

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12 «Программирование в командном процессоре ОС UNIX. Ветвления и циклы»

дисциплина: *Операционные системы*

Студент: Саргсян Арам Грачьевич

Группа: НПИбд-02-20

МОСКВА 2021 г.

Цель работы: Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

Ход работы

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

`-i` — прочитать данные из указанного файла;

`-o` — вывести данные в указанный файл;

`-r` — указать шаблон для поиска;

`-C` — различать большие и малые буквы (по умолчанию включен)

`-n` — выдавать номера строк. (Рис.1-2)

```
#!/bin/bash

l_=""
o_=""
p_=""
C_=0
n_=0

while getopts "l:o:p:C:n" opt
do
case $opt in
    l)l_="$OPTARG";;
    o)o_="$OPTARG";;
    p)p_="$OPTARG";;
    C)C_=$((C_+1));;
    n)n_=$((n_+1));;
    esac
done
if (($C_ + $n_ == 2))
then
grep -l -n "$p_" "$l_" > "$o_"
elif (($C_ == 1))
then
grep -n "$p_" "$l_" > "$o_"
fi
```

рис. 1

```
agsargsyan@agsargsyan-VirtualBox:~/lab12$ .
C
agsargsyan@agsargsyan-VirtualBox:~/lab12$ c
abc
```

рис. 2

2. 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. (Рис. 3-4)

```
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
int main(){
    int j;
    scanf("%d > 0 \n", j);
    if (j>0) cout<<j<<">0">>endl;
    else if (j<0) cout<<j<<"<0"<<endl;
    else printf("0 = 0\n");
    exit(j);
    return j;
}
```

рис. 3

```
#!/bin/bash

./sch2.sh
echo $?
```

рис. 4

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 должен уметь удалять все созданные им файлы (если они существуют).

```
#!/bin/bash
r_=0
n_=0
name=1
while getopts "r:n:" opt
do
case opt in
r)r_="$OPTARG";;
n)n_="$OPTARG";;
esac
done
if [ "$r_" -eq "0" ]
then
while (($name!=($n_+1)))
do
for i in $name
do
touch $i.tmp
done
((name+=1))
done
else
name=1
while (($name!=($n_+1)))
do
for i in $name
do
rm $i.tmp
done
((name+=1))
done
fi
```

(Рис. 5-6)

рис. 5

```

agsargsyan@agsargsyan-VirtualBox:~/lab12$ ls
j      s1.sh~  s2.sh~  sch1.sh~ sch2.sh~ sch3.sh
s1.sh  s2.sh  sch1.sh sch2.sh  sch3.sh soldado
agsargsyan@agsargsyan-VirtualBox:~/lab12$ ./sch3.sh
agsargsyan@agsargsyan-VirtualBox:~/lab12$ ls
j      s1.sh~  s2.sh~  sch1.sh~ sch2.sh~ sch3.sh
s1.sh  s2.sh  sch1.sh sch2.sh  sch3.sh soldado
agsargsyan@agsargsyan-VirtualBox:~/lab12$

```

рис. 6

4. Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад. (Рис. 7-8)

```

emacs@agsargsyan-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
+ Save Undo
#!/bin/bash
arh=0
back=0
read arh
read back
if [ -e $arh ]
then
rm $arh
fi
find "$back" -maxdepth -ctime -7 -type f -print0 |
> /dev/null
gzip $arh.tar

```

рис. 7

```
agsargsyan@agsargsyan-VirtualBox:~/lab12$ ls
j      s1.sh~  s2.sh~  sch1.sh~ sch2.sh~ sch3.sh
s1.sh  s2.sh   sch1.sh sch2.sh  sch3.sh  sch4.sh
agsargsyan@agsargsyan-VirtualBox:~/lab12$ chmod u+x
agsargsyan@agsargsyan-VirtualBox:~/lab12$ ./sch4.sh
aaaaaaaaaacccccc

find: Expected a positive decimal integer argument
e'
gzip: aaaaaaaaaaacccccc.tar: No such file or directory
agsargsyan@agsargsyan-VirtualBox:~/lab12$ ls
arh.tar  s1.sh  s2.sh  sch1.sh  sch2.sh  sch3.s
j      s1.sh~  s2.sh~  sch1.sh~ sch2.sh~ sch3.s
agsargsyan@agsargsyan-VirtualBox:~/lab12$
```

рис. 8

ВЫВОД: В ходе работы мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командные файлы с использованием логических управляющих инструкций и циклов.