

091 Отчет по лабораторной работе №12

Тема:

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: *Операционные системы*

Студент: Яссин Оулед Салем

Группа: НПИБД02-20

Москва, 2021 г.

Цель работы

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

Введение

Операции с блоками кода — ключ к структурированным и упорядоченным сценариям оболочки. Циклы и ветвления являются теми инструментальными средствами, которые предоставляют возможность достигнуть этой цели.

Циклы

Цикл это блок команд, который повторяется (итерируется) до тех пор, пока не будет выполнено условие выхода из цикла .

Ход работы:

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

* `-inputfile` — прочитать данные из указанного файла; * `-outputfile` — 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`.

```
yassine@ubuntu:~$ cat lab09_1.txt
"Nina, this is Carmen," Felipa introduced her.
This is our planting-ground.
"Who is this?" asked the Wizard, curiously.
yassine@ubuntu:~$ █
```

```
File Edit Options Buffers Tools Sh-Script Help
[Icons: File, Folder, Folder with plus, Close, Save, Undo, Cut, Copy, Paste, Search]

#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0 ;
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 $optletter
esac
done
if (($pflag==0))
then echo "шаблон не найден "
else
if (($iflag==0))
then echo "файл не найден "
else
if((oflag==0))
then if (($cflag))
then if (($nflag==0))
then grep $pval $ival
else grep -n $pval $ival
fi
else if (($nflag==0))
then grep -i $pval $ival
else grep -i -n $pval $ival
fi
fi
fi
fi
fi
```

```
yassine@ubuntu:~$ ./lab09.sh -i lab09_1.txt -o lab09_2.txt -p is -C -n
yassine@ubuntu:~$ cat lab09_2.txt
1:"Nina, this is Carmen," Felipa introduced her.
2:This is our planting-ground.
3:"Who is this?" asked the Wizard, curiously.
yassine@ubuntu:~$
```

1. 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 о коде завершения в оболочку. Командный файл вызывает эту программу и, проанализировав с помощью команды \$?, выдает сообщение о том, какое число было введено.

```
#!/bin/bash
gcc include.c -o include
./include
code=$?
case $code in
0) echo "число меньше 0";;
1) echo "число больше 0";;
2) echo "число равно 0"
esac
```

```

/home/ya-nclude.c [----] 12 L: [ 1+ 0 1/ 13] *(12
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    printf("введите числ\n");
    int a;
    scan ("%d",&a);
    if (a<0) exit(0) ;
    if (a>0) exit (1);
    if (a==0) exit(2);
    return 0;
}

```

Введите число
-9
число меньше 0

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 удаляет все созданные им файлы (если они существуют).

```

#!/bin/bash
let dflag=0
while getopts a:d optletter
do case $optletter in
    a) aflag=1; aval=$OPTARG;;
    d) dflag=1;;
    *) echo не правильный ввод $optletter
        esac
done
#ech ${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

```

```
yassine@ubuntu:~$ bash lab09_3.sh -a4 -d
```

```
yassine@ubuntu:~$ ls -l
```

```

total 308
-rw-rw-r-- 1 yassine yassine  0 мая 29 05:54 a1.txt
-rw-rw-r-- 1 yassine yassine  0 мая 29 05:54 a2.txt
-rw-rw-r-- 1 yassine yassine  0 мая 12 08:13 abc1
-rwxrwxr-x 1 yassine yassine 9064 окт  9  2020 asdfg
-rw-rw-r-- 1 yassine yassine  421 окт  9  2020 asdfg.asm
-rw-rw-r-- 1 yassine yassine 1104 окт  9  2020 asdfg.o
drwxrwxr-x 2 yassine yassine 4096 мая 12 08:25 australia
drwxrwxr-x 2 yassine yassine 4096 мая 28 05:02 backup
-rw-rw-r-- 1 yassine yassine  98 мая 15 13:18 c++.cpp
drwxr-xr-x 2 yassine yassine 4096 мая 28 04:59 Desktop
drwxr-xr-x 6 yassine yassine 4096 окт 24  2020 Documents
drwxr-xr-x 2 yassine yassine 4096 мая 10 08:48 Downloads
-rw-rw-r-- 1 yassine yassine  187 мая 29 08:00 f1.txt
-rw-rw-r-- 1 yassine yassine  113 мая 29 07:58 f1.txt~
-rw-rw-r-- 1 yassine yassine   0 мая 29 07:55 f2.txt
-rw-rw-r-- 1 yassine yassine   0 мая 12 08:26 feathers
-rw-rw-r-- 1 yassine yassine   0 мая 12 08:30 file.old

```

```
yassine@ubuntu:~$ bash lab09_3.sh
yassine@ubuntu:~$ ls -l
total 308
-rw-rw-r-- 1 yassine yassine 0 мая 29 05:54 a1.txt
-rw-rw-r-- 1 yassine yassine 0 мая 29 05:54 a2.txt
-rw-rw-r-- 1 yassine yassine 0 мая 12 08:13 abc1
-rwxrwxr-x 1 yassine yassine 9064 окт 9 2020 asdfg
-rw-rw-r-- 1 yassine yassine 421 окт 9 2020 asdfg.asm
-rw-rw-r-- 1 yassine yassine 1104 окт 9 2020 asdfg.o
drwxrwxr-x 2 yassine yassine 4096 мая 12 08:25 australia
drwxrwxr-x 2 yassine yassine 4096 мая 28 05:02 backup
-rw-rw-r-- 1 yassine yassine 98 мая 15 13:18 c++.cpp
drwxr-xr-x 2 yassine yassine 4096 мая 28 04:59 Desktop
drwxr-xr-x 6 yassine yassine 4096 окт 24 2020 Documents
```

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`).

```
#!/bin/bash
tar -cf lab12_4_l.tar $@
fin $@ -mtime -7 -exec tar -rf lab12_4_l_modif.tar {} \;
```

-	#lab#	lab11_4.sh~	lab12_3.sh~	my_os	Документы
9.tar	#lab07.sh#	lab11.sh	lab12_4_l_modif.tar	play	Загрузки
abc1	lab07.sh	lab11.sh~	lab12_4_l.tar	reports	Изображения
abcd1	lab_08	lab12_2.C++~	lab12_4.sh	ski.places	Музыка
australia	lab11-12	lab12_2.cpp	lab12_4.sh~	test	Общедоступные
backup	lab11_2.c~	lab12_2.cpp~	lab12.sh	testlab	Рабочий стол
conf.txt	lab11_2.sh	lab12_2.sh	lab12.sh~	text.txt	Шаблоны
cprog	lab11_2.sh~	lab12_2.sh~	lab12.txt	tutorial	
feathers	lab11_3.sh	lab12_2.txt	lab12.txt~	work	
file.txt	lab11_3.sh~	lab12_2.txt~	may	x	
k.sh~	lab11_4.sh	lab12_3.sh	monthly	Видео	

Вывод

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

Библиография

[Домашний склерозник](#)

Ответы на 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.