РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

дисциплина:Операционные системы

Студент: Мартемьянов Александр

Группа: НПМбв-02-18

МОСКВА

2022 г.

2)Задание

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

3) Последовательность выполнения работы

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.

```
#!/bin/bash
while getopts i:o:p:Cn letters
do case $letters in
       i) i=1; iarg=SOPTARG;;
       o) o=1; oarg=$OPTARG;;
       P) p=1; parg=SOPTARG; ;
       C) C=13;
       n) n=133
       *) echo wrongoption $letters
    esac
done
          if(((C==1)&&(n==1)))
          then grep -eS${parg} -i -n ${iarg}
               if((o==1))
               then grep -e${parg} -i -n ${iarg} > ${oar
               fi
fi
               if(((c==1)aa(n==0)))
               then grep -e${parg} -i -n ${iarg}
               if((o==1))
               then grep -e${parg} -i ${iarg} > ${oarg}
               fi
fi
               if(((c==0)88(n==1)) ll
               then grep -eS${parg} -i -n ${iarg}
               if((o==1))
               then grep -e${parg} -n ${iarg} > ${oarg}
               fi
fi
                  if(((C==0)&&(n==0)))
                  then grep -eS{parg} -i -n ${iarg}
                  if((o==1))
                  then grep -e${parg} ${iarg} > ${oarg}
                  fi
fi
```

Рис 3. 1 «Анализ коммандной строки»

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

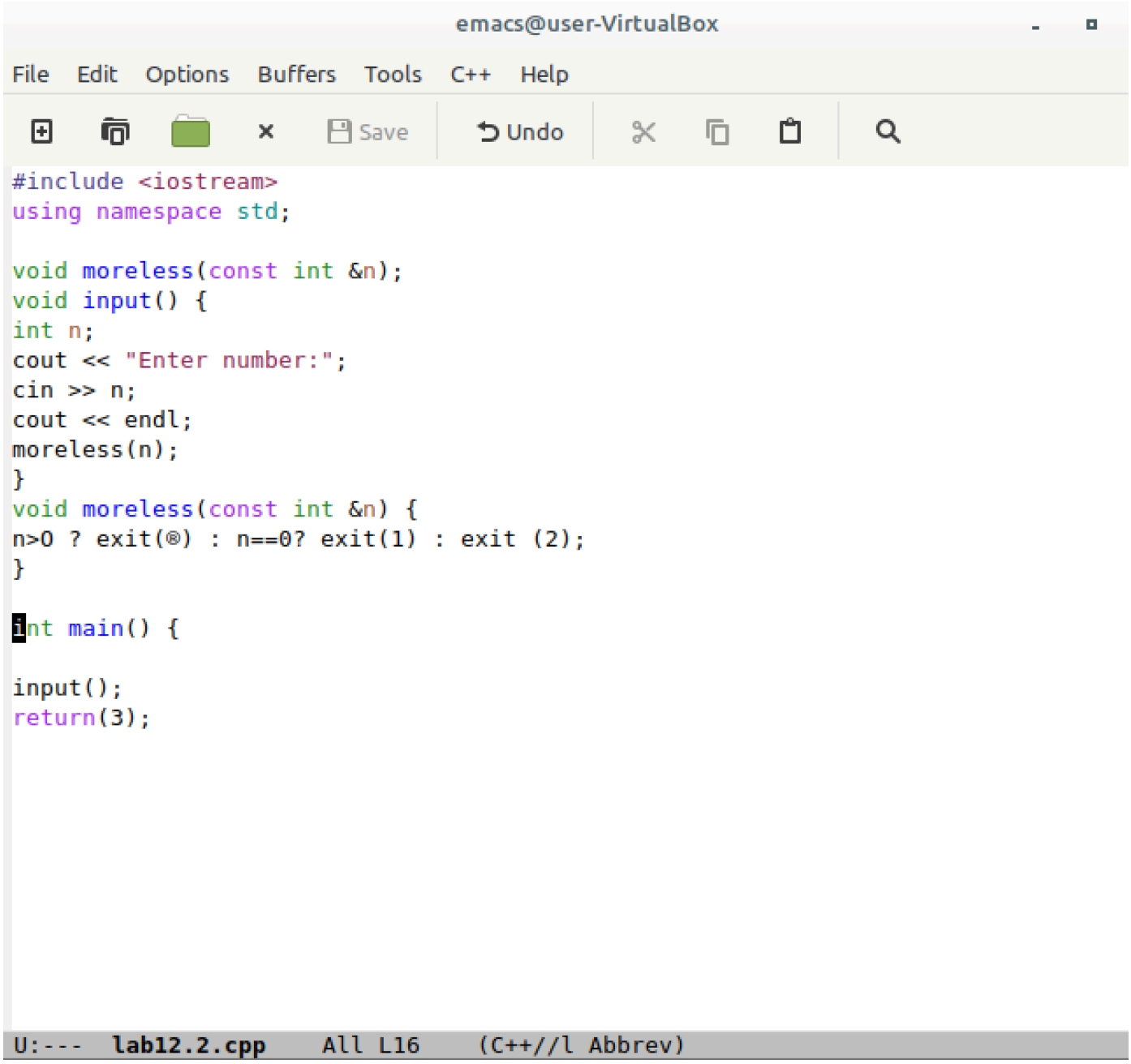
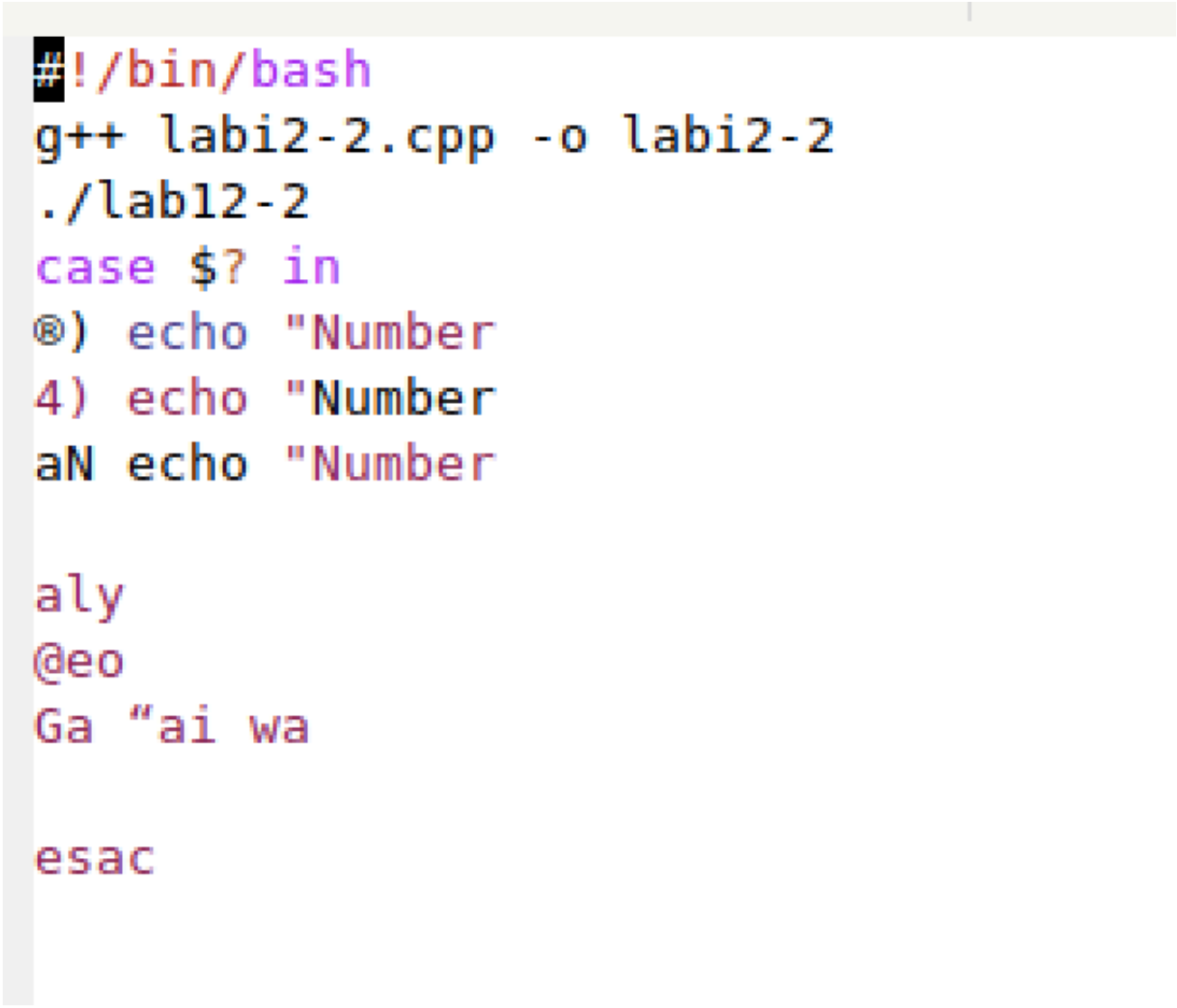


Рис 3. 2 «Сравненеи чисел»



```
#!/bin/bash
g++ labi2-2.cpp -o labi2-2
./lab12-2
case $? in
®) echo "Number
4) echo "Number
aN echo "Number

aly
@eo
Ga “ai wa

esac
```

Рис 3. 3 «Сравненеи чисел»

3. 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).

```
#!/bin/bash

let delete=0;

while getopts c:d letters

do case $letters in
c)create=1; arg=SOPTARG;;
d)delete=1;;
*) echo wrongoption $letters

esac
done

if ((delete==0))
then for((i=1;i<=arg;i++))
do touch ${i}.txt
echo document NS{i} was created
done
fi
if ((delete==1))
then for((i=1;i<=arg;i++))
do rm ${i}.txt

echo document WS{i} was deleted
done
fi
```

Рис 3. 4 «Создание указанное число файлов»

Рис 3. 5 «Создание указанное число файлов»

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).

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

Рис 3. 6 «Упаковка файлов в архив»

```
user@user-VirtualBox:~/work/os/lab12$ chmod +x lab12-4.sh
user@user-VirtualBox:~/work/os/lab12$ ./lab12-4.sh
welcome to compressor-tar
which directory you need to compress?
/home/user/work/os
tar: Удаляется начальный `/' из имен объектов
tar: /home/user/work/os/lab12/compress.tar: файл является архивом; не сброшен
user@user-VirtualBox:~/work/os/lab12$ ls -l
итого 292
-rw-rw-r-- 1 user user 256000 июн 13 01:32  compress.tar
-rw-rw-r-- 1 user user    820 июн 12 18:07 '#lab12-1.sh#'
-rw-rw-r-- 1 user user    823 июн 12 18:04  lab12-1.sh
-rw-rw-r-- 1 user user      0 июн 12 17:59  lab12-1.sh~
-rw-rw-r-- 1 user user    267 июн 12 18:18  lab12.2.cpp
-rw-rw-r-- 1 user user      0 июн 12 18:18  lab12.2.cpp~
-rw-rw-r-- 1 user user    136 июн 12 18:17  lab12-2.sh
-rw-rw-r-- 1 user user    267 июн 12 18:14  lab12-2.sh~
-rwxrwxr-x 1 user user    401 июн 12 18:32 '#lab12-3.sh#'
-rwxrwxr-x 1 user user    400 июн 12 18:50  lab12-3.sh
-rwxrwxr-x 1 user user    404 июн 12 18:40  lab12-3.sh~
-rwxrwxr-x 1 user user    208 июн 13 01:28  lab12-4.sh
-rw-rw-r-- 1 user user    823 июн 12 18:04  lab12-4.sh~
user@user-VirtualBox:~/work/os/lab12$ d
```

Рис 3. 7 «Упаковка файлов в архив»

4) Выводы согласованные с заданием работы

В результате выполнения этой работы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

5) Ответы на контрольные вопросы

1. Каково предназначение команды getopts?

getpods - эта утилита анализирует аргументы команд из исполняемого файла

2. Какое отношение метасимволы имеют к генерации имён файлов?

   Следующие метасимволы используют для генерации имен файлов:

```
* – любая или пустая последовательность символов

? – один любой символ
[...] – любой из символов указанных в квадратных скобках с перечислением или
указанием диапазона
cat N* – выдает все файлы начинающиеся с N
cat *N* – выдает все файлы содержащие N
cat – выдаст все файлы с однобуквенным расширением hello.o, hello.c, но не
hello.cpp
program.? – выдаст program.com
cat [a–d]* – выдаст файлы которые начинаются с буквы a и заканчиваются d
```

3. Какие операторы управления действиями вы знаете?

Операторы управления действиями - > (вывод информации), < (ввод информации), & (управляет потоком исполнения команд), && (запускает исполнения команды или команд в фоне), | (передает данные между программами), || (проверяет код завершения предыдущей команды)

4. Какие операторы используются для прерывания цикла?

Команда break служит для прерывания цикла и передает управление программой команде, которая идет следующей за циклом

5. Для чего нужны команды false и true?

alse - логическое нет, отрицание, то есть дальнейшую остановку программы или переход в другую ветвь ветвления программы в зависимости от условий
true - логическое да, согласие на дальнейшее исполнение программы согласно заданным условия

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

Строка if test -f man$s/$i.$s означает условие для проверки существования файла man

7. Объясните различия между конструкциями while и until

while - выполняет цикл пока указанное в нем условие истинно (1, true), а until - выполняет цикл пока указанное в нем условие ложно (0, false)