

Презентация по лабораторной работе №11

Шмырин Михаил Сергеевич

Российский Университет Дружбы Народов

Цель работы

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

Выполнение лабораторной работы

Задание 1

1. 1) Используя команды `grep`, написал 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. а затем ищет в указанном файле нужные строки, определяемые ключом `-r`. Для данной задачи я создал файл `progra1.sh` и написал соответствующий скрипт (рис. 1) , (рис. 2)

```
#!/bin/bash
iflag=0; oflag=0; pflag=0; Cflag=0; nflag=0;
while getopts i:op: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 "Даблон не найден"
    exit
fi
if ((iflag==0))
then
    echo "Входящий файл не найден"
    exit
fi
if ((oflag==0))
then
    echo "Исходящий файл не найден"
    exit
fi
```

Figure 1: Скрипт

```
if ((nflag==0))
then
    if ((iflag==0))
    then
        grep $pval $ival > $oval
    else
        grep -i $pval $ival > $oval
    fi
else
    if ((iflag==0))
    then
        grep -n $pval $ival > $oval
    else
        grep -i -n $pval $ival > $oval
    fi
fi
```

Figure 2: Скрипт

- 2) Проверил работу написанного скрипта, используя различные опции (например команду `./progra1.sh -i a1.txt -o a2.txt -C -n`), предварительно добавив право на исполнение файла (`chmod +x progra1.sh`) и создав 2 файла, которые необходимы для выполнения программы (`a1.txt`, `a2.txt`). Скрипт работает корректно.

2. 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 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 создал 2 файла: `chslo.c` `chislo.sh` и написал соответствующие скрипты (рис. 3) (рис. 4)


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

~:*** chslo.sh All L10 (

Figure 3: Скрипт

```
#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);
    if (a==0) exit(2);
    return 0;
}
```

~:--- chslo.c All L11

Figure 4: Скрипт

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

3. 1) Написал командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до (например 1.tmp, 2.tmp, 3.tmp, 4.tmp и т.д.). Число файлов, которые необходимо создать, передается в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют). Для данной задачи я создал файл files.sh и написал соответствующий скрипт (рис. 5)

```
file Edit Options Buffers Tools Sh-Script 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
```

Figure 5: Скрипт

- 2) Далее я проверил работу написанного скрипта (`./files.sh`), предварительно добавив право на исполнение файла. Сначала я создал три файла, удовлетворяющих условию задач, а потом удалил их. Скрипт работает корректно (рис. ??)

Проверка работы

4. 1) 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 (использовать команду `find`). Для данной задачи я содал файл `pr4.sh` и написал соответствующий скрипт (рис. 6)

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

Figure 6: Скрипт

- 2) Далее я проверил работу написанного скрипта, предварительно добавив право на исполнение файла и создав отдельный каталог с несколькими файлами

Выводы

В ходе выполнения лабораторной работы я изучил основы программирования в оболочке ОС UNIX и научился писать более сложные командные файлы с использованием логических управляющих конструкций циклов.