

Презентация по лабораторной работе №12

Программирование в командном процессоре ОС UNIX. Ветвления и циклы

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: *Операционные системы*

Работу выполняла: *Артеева Кристина Юрьевна*

1032202463

НКНБД-01-20

В ходе изучения курса "Операционные системы" передо мной встала следующая цель: изучить основы программирования в оболочке ОС UNIX, научиться писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

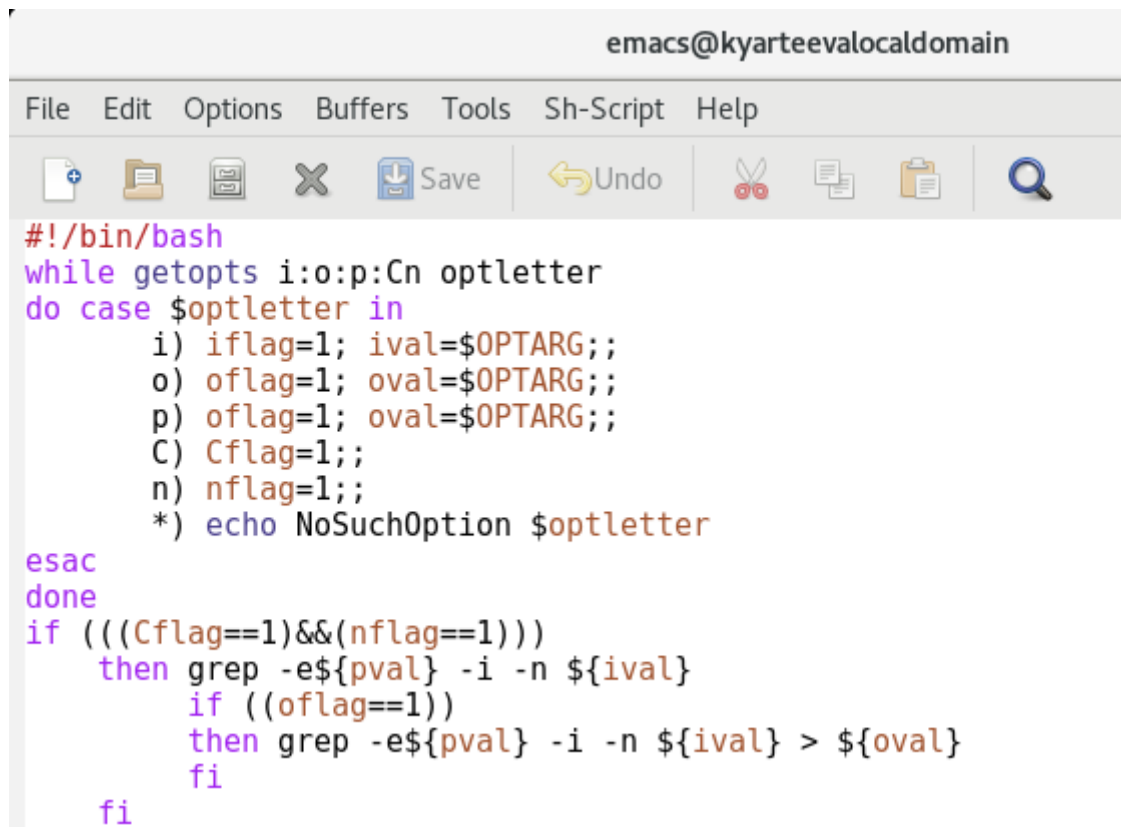
Данная задача была разбита на следующие подзадачи:

- изучить основы программирования в оболочке ОС UNIX/Linux
- написать более сложные командные файлы с использованием логических управляющих конструкций и циклов

Изучив теоретический материал, мне удалось достигнуть поставленной цели.

- В ходе работы возникали некоторые сложности, но изучив теоретические материалы, мне удалось их разрешить. Сейчас я могу писать и запускать небольшие командные файлы, а именно программы для архивации файлов, для анализа командной строки с ключами, для создания и удаления определенного количества файлов, вызов в командном файле программы на языке Си.
-

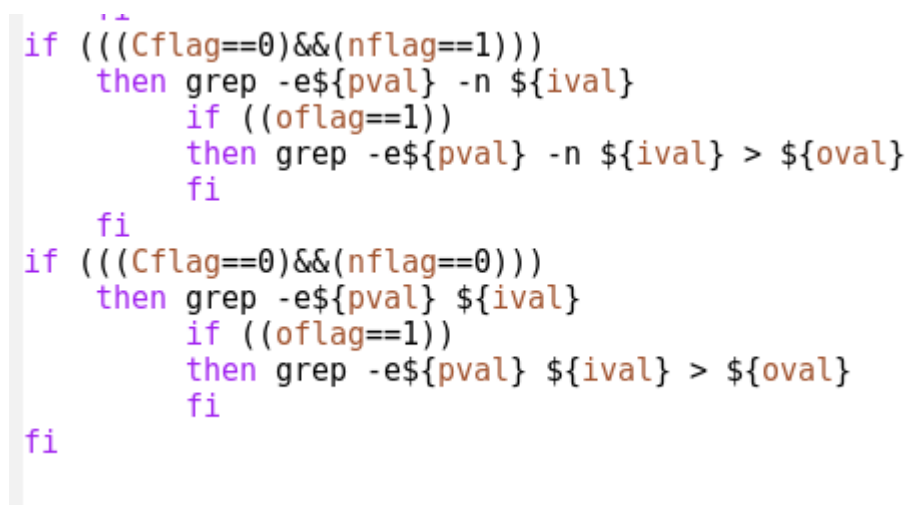
Рис.1 Анализ командной строки с ключами



The screenshot shows an Emacs editor window titled 'emacs@kyarteevalocaldomain'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for file operations and editing. The main text area displays a shell script in a monospaced font with syntax highlighting. The script uses 'while getopts' to parse flags and values, and 'grep' to compare values.

```
#!/bin/bash
while getopts i:o:p:Cn optletter
do case $optletter in
    i) iflag=1; ival=$OPTARG;;
    o) oflag=1; oval=$OPTARG;;
    p) oflag=1; oval=$OPTARG;;
    C) Cflag=1;;
    n) nflag=1;;
    *) echo NoSuchOption $optletter
esac
done
if (((Cflag==1)&&(nflag==1)))
then grep -e${pval} -i -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -i -n ${ival} > ${oval}
    fi
fi
```

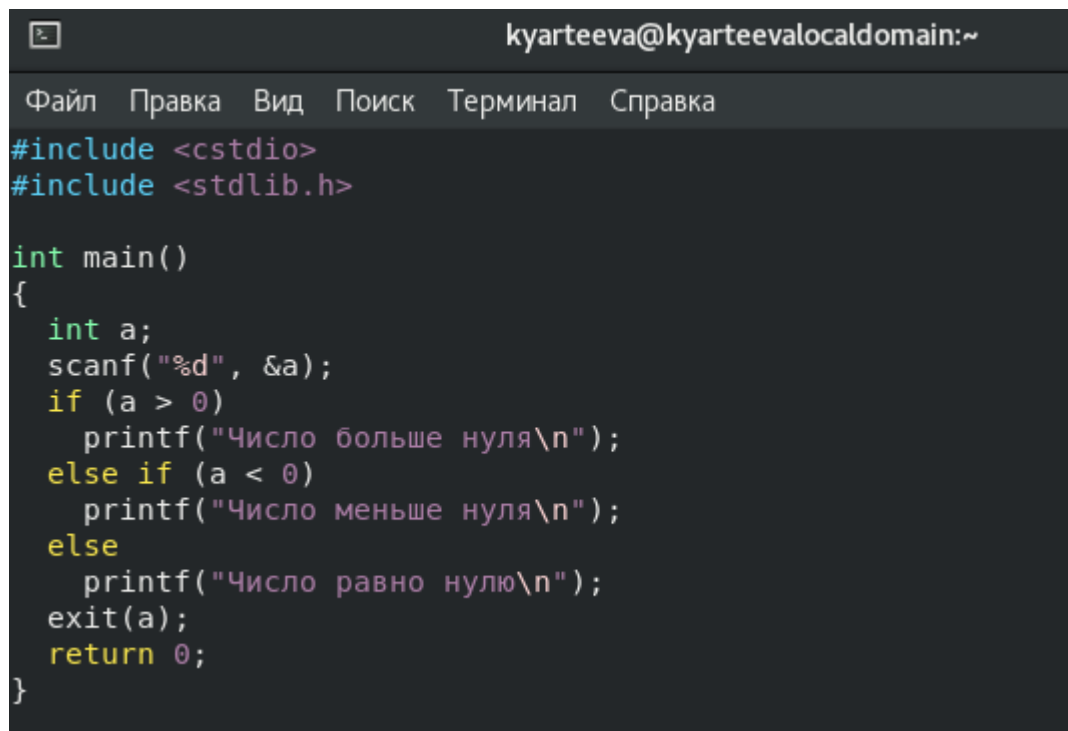
Рис.2 Анализ командной строки с ключами



The screenshot shows a C program with syntax highlighting. It implements the same logic as the shell script in Figure 2, using conditional operators and 'grep' to compare values. The program is enclosed in a 'main' function.

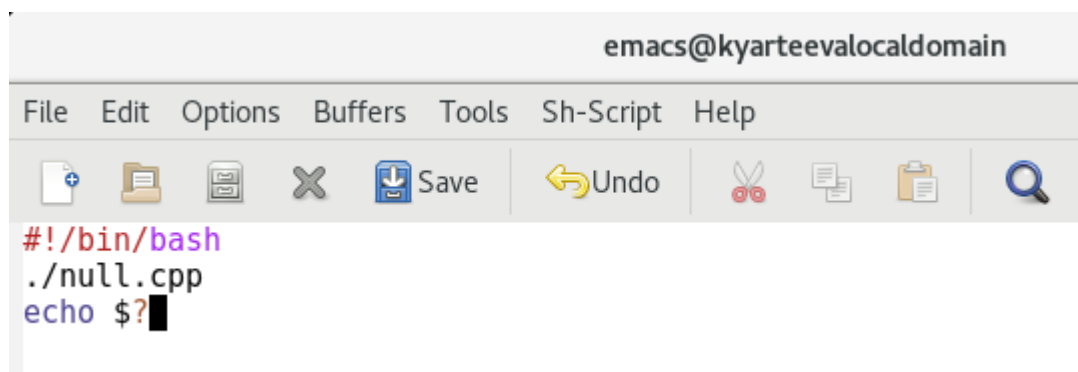
```
if (((Cflag==0)&&(nflag==1)))
then grep -e${pval} -n ${ival}
    if ((oflag==1))
    then grep -e${pval} -n ${ival} > ${oval}
    fi
fi
if (((Cflag==0)&&(nflag==0)))
then grep -e${pval} ${ival}
    if ((oflag==1))
    then grep -e${pval} ${ival} > ${oval}
    fi
fi
```

Рис.3 Программа на Си



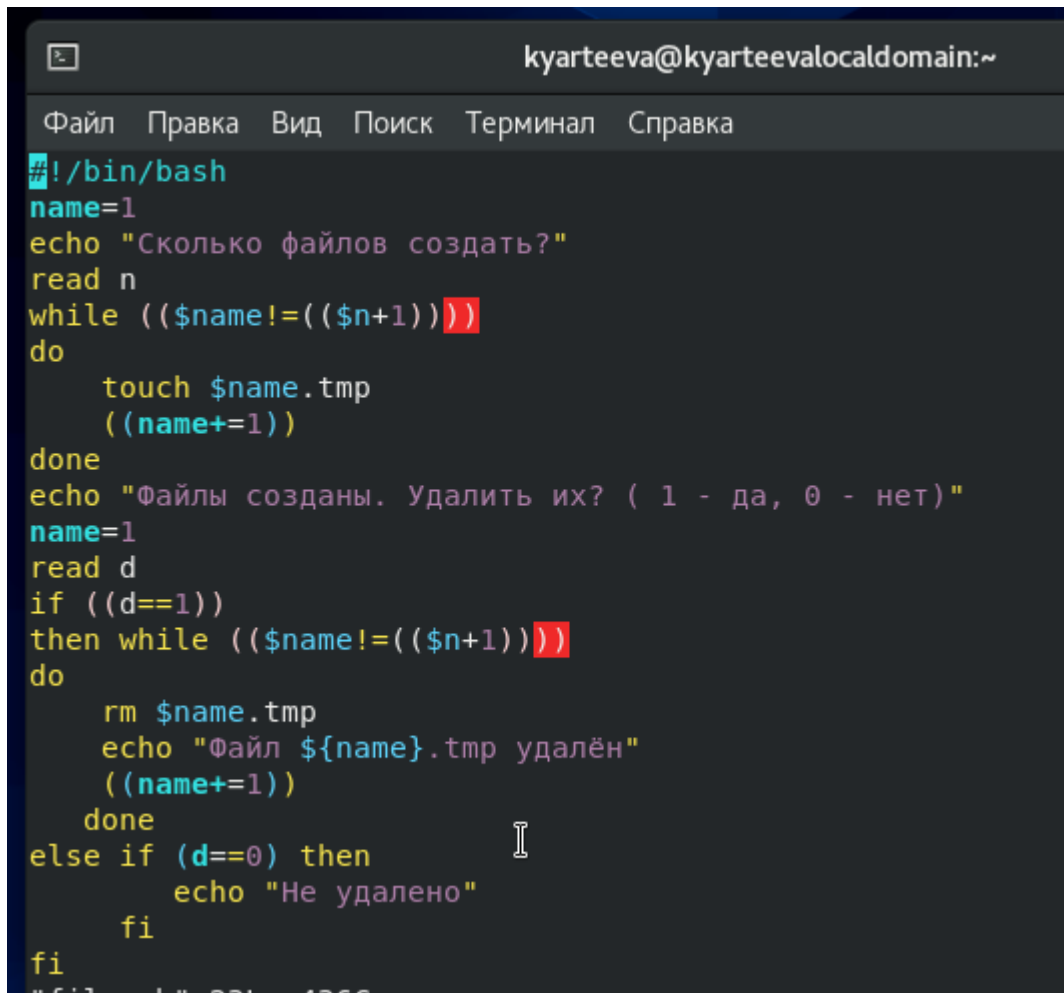
```
kyarteeva@kyarteevalocaldomain:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
#include <stdio>  
#include <stdlib.h>  
  
int main()  
{  
    int a;  
    scanf("%d", &a);  
    if (a > 0)  
        printf("Число больше нуля\n");  
    else if (a < 0)  
        printf("Число меньше нуля\n");  
    else  
        printf("Число равно нулю\n");  
    exit(a);  
    return 0;  
}
```

Рис.4 Командный файл



```
emacs@kyarteevalocaldomain  
File  Edit  Options  Buffers  Tools  Sh-Script  Help  
[Icons: New, Open, Save, Undo, Cut, Copy, Paste, Find]  
#!/bin/bash  
./null.cpp  
echo $?
```

Рис.5 Создание файлов

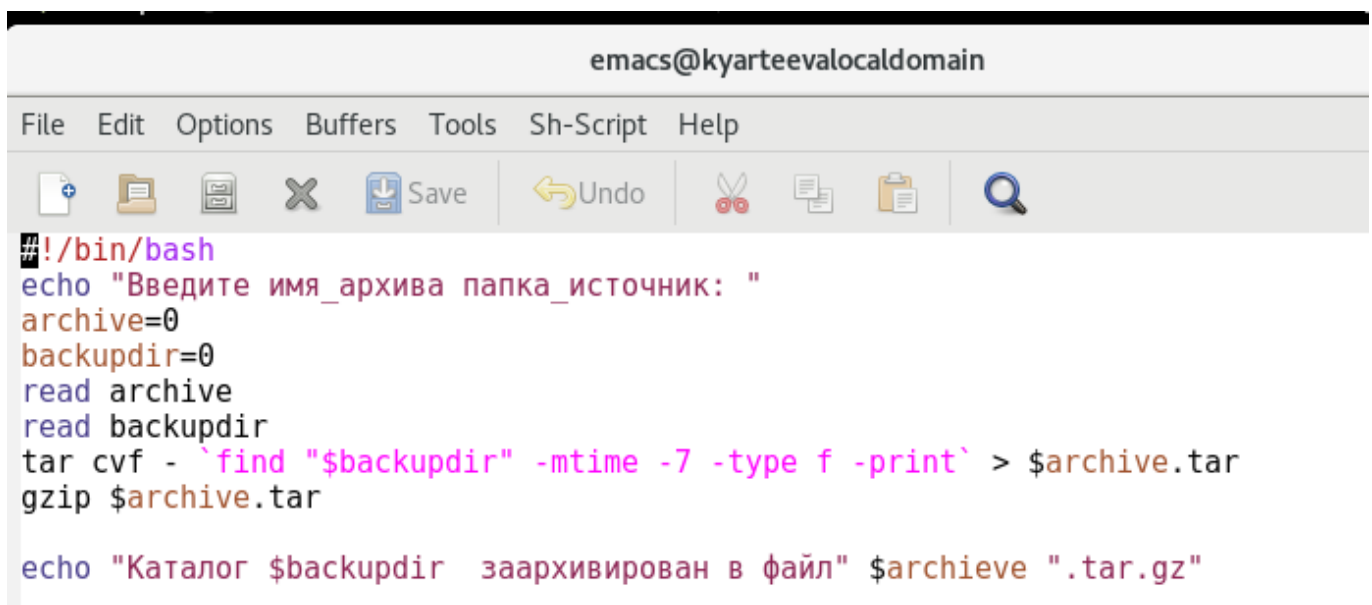


```

kyarteeva@kyarteevalocaldomain:~
Файл  Правка  Вид  Поиск  Терминал  Справка
#!/bin/bash
name=1
echo "Сколько файлов создать?"
read n
while (($name!=($n+1)))
do
    touch $name.tmp
    ((name+=1))
done
echo "Файлы созданы. Удалить их? ( 1 - да, 0 - нет)"
name=1
read d
if ((d==1))
then while (($name!=($n+1)))
do
    rm $name.tmp
    echo "Файл ${name}.tmp удалён"
    ((name+=1))
done
else if ((d==0)) then
    echo "Не удалено"
fi
fi

```

Рис.6 Архивация



```

emacs@kyarteevalocaldomain
File  Edit  Options  Buffers  Tools  Sh-Script  Help
[Icons: New, Open, Save, Undo, Cut, Copy, Paste, Find]
#!/bin/bash
echo "Введите имя_архива папка_источник: "
archive=0
backupdir=0
read archive
read backupdir
tar cvf - `find "$backupdir" -mtime -7 -type f -print` > $archive.tar
gzip $archive.tar

echo "Каталог $backupdir заархивирован в файл" $archive ".tar.gz"

```

Вывод

В ходе лабораторной работы я получила изучила основы программирования в оболочке ОС UNIX/Linux, научилась писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

