

Лабораторная работа-11

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

Овчинников Данил НБИбд-03-22

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	11
4	Ответы на контрольные вопросы:	12
	Список литературы	13

Список иллюстраций

2.1	Команды1	6
2.2	Код1	7
2.3	Команды2	8
2.4	Код2	8
2.5	Команды3	9
2.6	Код3	9
2.7	Команды4	10
2.8	Код4	10

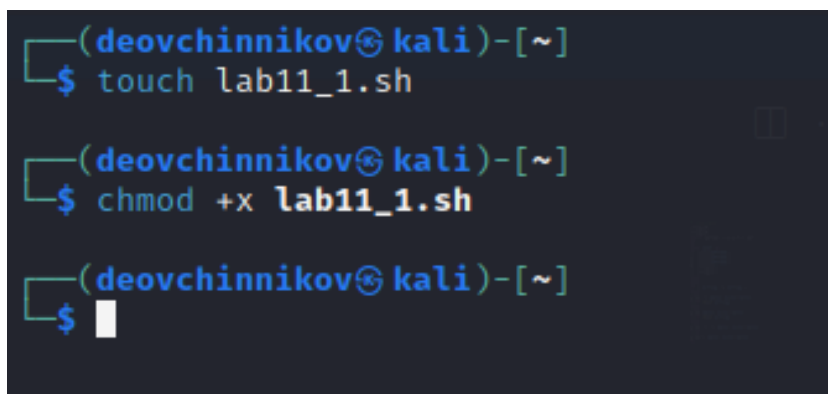
Список таблиц

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

Используя команды `getopts` `grep`, написать командный файл, который анализирует командную строку с ключами: `-iinputfile` — прочитать данные из указанного файла; `-ooutputfile` — вывести данные в указанный файл; `-р` — 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 `-p`. (рис. 2.1 2.2)



```
(deovchinnikov@kali)-[~]  
$ touch lab11_1.sh  
  
(deovchinnikov@kali)-[~]  
$ chmod +x lab11_1.sh  
  
(deovchinnikov@kali)-[~]  
$
```

Рис. 2.1: Команды1

```
$ lab11_1.sh x presentation.md
home > deovchinnikov > $ lab11_1.sh
1  #!/bin/bash
2  cflag=0;
3  nflag=0;
4  while getopts i:o:p:C:n opt
5  do
6  case $opt in
7  i) ival=$OPTARG;;
8  o) ival=$OPTARG;;
9  p) ival=$OPTARG;;
10 C) cflag=1;;
11 n) nflag=1;;
12 esac
13 done
14 if [ $cflag -a $nflag ]
15 then
16 grep -n $pval $ival>$oval
17 elif test $cflag
18 then
19 grep $pval $ival>$oval
20 elif test $cflag
21 then
22 grep -n -i $pval $ival>$oval
23 else
24 grep -i $pval $ival>$oval
25 fi
```

Рис. 2.2: Код1

Написать 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(рис. 2.3 2.4).

```
(deovchinnikov@kali)-[~]  
$ touch lab11_2.sh  
  
(deovchinnikov@kali)-[~]  
$ chmod +x lab11_2.sh  
  
(deovchinnikov@kali)-[~]  
$
```

Рис. 2.3: Команды2

```
$ lab11_2.sh presentation.md  
home > deovchinnikov > $ lab11_2.sh  
1  #!/bin/bash  
2  gcc -c script2.c  
3  gcc -o script2 script2.c  
4  ./script2  
5  case $? in  
6  1) echo отрицательное;;  
7  2) echo равно нулю;;  
8  3) echo положительное;;  
9  esac
```

Рис. 2.4: Код2

Написать 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, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют). (рис. 2.5 2.6).


```

(deovchinnikov@kali)-[~]
$ touch lab11_3.sh

(deovchinnikov@kali)-[~]
$ chmod +x lab11_3.sh

```

Рис. 2.5: Команды

```

$ lab11_2.sh ● $ lab11_3.sh ● presentation.md ●
home > deovchinnikov > $ lab11_3.sh
1  #!/bin/bash
2  let i=$1+1
3  while (( i-=1 ))
4  do touch $i.tmp
5  done
6  let j=$2+1
7  while (( j-=1 ))
8  do rm $j.tmp
9  done

```

Рис. 2.6: Код

Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).(рис. 2.7 2.8)

```
(deovchinnikov@kali)-[~]  
$ touch lab11_4.sh  
  
(deovchinnikov@kali)-[~]  
$ chmod +x lab11_4.sh
```

Рис. 2.7: Команды4

```
$ lab11_2.sh • $ lab11_3.sh • $ lab11_4.sh • presentation.md •  
home > deovchinnikov > $ lab11_4.sh  
1  #!/bin/bash  
2  (find $1 -mtime -7 -daystart) | xargs tar -cf archiv.tar
```

Рис. 2.8: Код4

3 Выводы

Я изучил основы программирования в оболочке ОС UNIX/Linux. Научился писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

4 Ответы на контрольные вопросы:

1. Каково предназначение команды `getopts`? Ответ: Создание по пользовательским аргументам.
2. Какое отношение метасимволы имеют к генерации имён файлов? Ответ: Используют как файлы так и аргументы.
3. Какие операторы управления действиями вы знаете? Ответ: `if`, `else`, `elif`, `fi`, `while`, `do`, `done`, `until`, `do`, `done`, `for`, `in`, `do`, `done`, `case`, `in`, `esac`
4. Какие операторы используются для прерывания цикла? Ответ:
 - a) `for` – будет выполнять действие до тех пор, пока есть объекты для выполнения.
 - b) `while` – выполняет действие до тех пор, пока условие является истинным.
 - c) `until` – будет выполняться пока условие не станет правдиво.
5. Для чего нужны команды `false` и `true`? Ответ: `until` – будет выполняться до тех пор, пока условие не станет `true`, т.е. пока оно не станет `false`.
6. Что означает строка `if test -f man/i.$s`, встреченная в командном файле? Ответ: Проверяет если существует файл его размерность и тип с двумя разными расширениями, заменяя через переменные. 12
7. Объясните различия между конструкциями `while` и `until`. Ответ: `while` – выполняет действие до тех пор, пока условие является истинным. `until` – будет выполняться до тех пор, пока условие не станет истинным, т.е. пока оно `false`.

Список литературы