

Лабораторная работа № 12.

Кальсин З. А.

¹RUDN University, Moscow, Russian Federation

Цель работы

Изучить основы программирования в оболочке ОС UNIX, научиться

Выполнение лабораторной работы

Выполнение лабораторной работы

Ход работы:

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

- `-iinputfile` — прочитать данные из указанного файла;
- `-ooutputfile` — 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 `-p`.

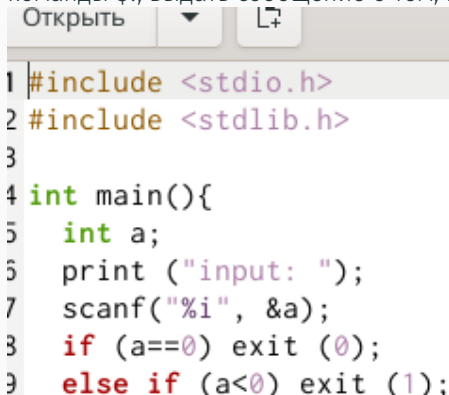
```
Two roads diverged in a yellow wood,  
And sorry I could not travel both
```

Слайд 1

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 Illegaloption $optletter
    esac
done
if (((Cflag==1)&&(nflag==1)))
then grep -e${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} > ${oval}
    fi
fi
if (((Cflag==1)&&(nflag==0)))
then grep -e${pval} ${ival}
    if ((oflag==1))
    then grep -e${pval} ${ival} > ${oval}
    fi
fi
fi
```

2. Написал 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.



```
Открыть ▼ L7
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int a;
6     print ("input: ");
7     scanf("%i", &a);
8     if (a==0) exit (0);
9     else if (a<0) exit (1);
```

```
file Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
gcc -o cprog script0.c
./cprog
case $? in
    0)echo "input number is equal to 0";;
    1)echo "input number is smaller then 0";;
    2)echo "input number is bigger then 0";;
esac
```



```
zakaIjsin@dk8n68 ~ $ bash script2.sh
input: 0
input number is equal to 0
zakaIjsin@dk8n68 ~ $ bash script2.sh
input: 17
input number is bigger then 0
zakaIjsin@dk8n68 ~ $ bash script2.sh
input: -9
input number is smaller then 0
zakaIjsin@dk8n68 ~ $ █
```

3. Написал 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).

```
e Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
let dflag=0;
while getopts a:d optletter
do case $optletter in
    a) aflag=1; aval=$OPTARG;;
    d) dflag=1;;
    *) echo Illegal option $optletter
        esac
done
#echo ${aval}
if ((dflag==0))
then for ((i=1;i<=aval;i++))
    do rm ${i}.txt
    done
fi
```

```
zakaljsin@dk8n68 ~ $ ls -l
итого 1184
-rw-rw-r-- 1 zakaljsin studsci      0 мая 14 16:34 abc1
drwxr-xr-x 2 zakaljsin studsci    2048 мая 14 16:56 abcl
drwxr--r-x 2 zakaljsin studsci    2048 мая 14 17:18 australia
-rw-r--r-- 1 zakaljsin studsci    1233 мая 15 12:19 conf.txt
-rwxr-xr-x 1 zakaljsin studsci    8072 мая 22 13:56 cprog
drwxr-xr-x 2 zakaljsin studsci    2048 мая 14 18:00 etc
-rw-r--r-- 1 zakaljsin studsci      0 мая 14 17:26 feathers
-rw-r--r-- 1 zakaljsin studsci    4185 мая 15 12:17 file.txt
drwxr-xr-x 3 zakaljsin studsci    2048 мая 14 13:15 GNUstep
-rwxr-xr-x 1 zakaljsin studsci     952 сен 10 2020 hello
-rw-r--r-- 1 zakaljsin studsci     211 сен 10 2020 hello.asm
-rw-r--r-- 1 zakaljsin studsci      0 сен 10 2020 lab02.asm
drwxr-xr-x 2 zakaljsin studsci    2048 сен 24 2020 lab03a
```

```

drwxr-xr-x 2 zakaljsin studsci 2048 сен 9 2020 abc1
zakaljsin@dk8n68 ~ $ bash script3.sh -a4 -d
zakaljsin@dk8n68 ~ $ ls -l
итого 1184
-rw-rw-r-- 1 zakaljsin studsci 0 мая 14 16:34 abc1
drwxr-xr-x 2 zakaljsin studsci 2048 мая 14 16:56 abc1
drwxr--r-x 2 zakaljsin studsci 2048 мая 14 17:18 australia
-rw-r--r-- 1 zakaljsin studsci 1233 мая 15 12:19 conf.txt
-rwxr-xr-x 1 zakaljsin studsci 8072 мая 22 13:56 cprog
drwxr-xr-x 2 zakaljsin studsci 2048 мая 14 18:00 etc
-rw-r--r-- 1 zakaljsin studsci 0 мая 14 17:26 feathers
-rw-r--r-- 1 zakaljsin studsci 4185 мая 15 12:17 file.txt
drwxr-xr-x 3 zakaljsin studsci 2048 мая 14 13:15 GNUstep
-rwxr-xr-x 1 zakaljsin studsci 952 сен 10 2020 hello
-rw-r--r-- 1 zakaljsin studsci 211 сен 10 2020 hello.asm

```

4. Написал командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировала его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовала команду find).

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

```

zakaljsin@dk8n68 ~ $ bash script4.sh /script
tar: Удаляется начальный '/' из имен объектов
tar: /script: Функция stat завершилась с ошибкой: Нет такого файла или каталога
tar: Завершение работы с состоянием неисправности из-за возникших ошибок
tar: Роккий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
find: '/script': Нет такого файла или каталога
zakaljsin@dk8n68 ~ $ ls
9 tar          GNUstep      lab07        monthly      script0.c     script.sh~   Загрузки
abc1          hello        lab10.sh     my_os        script1.sh     script.txt   Изображения
abcl          hello.asm    lab10.sh~    Otchet_laboratornaya_2.docx  script1.sh~    ski_places   Музыка
australia     lab02.asm    Lab05        play         script2.sh     tmp          Общедоступные
conf.txt      lab03a       letters      PPP          script2.sh~    Vai         'Рабочий стол'
cprog        lab03b       logfile      Prog1        script3.sh     work        Шаблоны
etc           lab05        may          public       script3.sh~    XXX
feathers      lab06        memos        public_html  script4.sh     Видео
file.txt      lab06.o     misk         reports      script4.sh~    Документы
zakaljsin@dk8n68 ~ $

```

Вывод 10

Изучил основы программирования в оболочке ОС UNIX, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов. # Ответы на контрольн## Слайд 1

с собой мощный экраннй редактор текста, написанный на 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

[...] 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 `mans/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.