

Лабораторная работа №12

Журавлев Георгий Иванович

Цель работы

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

Ход работы.

1. Используя команды `getopts` `grep`, написал командный файл, который анализирует командную строку с ключами:

- `-i` — прочитать данные из указанного файла;
- `-o` — вывести данные в указанный файл;
- `-r` - 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 `-r`.

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
while getopts i:o:p:Cn letter
do case $letter in
  i) i=1; larg=$OPTARG;;
  o) o=1; oarg=$OPTARG;;
  p) p=1; parg=$OPTARG;;
  C) C=1;;
  n) n=1;;
  *) echo wrongoption $letter
  esac
done

if(((C==1)&&(n==1)))
then grep -e${parg} -i -n ${larg}
  if((o==1))
  then grep -e${parg} -i -n ${larg} > ${oarg}
  fi
fi

if(((C==1)&&(n==0)))
then grep -e${parg} -i -n ${larg}
  if((o==1))
  then grep -e${parg} -i ${larg} > ${oarg}
  fi
fi

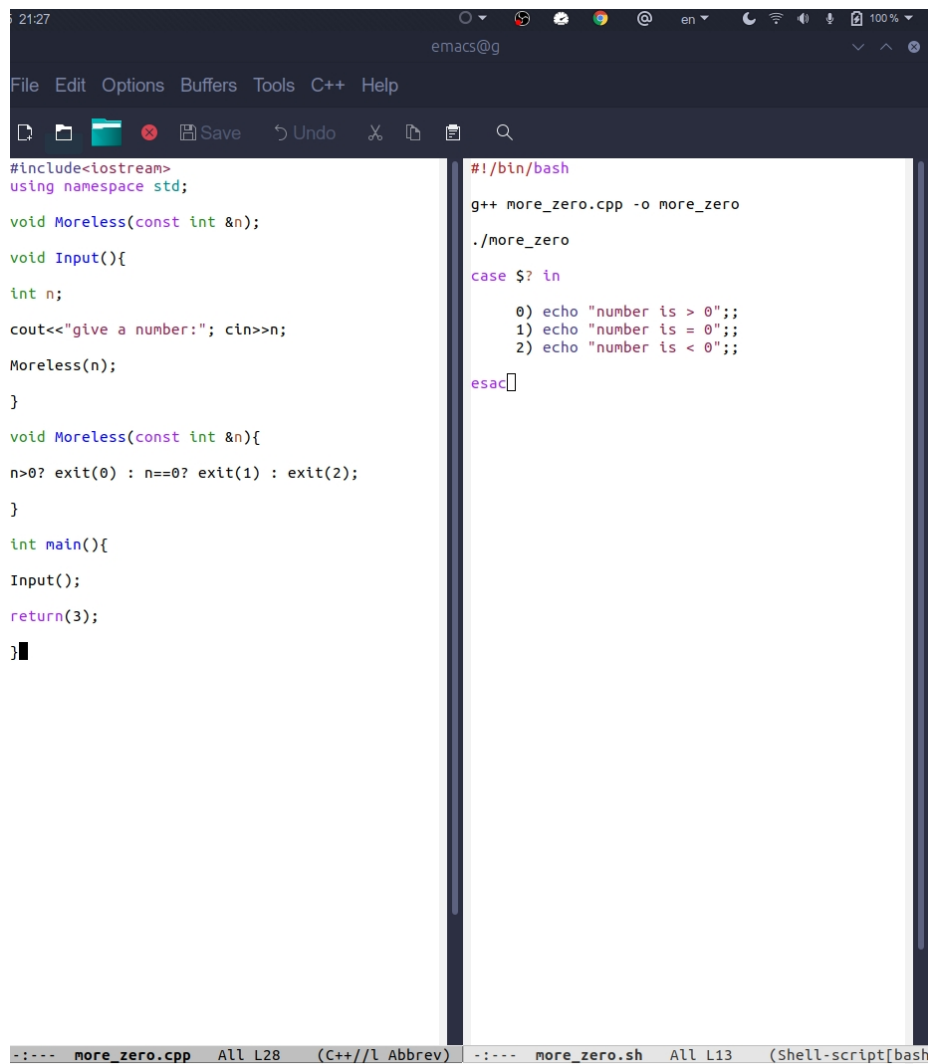
if(((C==0)&&(n==1)))
then grep -e${parg} -i -n ${larg}
  if((o==1))
  then grep -e${parg} -n ${larg} > ${oarg}
  fi
fi

if(((C==0)&&(n==0)))
then grep -e${parg} -i -n ${larg}
  if((o==1))
  then grep -e${parg} ${larg} > ${oarg}
  fi
fi

fi

-UU-:---F1 lab12.01.sh All L1 (Shell-script[bash]) -----
Beginning of buffer
```

Рис. 1: командный файл



The screenshot shows the Emacs editor interface with two files open. The left pane displays the C++ source code for `more_zero.cpp`, and the right pane displays the shell script for `more_zero.sh`.

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

void Moreless(const int &n);

void Input(){
    int n;
    cout<<"give a number:"; cin>>n;
    Moreless(n);
}

void Moreless(const int &n){
    n>0? exit(0) : n==0? exit(1) : exit(2);
}

int main(){
    Input();
    return(3);
}
```

```
#!/bin/bash

g++ more_zero.cpp -o more_zero

./more_zero

case $? in
    0) echo "number is > 0";;
    1) echo "number is = 0";;
    2) echo "number is < 0";;
esac
```

The status bar at the bottom indicates the current file is `more_zero.sh` at line 13, and the shell is `(Shell-script[bash])`.

Рис. 2: командный файл



The screenshot shows a terminal window with the command `chmod +x more_zero.sh` being executed. The prompt is `g_zhuravlev@`. The status bar at the bottom shows the user is `g_zhuravlev`, the command is `chmod +x more_zero.sh`, and the time is `21:27:50`.

Рис. 3: изменение прав



```
g_zhuravlev@g ~$ ./more_zero.sh
give a number:-9
number is < 0
g_zhuravlev@g ~$ ./more_zero.sh
give a number:9
number is > 0
g_zhuravlev@g ~$ ./more_zero.sh
give a number:0
number is = 0
g_zhuravlev@g ~$
```

Рис. 4: результат работы командного файла

2. Написал на языке Сpp 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. Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до `i`. Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

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 (использовал команду `find`).

Вывод

Благодаря этой лабараторной работе, я написал некоторые интересные скрипты , которые оказались сложнее предыдущих; развился в сфере взаимодействия с `bash`.

Контрольные вопросы.

1. Команда `getopts` предназначена для разбора параметров сценариев. Она обрабатывает исключительно
2. Метасимволы используются для генерации имен:
 - 2.1. `?` произвольный символ;
 - 2.2. `*` произвольная последовательность символов;
 - 2.3. `[...]` любой из символов, указанных в скобках перечислением и/или с указанием диапазона;
3. Операторы: `"`; `"` ; `&&(и)` ; `||`(или); `$`. - операторы управления действиями.
4. Оператор `break` используется для прерывания цикла.

```
#!/bin/bash
let delete=0;
while getopts c:d letter
do case $letter in
    c)create=1; arg=$OPTARG;;
    d)delete=1;;
    *) echo WrongOption $letter
    esac
done

if((delete==0))
then for((i=1;i<=arg;i++))
do touch ${i}.doc
echo document №${i} has been created
done
fi
if ((delete==1))
then for((i=1;i<=arg;i++))
do rm ${i}.doc
echo document №${i} has been deleted
done
fi
```

```
U:--- lab12.03.sh All L20 (Shell-script[sh])
Wrote /home/q zhuravlev/lab12.03.sh
```

Рис. 5: командный файл

```
document #3 has been created
g_zhuravlev@g ~$ ./lab12.03.sh -c3
document #1 has been created
document #2 has been created
document #3 has been created
g_zhuravlev@g ~$ ls -l
total 224
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev  0 мая 5 21:51 1.doc
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev  0 мая 5 21:51 2.doc
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev  0 мая 5 21:51 3.doc
```

Рис. 6: результат работы командного файла

```
g_zhuravlev@g ~$ ./lab12.03.sh -c3 -d
document #1 has been deleted
document #2 has been deleted
document #3 has been deleted
g_zhuravlev@g ~$ ls -l
total 228
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev 3424 мая 5 20:35 50426c36-62f3-4a41-9440-ad5cb42280c9.o
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev  0 мая 2 13:59 abc1
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev  0 мая 2 14:10 abc2
-rw-rw-r-- 1 g_zhuravlev g_zhuravlev 4086 мая 2 08:34 abc3
```

Рис. 7: результат работы командного файла

5. Команда true всегда возвращает. Команда false всегда возвращает не ноль. Программа true – всегда за
6. Эта строка означает условие существования файла man\$\$/\$i.\$\$
7. Цикл While выполняется до тех пор, пока указанное в нем условие истинно. Цикл Until выполняется до

```
#!/bin/bash
directory=""
echo welcome to tar compressor
echo which directory i need to compress?
read directory
#tar -cf compressed.tar $directory
find $directory -mtime -7 | tar -cf compressed.tar $directory
```

```
U:--- lab12.04.sh All L4 (Shell-script[sh])
Input method: 
```

Рис. 8: командный файл

```
g_zhuravlev@g ~$ ./lab12.04.sh
welcome to tar compressor
which directory do i need to compress?
python-projects/
```

Рис. 9: результат работы командного файла

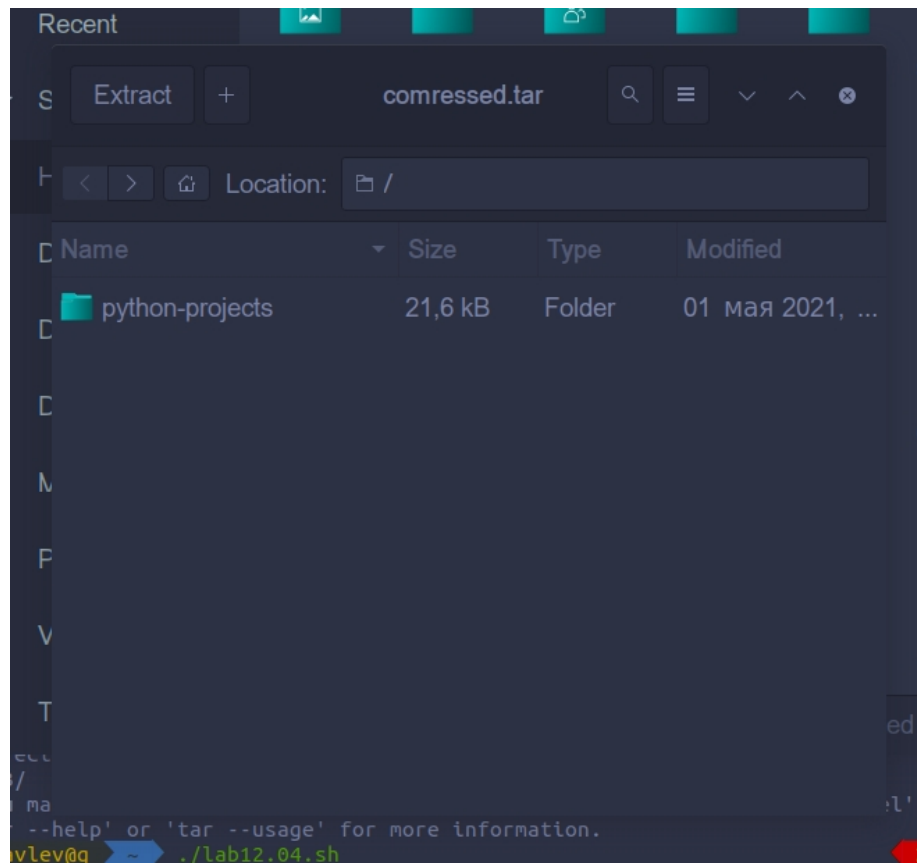


Рис. 10: результат работы командного файла