

Отчёт по лабораторной работе №11

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Мокочунина В.С.

17 апреля 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Мокочунина Влада Сергеевна
- Российский университет дружбы народов
- vmokochunina@gmail.com
- https://github.com/Vmokochunina/study_2022-2023_os-intro.git

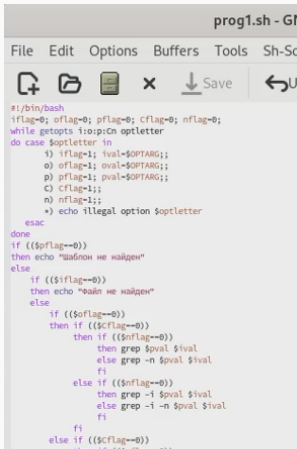
Вводная часть

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

Создание презентации

Используя команды `getopts` `grep`, написала командный файл, который анализирует

командную строку с ключами: – `-i`inputfile — прочитать данные из указанного файла; – `-o`outputfile — вывести данные в указанный файл; – `-r`шаблон — указать шаблон для поиска; – `-C` — различать большие и малые буквы; – `-n` — выдавать номера строк.(рис. (fig:001?)).



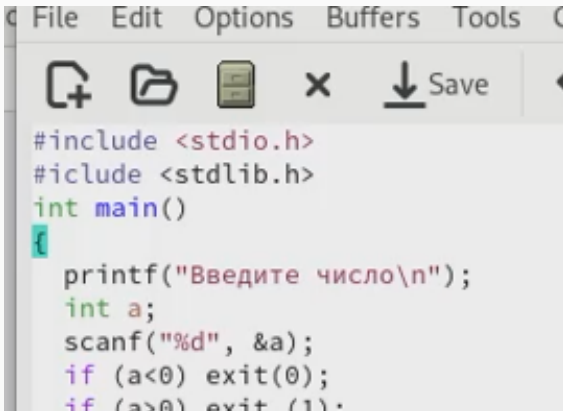
```
#!/bin/bash
iflag=0; oflag=0; pflag=0; cflag=0; nflag=0;
while getopts i:o:p:Cn optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1;  oval=$OPTARG;;
    p) pflag=1;  pval=$OPTARG;;
    C) cflag=1;;
    n) nflag=1;;
    *) echo illegal option $optletter
       esac
done
if (($pflag==0))
then echo "Шаблон не найден"
else
    if (($iflag==0))
    then echo "Файл не найден"
    else
        if (($oflag==0))
        then if (($cflag==0))
            then if (($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 $ival
        fi
    fi
    else if (($cflag==0))
        then
```

Проверила на выполнение

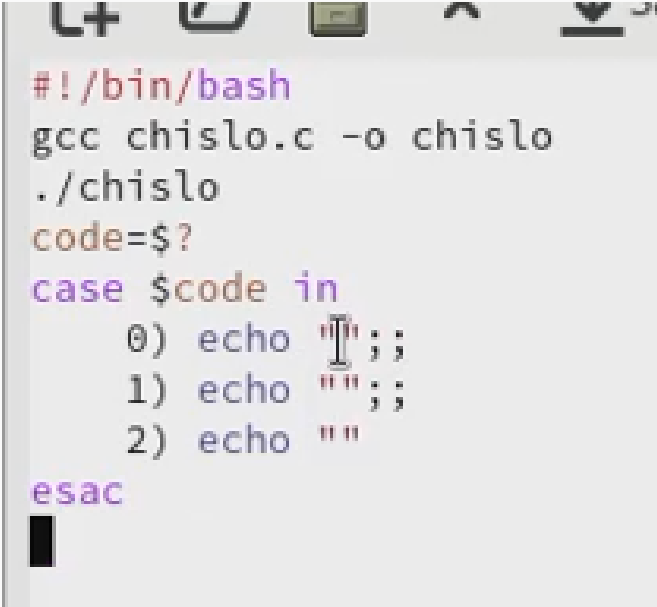
```
[vsmokochunina@10 ~]$ cat 1.txt
Hello
HELLO
hello
text
[vsmokochunina@10 ~]$ ./prog1.sh -i 1.txt -o 2.txt -p hello -C -n
[vsmokochunina@10 ~]$ cat 2.txt
1:Hello
2:HELLO
3:hello
[vsmokochunina@10 ~]$ ./prog1.sh -i 1.txt -o 2.txt -p hello -n
./prog1.sh: строка 33: prep: команда не найдена
[vsmokochunina@10 ~]$ emacs
[vsmokochunina@10 ~]$ ./prog1.sh -i 1.txt -o 2.txt -p hello -n
[vsmokochunina@10 ~]$ cat 2.txt
3:hello
[vsmokochunina@10 ~]$ ./prog1.sh -i 1.txt -C -n
Шаблон не найден
[vsmokochunina@10 ~]$ ./prog1.sh -o 2.txt -p hello -C -n
Файл не найден
[vsmokochunina@10 ~]$
```


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. Команд- ный файл должен вызывать эту программу и, проанализировав с помощью команды `$?`, выдать сообщение о том, какое число было введено

A screenshot of a text editor window showing a C program. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', and 'Tools'. Below the menu is a toolbar with icons for opening, saving, and closing files, along with a 'Save' button. The code is as follows:

```
#include <stdio.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);
```



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

The image shows a terminal window with a light gray background. At the top, there are several icons: a plus sign, a square, a folder, a person, and a downward arrow. The terminal content is a shell script. The first line is a shebang `#!/bin/bash`. The second line is `gcc chislo.c -o chislo`. The third line is `./chislo`. The fourth line is `code=$?`. The fifth line is `case $code in`. The sixth line is `0) echo " "; ;`. The seventh line is `1) echo " "; ;`. The eighth line is `2) echo " "`. The ninth line is `esac`. There is a black cursor at the end of the `esac` line.

Рис. 4: Командный файл

```
[vsmokochunina@10 ~]$ ./chislo.sh
Введите число
0
Число равно 0
[vsmokochunina@10 ~]$ ./chislo.sh
Введите число
3
Число больше 0
[vsmokochunina@10 ~]$ ./chislo.sh
Введите число
-1
Число меньше 0
```

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).

```
#!/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
}
```

Проверила на выполнение

```
[vsmokochunina@10 ~]$ ./prog3.sh -c abc#.txt 3
[vsmokochunina@10 ~]$ ls
1.txt          backup4.sh~   file1.doc     ssh
2.txt          backup.sh     file2.doc     ssh.pub
abc1.txt       backup.sh~    file.pdf      Vsmokochunina.github.io
abc2.txt       bin           file.txt      work
abc3.txt       blog          lab07.sh      Видео
abc.txt        chislo        lab07.sh~     Документы
backup         chislo.c      lab101.sh     Загрузки
backup2.sh     chislo.c~     lab101.sh~    Изображения
backup2.sh~    chislo.sh     prog1.sh      Музыка
backup3.sh     chislo.sh~    prog1.sh~     Общедоступные
backup3.sh~    conf.txt      prog3.sh      'Рабочий стол'
backup4.sh     feathers      prog3.sh~     Шаблоны
[vsmokochunina@10 ~]$ #./prog3.sh -r abc.txt 3
[vsmokochunina@10 ~]$ ./prog3.sh -r abc#.txt 3
[vsmokochunina@10 ~]$ ls
```

Рис. 7: Проверка

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).

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

```
[vsmokochunina@10 ~]$ cd ~/progs
[vsmokochunina@10 progs]$ ls -l
итого 24
-rwxr-xr-x. 1 vsmokochunina vsmokochunina 56 anp 11 20:15 backup2.sh
-rwxr-xr-x. 1 vsmokochunina vsmokochunina 352 anp 11 20:26 backup3.sh
-rwxr-xr-x. 1 vsmokochunina vsmokochunina 195 anp 11 20:32 backup4.sh
-rwxr-xr-x. 1 vsmokochunina vsmokochunina 98 anp 11 20:11 backup.sh
-rw-r--r--. 1 vsmokochunina vsmokochunina 122 anp 7 18:17 lab07.sh
-rw-r--r--. 1 vsmokochunina vsmokochunina 2 anp 11 20:08 lab101.sh
[vsmokochunina@10 progs]$ sudo ~/prog4.sh
[sudo] пароль для vsmokochunina:
```

Рис. 9: Проверка

Результаты

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

Спасибо за внимание