

Цель работы

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

Задание

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;
- `-р`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 `-р`.

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.

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).

4. Написать командный файл, который с помощью команды `tar` запаковывает в архив все файлы в указанной директории.

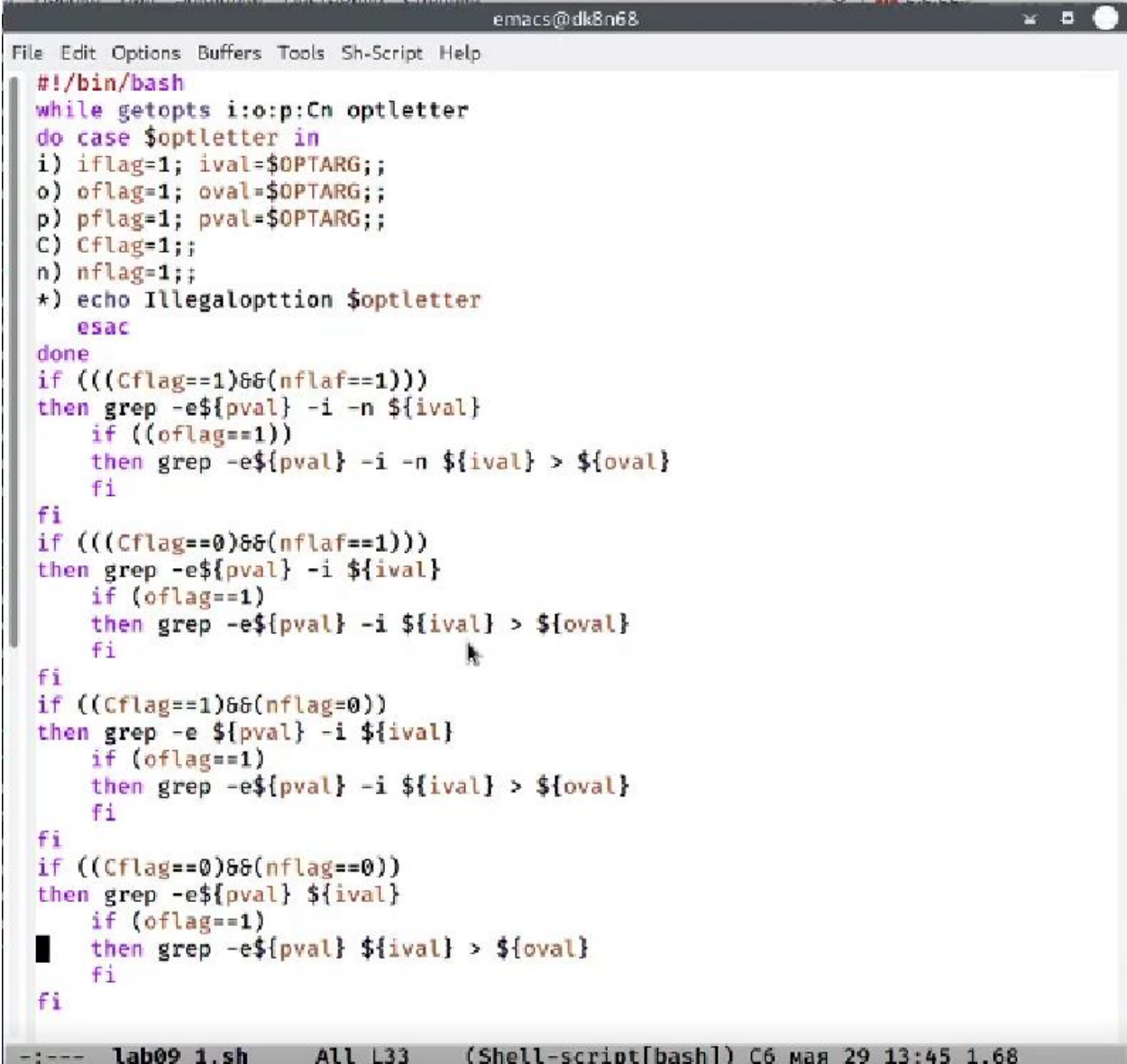
Выполнение лабораторной работы

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;
- `-ршаблон` — указать шаблон для поиска;
- `-C` — различать большие и малые буквы;
- `-n` — выдавать номера строк.

а затем ищет в указанном файле нужные строки, определяемые ключом `-р`.

```
user63@dk8n68 - $ touch lab09_1.sh
user63@dk8n68 - $ chmod +x lab09_1.sh
```



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
while getopts i:o:p:Cn optletter
do case $optletter in
i) iflag=1; ival=$OPTARG;;
o) oflag=1; oval=$OPTARG;;
p) pflag=1; pval=$OPTARG;;
C) Cflag=1;;
n) nflag=1;;
*) echo Illegal option $optletter
esac
done
if (((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
if ((oflag==1))
then grep -e${pval} -i -n ${ival} > ${oval}
fi
fi
if (((Cflag==0)&&(nflag==1)))
then grep -e${pval} -i ${ival}
if (oflag==1)
then grep -e${pval} -i ${ival} > ${oval}
fi
fi
if ((Cflag==1)&&(nflag==0))
then grep -e ${pval} -i ${ival}
if (oflag==1)
then grep -e${pval} -i ${ival} > ${oval}
fi
fi
if ((Cflag==0)&&(nflag==0))
then grep -e${pval} ${ival}
if (oflag==1)
then grep -e${pval} ${ival} > ${oval}
fi
fi
fi

--:--- lab09_1.sh All L33 (Shell-script[bash]) C6 мая 29 13:45 1.68
```

2. Написак 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 вызывает эту программу и, проанализировав с помощью команды `$?`, выдает сообщение о том, какое число было введено.

```
~: bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
user63@dk8n68 ~ $ touch lab09_2.c
user63@dk8n68 ~ $ chmod +x lab09_2.c
```

```
lab09_2.c  [-M--] 16
#include <stdio.h>
#include <stdlib.h>

int main ()
{
    int s;
    printf("input: ");
    scanf("%i", &s);
    if (s==0) exit (0);
    else if (s<0) exit (1);
    else if (s>0) exit (2);
    return 3;
}
```

```
emacs@dk8n68
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
gcc -o cprog lab09_2.c
./cprog
case $? in
    0)echo "ravno 0";;
    1)echo "menshe 0";;
    2)echo "bolshe 0";;
esac

-:*** lab09_2.sh All L6 (Shell-script[bash]) C6 мая 29 13:57 1.70
```

```

user63@dk8n68 ~ $ ./lab09_2.sh
input: 0
ravno 0
user63@dk8n68 ~ $ ./lab09_2.sh
input: 1
bolshe 0
user63@dk8n68 ~ $ ./lab09_2.sh
input: -1
menshe 0

```

3. Написак 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 уметь удалять все созданные им файлы

```

user63@dk8n68 ~ $ touch lab09_3.sh
user63@dk8n68 ~ $ chmod +x lab09_3.sh

```

```

#!/bin/bash
let dflag=0;
while getopts a:d optletter
do case $optletter in
    a) aflag=1; aval=$OPTARG;;
    B) dflag=1;;
    *) echo Illegaloption $optletter
    esac
done
#echo ${aval}
if ((dflag==0))
then for ((i=1;i<=aval;i++))
do touch ${i}.txt
done
fi
if ((dflag==1))
then for ((i=1;i<=aval;i++))
do rm ${i}.txt
done

```

lab09_3.sh Top L5 (Shell-script[bash])

```

user63@dk8n68 ~ $ ./lab09_3.sh -a3

```

```

user63@dk8n68 ~ $ ls
1.txt      file.txt    my_os      work
2.txt      games      play       Видео
3.txt      lab09_1.sh public      Документы
australia  lab09_1.sh~ public_html Загрузки
conf.txt   lab09_2.c  script2.sh Изображения
cprog      lab09_2.c~ script.sh   Музыка
equiplist2 lab09_2.sh ski.places  Общедоступные
feathers    lab09_2.sh~ tmp         'Рабочий стол'
file2.sh    lab09_3.sh user        Шаблоны
file.sh     lab09_3.sh~ user63

```

```

user63@dk8n68 ~ $ ./lab09_3.sh -a3 -d

```

4. Написал командный файл, который с помощью команды tar запаковывает в

архив все файлы в указанной директории.

```

user63@dk8n68 ~ $ touch lab09_4.sh
user63@dk8n68 ~ $ chmod +x lab09_4.sh

```

```

emacs@dk8n68
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
tar -cf 9.tar $@
tar -cf 9l.tar
find $@ -mtime -7 -exec tar -rf 9l.tar "{}" ";"

```

```

U:★★- Lab09_4.sh All L4 (Shell-script[sh]) C6 мая 29 14:15 1.85

```

```
user63@dk8n68 ~ $ ./lab09_4.sh /play
tar: Удаляется начальный '/' из имен объектов
tar: /play: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: Завершение работы с состоянием неисправности из-за возникших ошибок
tar: Робкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для получения более подробного описания.
find: '/play': Нет такого файла или каталога
```

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.

Выводы

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