

## Цель работы

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

## Ход работы

1. Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `--inputfile` — прочитать данные из указанного файла; `--outputfile` — вывести данные в указанный файл; `-rшаблон` — 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`. (см. Рис. 1,2)
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.(см. Рис. 3,4)
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).(см. Рис. 5,6)
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`). (см. Рис. 7,8)

## Вывод

Изучили основы программирования в оболочке ОС UNIX/Linux. Научились писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

## Контрольные вопросы

1. Весьма необходимой при программировании является команда `getopts`, которая 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 аргументы, 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`. Предположим, необходимо распознать командную строку следующего формата: `testprog -ifile_in.txt -ofile_out.doc -L -t -r` Вот как выглядит использование оператора `getopts` в этом случае: `while getopts o:i:Ltr optletter do case $optletter in o) oflag=1; oval=$OPTARG;; i) iflag=1; ival=$OPTARG;; L) Lflag=1;; t) tflag=1;; r) rflag=1;;`

\*) `echo Illegal option $optletter esac done` Функция `getopts` включает две специальные переменные среды – `OPTARG` и `OPTIND`. Если ожидается дополнительное значение, то `OPTARG` устанавливается в значение этого аргумента (будет равна `file_in.txt` для опции `i` и `file_out.doc` для опции `o`).

`OPTIND` является числовым индексом на упомянутый аргумент. Функция `getopts` также понимает переменные типа массив, следовательно, можно использовать ее в функции не только для синтаксического анализа аргументов функций, но и для анализа введенных пользователем данных.

2. 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 находящемуся между символами `c1` и `c2`.

`echo *` — выведет имена всех файлов текущего каталога, что представляет собой простейший аналог команды `ls`;

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

`echo prog.?` — выдаст все файлы, состоящие из пяти или шести символов, первыми пятью символами которых являются `prog.`

`[a-z]*` — соответствует произвольному имени файла в текущем каталоге, начинающемуся с любой строчной буквы латинского алфавита.

3. Часто бывает необходимо обеспечить проведение каких-либо действий циклически и управление дальнейшими действиями в зависимости от результатов проверки некоторого условия. Для решения подобных задач язык программирования `bash` предоставляет Вам возможность использовать такие управляющие конструкции, как `for`, `case`, `if` и `while`. С точки зрения командного процессора эти управляющие конструкции являются обычными командами и могут использоваться как при создании командных файлов, так и при работе в интерактивном режиме. Команды, реализующие подобные конструкции, по сути дела являются операторами языка программирования `bash`. Поэтому при описании языка программирования `bash` термин оператор будет использоваться наравне с термином команда.

4. Два несложных способа позволяют вам прерывать циклы в оболочке `bash`. Команда `break` завершает выполнение цикла, а команда `continue` завершает данную итерацию блока операторов. Команда `break` полезна для завершения цикла `while` в ситуациях, когда условие перестает быть правильным. Пример бесконечного цикла `while`, с прерыванием в момент, когда файл перестает существовать:

```
while true
do
    if [! -f $file]
```

```
then

break

fi

sleep 10

done
```

5. Команды ОС UNIX возвращают код завершения, значение которого может быть использовано для принятия решения о дальнейших действиях. Команда `test`, например, создана специально для использования в командных файлах. Единственная функция этой
6. Введенная строка означает условие существования файла `man$s/$i.$s`
7. Если речь идет о 2-х параллельных действиях, то это `while`. когда мы показываем, что сначала делается 1-е действие. потом оно заканчивается при наступлении 2-го действия,

## Приложение

```
#!/bin/bash
while getopts i:op: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} > ${oval}
    fi
fi
if (((Cflag==1)&&(nflag==0)))
then grep -e${pval} -i ${ival}
    if ((oflag==1))
    then grep -e${pval} -i ${ival} > ${oval}
    fi
fi
if (((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -n ${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
```

Рис.1

```
Last login: Fri May 28 16:00:11 on tty001
(base) dimashestakov@MBP-Dima ~ % vim lab09.sh

(base) dimashestakov@MBP-Dima ~ % vim lab09.txt
(base) dimashestakov@MBP-Dima ~ % bash lab09.sh -ilab09.txt -olab09.txt -pice
Some say in ice.
But if it had to perish twice.
To say that for destruction ice
And would suffice.
(base) dimashestakov@MBP-Dima ~ % bash lab09.sh -ilab09.txt -olab09.txt -pice -n
(base) dimashestakov@MBP-Dima ~ % bash lab09.sh -ilab09.txt -olab09.txt -pice -C
(base) dimashestakov@MBP-Dima ~ %
```

Рис.2

```
#include<stdlib.h>
#include<stdio.h>

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

"lab09.c" 12L, 197C

1,1

All

Рис.3

```
~/bin/bash
gcc -o cprog lab09.c
./cprog
case $? in
    0) echo 'input number is equal to 0';;
    1) echo 'input number is smaller than 1';;
    2) echo 'input number is bigger than 0';;
esac
```

**Рис.4**

```

(base) dimashestakov@MBP-Dima ~ % vim lab09.c
(base) dimashestakov@MBP-Dima ~ % vim lab09.sh
(base) dimashestakov@MBP-Dima ~ % bash lab09.sh
input: 1
input number is bigger then 0
(base) dimashestakov@MBP-Dima ~ %

```

**Рис.5**

```
#!/bin/bash
let dflag=0;
while getopts a:d optletter
do case $optletter in
    a) dflag=1; aval=$OPTARG
    d) dflag=1;
    *) echo Illegal opttion $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
fi
```

**Рис.6**

```
#!/bin/bash
tar -cf 9.tar $@
tar -cf 9l.tar
find $@ -mtime -7 -exec tar -rf 9l.tar '{}' ';'

~
~
~
~
~
~
~
~
~
~
```

**Рис.7**



```
(base) dimashestakov@MBP-Dima ~ % ls -l
total 96
-rw-r--r--  1 dimashestakov  staff    0 28 май 17:13 1.txt
-rw-r--r--  1 dimashestakov  staff    0 28 май 17:13 2.txt
-rw-r--r--  1 dimashestakov  staff    0 28 май 17:13 3.txt
-rw-r--r--  1 dimashestakov  staff    0 28 май 17:13 4.txt
drwx-----  5 dimashestakov  staff  160 11 апр 13:55 Applications
drwx-----@  5 dimashestakov  staff  160 28 май 13:26 Applications (Parallels)
drwx-----@ 75 dimashestakov  staff 2400 28 май 17:05 Desktop
drwx-----@ 29 dimashestakov  staff  928 10 янв 12:30 Documents
drwx-----@ 426 dimashestakov  staff 13632 28 май 16:34 Downloads
drwx-----@ 81 dimashestakov  staff 2592 30 апр 12:10 Library
drwx-----+  7 dimashestakov  staff  224 23 апр 11:50 Movies
drwx-----+  8 dimashestakov  staff  256  4 авг 2020 Music
drwx-----  5 dimashestakov  staff  160 23 сен 2020 Parallels
drwx-----+ 11 dimashestakov  staff  352  7 фев 09:11 Pictures
drwxr-xr-x+  4 dimashestakov  staff  128 11 июл 2020 Public
drwxr-xr-x  3 dimashestakov  staff   96 30 апр 12:12 VirtualBox VMs
-rwxr-xr-x  1 dimashestakov  staff 12644 28 май 17:04 cprog
-rw-r--r--  1 dimashestakov  staff  341 28 май 17:13 lab09.sh
drwxr-xr-x  3 dimashestakov  staff   96 30 апр 13:03 lab_01
drwxr-xr-x  3 dimashestakov  staff   96 30 апр 20:37 lab_02
drwxr-xr-x  3 dimashestakov  staff   96 30 апр 21:24 lab_03
drwxr-xr-x  3 dimashestakov  staff   96 14 май 23:58 lab_07
drwxr-xr-x  3 dimashestakov  staff   96 22 май 17:12 lab_10
drwxr-xr-x  3 dimashestakov  staff   96 28 май 15:57 lab_11
-rwxr-xr-x  1 dimashestakov  staff 22576 24 май 17:15 main
drwxr-xr-x  4 dimashestakov  staff  128 23 апр 12:03 opt
-rw-r--r--  1 dimashestakov  staff  467 24 май 17:26 test.cpp
```

Рис.8