

Отчёт по лабораторной работе №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Кочина Д. С.

21 апреля 2023

Российский университет дружбы народов, Москва, Россия

Вводная часть

Целью данной лабораторной работы является изучение основ программирования в оболочке ОС UNIX. А также приобретение практических навыков написания более сложные командных файлов с использованием логических управляющих конструкций и циклов.

Основная часть

- Используя команды `getopts` `grep`, написала командный файл, который анализирует командную строку с ключами.

```
prog1.sh - GNU Emacs at dk8n80
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0; #Инициализация переменных-флагов, присваиваем им 0
while getopts i:o:p:C:n optletter #Анализируем командную строку на наличие опций
do case $optletter in #Если опция присутствует в строке, то присваиваем ей 1
    i) iflag=1; ival=$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)) #Проверка, указан ли шаблон для поиска
then echo "Шаблон не найден"
else
    if (($iflag==0))
    then echo "Файл не найден"
    else
        if (($oflag==0))
        then if (($Cflag==0))
```

- Проверила работу написанного скрипта, используя различные опции (например, команда «./prog.sh-la1.txt-oa2.txt-pcapital-C-n»), предварительно добавив право на исполнение файла (команда «chmod+xprog1.sh») и создав 2 файла, которые необходимы для выполнения программы: a1.txt и a2.txt. Скрипт работает корректно.

```
dskochina@dk8n80 ~ $ emacs &
[1] 3813
dskochina@dk8n80 ~ $ cat a1.txt
water abc abcs
asd
prog1
water water
[1]+  Завершён          emacs
dskochina@dk8n80 ~ $ ./prog1.sh -i a1.txt -o a2.txt -p water -n
dskochina@dk8n80 ~ $ cat a2.txt
1:water abc abcs
4:water water
dskochina@dk8n80 ~ $ ./prog1.sh -i a1.txt -o a2.txt -p water -n
dskochina@dk8n80 ~ $ cat a2.txt
1:water abc abcs
4:water water
dskochina@dk8n80 ~ $ ./prog1.sh -i a1.txt -C -n
Шаблон не найден
dskochina@dk8n80 ~ $ ./prog1.sh -o a2.txt -p water -C -n
Файл не найден
dskochina@dk8n80 ~ $
```

- Написала 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. Для данной задачи я создала 2 файла: `chslo.c` и `chislo.sh` и написала соответствующие скрипты. (команды «`touch prog2.sh`» и «`emacs &`»).

File Edit Options Buffers Tools C Outline Hide/Show Help

```
#include <studio.h>
#include <stdlib.h>
int main ()
{
    printf("Введите число\n");
    int a;
    scanf("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```

- Проверила работу написанных скриптов (команда «./chislo.sh»), предварительно добавив право на исполнение файла (команда «chmod+x chislo.sh»). Скрипты работают корректно.

- Написала 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). Для данной задачи я создала файл: files.sh. и написала соответствующий скрипт.

File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help

```
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function Files()
{
    for (( i=1; i≤$number; i++ )) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt = "-r" ]
        then
            rm -f $file
        elif [ $opt = "-c" ]
        then
            touch $file
        fi
    done
}
Files
```

- Далее я проверила работу написанного скрипта (команда «./files.sh»), предварительно добавив право на исполнение файла (команда «chmod+x files.sh»). Сначала я создала три файла (команда «./files.sh-cabc#.txt3»), удовлетворяющие условию задачи, а потом удалила их (команда «./files.sh-rabc#.txt3»).

```

tskochina@dk8n80 ~ $ emacs
tskochina@dk8n80 ~ $ chmod +x files.sh
tskochina@dk8n80 ~ $ ls
a1.txt      chslo.sh      GNUstep      public      Изображения
a2.txt      chslo.sh~    my_os       public_html Музыка
backup      file2.doc     prog1.sh     ski.plases  Общедоступные
backup.sh   file.doc      prog1.sh~    tmp         'Рабочий стол'
backup.sh~  file.pdf     prog2.sh     work        Шаблоны
bin         files.sh      prog2.sh~    Видео
chslo.c     format.sh     progl1.sh    Документы
chslo.c~    format.sh~   progl1.sh~   Загрузки
tskochina@dk8n80 ~ $ ./files.sh -c abc#.txt 3
tskochina@dk8n80 ~ $ ls
a1.txt      backup.sh~    file.doc     prog1.sh     public_html  Изображения
a2.txt      bin           file.pdf     prog1.sh~    ski.plases   Музыка
abc1.txt    chslo.c       files.sh     prog2.sh     tmp          Общедоступные
abc2.txt    chslo.c~     format.sh    prog2.sh~    work         'Рабочий стол'
abc3.txt    chslo.sh      format.sh~   progl1.sh    Видео        Шаблоны
backup      chslo.sh~    GNUstep      progl1.sh~   Документы
backup.sh   file2.doc     my_os       public       Загрузки
tskochina@dk8n80 ~ $ ./files.sh -r abc#.txt 3
tskochina@dk8n80 ~ $ ls
a1.txt      chslo.sh      GNUstep      public      Изображения
a2.txt      chslo.sh~    my_os       public_html Музыка
backup      file2.doc     prog1.sh     ski.plases  Общедоступные
backup.sh   file.doc      prog1.sh~    tmp         'Рабочий стол'
backup.sh~  file.pdf     prog2.sh     work        Шаблоны
bin         files.sh      prog2.sh~    Видео
chslo.c     format.sh     progl1.sh    Документы
chslo.c~    format.sh~   progl1.sh~   Загрузки
tskochina@dk8n80 ~ $ █

```

Проверка работы скрипта

- Проверила работу скрипта, предварительно добавив для него право на выполнение (команда «chmod+x*.sh»). Проверила, появился ли каталог backup/, перейдя в него (команда «cd backup/»), посмотрела его содержимое (команда «ls») и просмотрела содержимое архива (команда «bunzip2 -cbackup.sh.bz2»).

Терминал - dskochina@dk8n80:~

Файл Правка Вид Терминал Вкладки Справка

```
dskochina@dk8n80 ~ $ touch chslo.c
dskochina@dk8n80 ~ $ touch chslo.sh
dskochina@dk8n80 ~ $ emacs &
```

```
chslo.c - GNU Emacs at dk8n80
File Edit Options Buffers Tools C Outline Hide/Show Help
#include <studio.h>
#include <studlib.h>
int main ()
{
    printf("Введите число\n");
    int a;
    scahf("%d", &a);
    if (a<0) exit(0);
    if (a>0) exit(1);
    if (a==0) exit(2);
    return 0;
}
```


- Создала файл, в котором буду писать второй скрипт, и открыла его в редакторе emacs, используя клавиши.
- Написала пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.



```
chslo.sh - GNU Emacs at dk8n80
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
#!/bin/bash
gcc chslo.c -o chslo
./chslo.sh
code=$?
case $code in
    0) echo "Число меньше 0";;
    1) echo "Число больше 0";;
    2) echo "Число равно 0";;
esac
```

- Проверила работу написанного скрипта, предварительно добавив для него право на выполнение. Вводила аргументы, количество которых меньше 10 и больше 10. Скрипт работает корректно.

```
dskochina@dk8n80 ~ $ touch files.sh  
dskochina@dk8n80 ~ $ emacs
```

Проверка работы скрипта

files.sh - GNU Emacs at dk8n80

File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help

```
#!/bin/bash
opt=$1;
format=$2;
number=$3;
function Files()
{
    for (( i=1; i≤$number; i++ )) do
        file=$(echo $format | tr '#' "$i")
        if [ $opt = "-r" ]
        then
            rm -f $file
        elif [ $opt = "-c" ]
        then
            touch $file
        fi
    done
}
Files
```

- Создала файл, в котором буду писать третий скрипт, и открыла его в редакторе emacs, используя клавиши.
- Написала командный файл – аналог команды ls. Он должен выдавать информацию о нужном каталоге и выводить информацию о возможностях доступа к файлам этого каталога.

```
dskochina@dk8n80 ~ $ touch prog4.sh  
dskochina@dk8n80 ~ $ emacs
```

- Написала 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 (использовать команду find). Для данной задачи я создала файл: prog4.sh и написала соответствующий скрипт.

File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help

```
#!/bin/bash
files=$(find ./ -maxdepth 1 -mtime -7)
listing=""
for file in "$files" ; do
    file=$(echo "$file" | cut -c 3-)
    listing="$listing $file"
done
dir=$(basename $(pwd))
tar -cvf $dir.tar $listing
```


Заключение

В ходе выполнения данной лабораторной работы я изучила основы программирования в оболочке ОС UNIX. А также приобрела практические навыки написания более сложные командных файлов с использованием логических управляющих конструкций и циклов.