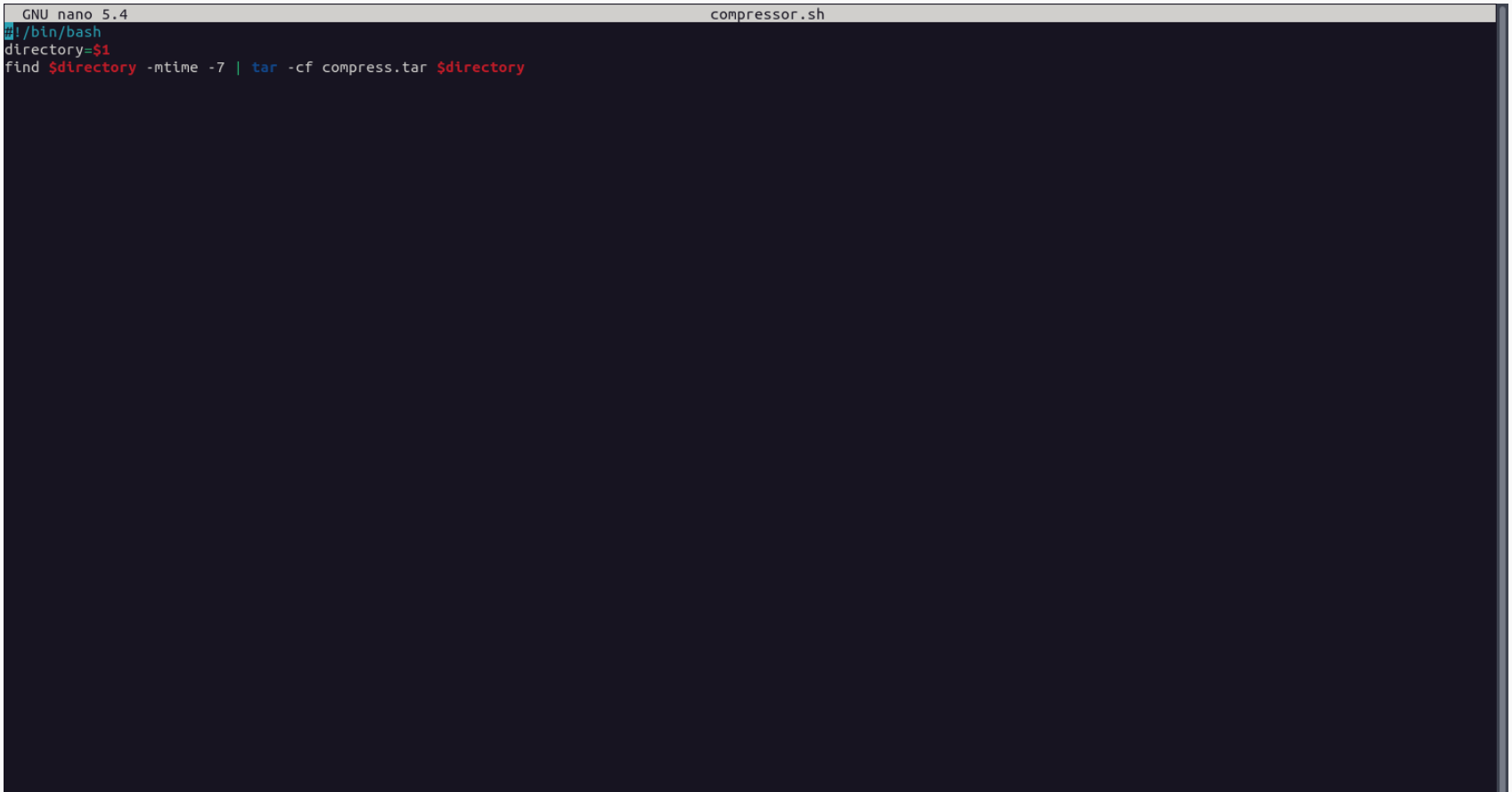
GNU nano 5.4 exit.cpp

```cpp
#include <iostream>

int main() {
  int n;
  std::cin >> n;

  if (n > 0) {
    exit(2);
  } else if (n == 0) {
    exit(1);
  } else {
    exit(0);
  }

  return 0;
}
```

Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (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)



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`)



Вывод

Вывод

В ходе работы я изучил основы программирования в оболочке ОС UNIX. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.