

Отчёт по лабораторной работе №12

**Программирование в командном процессоре ОС UNIX. Ветвления и
циклы**

Николаев Дмитрий Иванович

Содержание

1	Цель работы	3
2	Выполнение лабораторной работы	4
2.1	Контрольные вопросы	8
3	Выводы	10

1 Цель работы

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

2 Выполнение лабораторной работы

- 1) Написал командный файл, анализирующий командную строку вместе с ключами (на примере случайного текста).

```
#!/bin/bash
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 " Wrong input" $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
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))
```

- Командный

файл, анализирующий строку вместе с ключами

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

- Продолжение это-

го командного файла

```
[dinikolaev@dinikolaev ~]$ touch lab11.txt
[dinikolaev@dinikolaev ~]$ mcedit lab11.txt

[dinikolaev@dinikolaev ~]$ ./lab11.sh -ilab11.txt -olab11_1.txt -ppразнообразный
1:Разнообразный и богатый опыт начало повседневной работы по формированию позиции позво
ляет выполнить важнейшие задания по разработке системы обучения кадров, соответствующей
насущным потребностям.
3:Разнообразный и богатый опыт повышение уровня гражданского сознания требует определен
ия и уточнения модели развития!
7:Разнообразный и богатый опыт постоянный количественный рост и сфера нашей активности
играет важную роль в формировании модели развития.
```

- Результат выполнения командного файла на примере случайно сгенерированного текста

- 2) Написал на языке C++ программу, которая производит сравнение введённого числа относительно нуля, а командный файл должен вызывать эту программу и выдать сообщение о введённом.

```
#include <iostream>
int main()
{
    int x;
    std::cin >> x;
    if (x == 0) exit(0);
    else if (x < 0) exit(1);
    else if (x > 0) exit(2);
    return(3);
}
```

- Программа сравнения числа с ну-

лём на C++

```
#!/bin/bash
g++ -o cprog lab11_2.c
./cprog
case $? in
    0) echo "Number is equal to 0";;
    1) echo "Number is smaller than 0";;
    2) echo "Number is bigger than 0";;
esac
```

- Командный файл, вы-

зывающий программу и выводящий сообщение о введённом

```

[dinikolaev@dinikolaev ~]$ touch lab11_2.c
[dinikolaev@dinikolaev ~]$ touch lab11_2.sh
[dinikolaev@dinikolaev ~]$ chmod +x lab11_2.sh
[dinikolaev@dinikolaev ~]$ vi lab11_2.sh
[dinikolaev@dinikolaev ~]$ ./lab11_2.sh
0
Number is equal to 0
[dinikolaev@dinikolaev ~]$ ./lab11_2.sh
15
Number is bigger than 0
[dinikolaev@dinikolaev ~]$ ./lab11_2.sh
-10
Number is smaller than 0

```

- Результат выполне-

ния командного файла

- 3) Написал командный файл, создающий указанное число пронумерованных файлов, а также способный удалять созданные им файлы (если они существуют).

```

#!/bin/bash
let dflag=0;
while getopts a:d optletter
do case $optletter in
    a) aflag=1;aval=$OPTARG;;
    d) dflag=1;;
    *) echo " Wrong input" $optletter
    esac
done
#echo ${aval}
if ((dflag==0))
then for ((i=1; i<=aval; i++))
do touch ${i}.txt
done
fi
if ((dflag==1))
then for ((i=1; i<=aval; i++))
do rm ${i}.txt
done
fi

```

- Командный файл, создаю-

щий несколько пронумерованных файлов (и удаляющий их с другим флагом)

```
[dinikolaev@dinikolaev ~]$ vi lab11_3.sh
[dinikolaev@dinikolaev ~]$ ./lab11_3.sh -a6
[dinikolaev@dinikolaev ~]$ ls
1.txt  abc1      data.tar.gz  lab11_3.sh  ski.places  Музыка
2.txt  australia  debian-binary lab11.sh     work        Общедоступные
3.txt  bin        feathers     lab11.txt   Видео       'Рабочий стол'
4.txt  control.tar.gz lab11_1.txt  my_os      Документы   Шаблоны
5.txt  cprog      lab11_2.c   pandoc     Загрузки
6.txt  data       lab11_2.sh  play       Изображения
[dinikolaev@dinikolaev ~]$ ./lab11_3.sh -a6 -d
[dinikolaev@dinikolaev ~]$ ls
abc1      data.tar.gz  lab11_3.sh  ski.places  Музыка
australia  debian-binary lab11.sh     work        Общедоступные
bin        feathers     lab11.txt   Видео       'Рабочий стол'
control.tar.gz lab11_1.txt  my_os      Документы   Шаблоны
cprog      lab11_2.c   pandoc     Загрузки
data       lab11_2.sh  play       Изображения
```

- Результат выполнения командного файла (и удаление с флагом -d)

- 4) Написал командный файл, запаковывающий в архив с помощью команды tar все файлы указанной директории. И написал его модификацию для запаковывания только тех файлов, что были изменены менее недели назад с помощью команды find (модификация - последняя строка командного файла).

```
#!/bin/bash
tar -cf lab11_4_archive.tar $@
tar -cf lab11_4_modification.tar $@
find $@ -mtime -7 -exec tar -rf lab11_4_modification.tar {} \;
```

Командный файл, архивирующий введённый каталог (модификация - последняя строка)

```
[dinikolaev@dinikolaev ~]$ vi lab11_4.sh
[dinikolaev@dinikolaev ~]$ ./lab11_4.sh work
tar: Робкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
^C
[dinikolaev@dinikolaev ~]$ ./lab11_4.sh Изображения
tar: Робкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
[dinikolaev@dinikolaev ~]$ ./lab11_4.sh Изображения
tar: Робкий отказ от создания пустого архива
Попробуйте «tar --help» или «tar --usage» для
получения более подробного описания.
[dinikolaev@dinikolaev ~]$ ls
abcl          feathers      lab11.sh      Документы
australia     lab11_1.txt  lab11.txt     Загрузки
bin           lab11_2.c   my_os         Изображения
control.tar.gz lab11_2.sh   pandoc        Музыка
cprog         lab11_3.sh   play          Общедоступные
data          lab11_4_archive.tar ski.places    'Рабочий стол'
data.tar.gz   lab11_4_modification.tar work          Шаблоны
debian-binary lab11_4.sh   Видео
```

- Результат выполнения командного файла на примере каталогов work и Изображения

2.1 Контрольные вопросы

1. getopt - встроенная команда оболочки bash, предназначенной для разбора параметров сценариев. Обрабатываются только однобуквенные параметры.
2. При генерации имён применяются следующие метасимволы:
 - "*" - соответствует произвольной, в том числе и пустой строке;
 - "?" - соответствует одному любому символу;
 - "[c1-c2]" - соответствует любому символу между символами c1 и c2.
3. Операторы ";" (последовательное выполнение команд), "&&" (выполнение только после успешного выполнения предыдущей), "||" (выполнение только после ошибочного выполнения предыдущей). А также условные операторы (case, for, if, while).

4. Для прерывания используется оператор `break` для прерывания выполнения ближайшего включающего цикла или условного оператора и команда `continue`, которая завершает данную итерацию цикла.
5. Команда `true` возвращает ноль на выходе, что обозначает успешное выполнение. Команда `false` возвращает не ноль (часто 1), что обозначает неудачу. (Другими словами 0 - истина, другое значение - ложь).
6. Данная строка означает условие существования файла `mans/i.$s`, где `$` - означает какой-либо один символ.
7. Цикл `while` выполняется пока указанное условие истинно (когда ложно, то цикл завершается), а `until` выполняется пока условие ложно.

3 Выводы

Изучил основы программирования в оболочке ОС UNIX, освоил написание более сложных командных файлов, используя ещё логические управляющие конструкции и циклы.