

# Российский Университет Дружбы Народов

## Факультет Физико-Математических и Естественных Наук

### ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10

дисциплина: *Операционные системы*

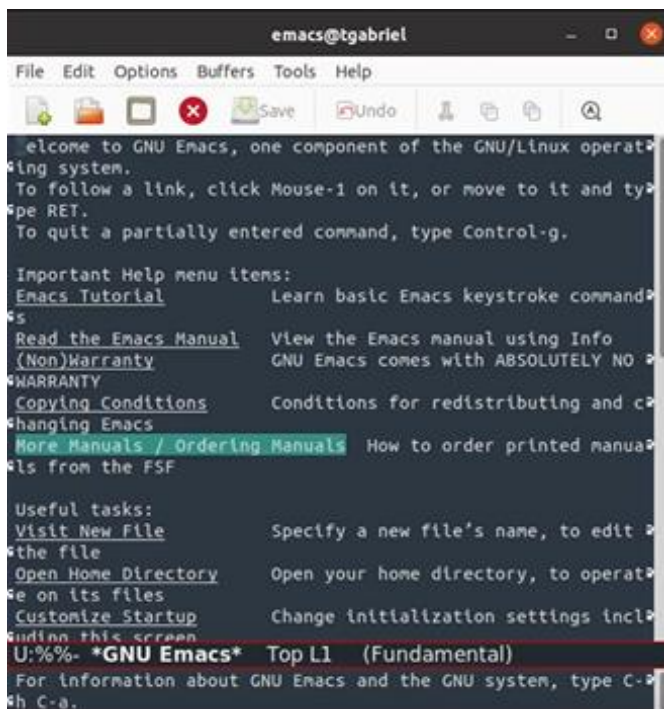
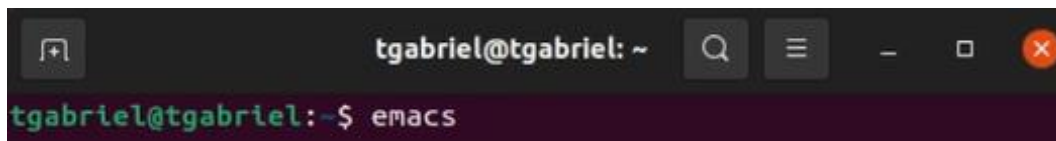
Студент: ГАБРИЭЛЬ ТЬЕРРИ Группа: НКНбд 01-20

МОСКВА 2021 г.

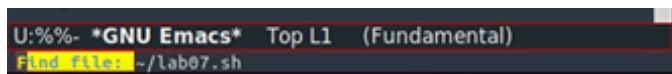
**Цель работы:** >Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs

#### Ход работы:

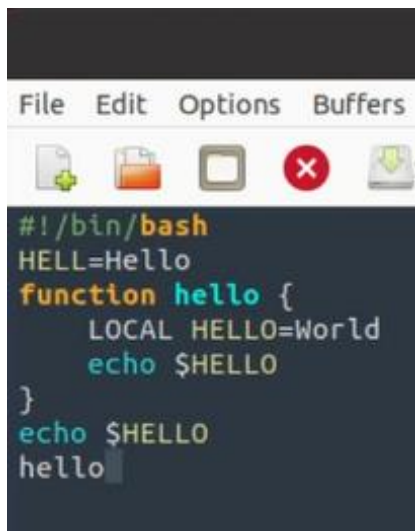
2. Откроем редактор Emacs с помощью команды «emacs &» (Рисунки 1,2)



3. Создадим файл lab07.sh с помощью комбинации «Ctrl-x» «Ctrl-f». (Рисунок 3)

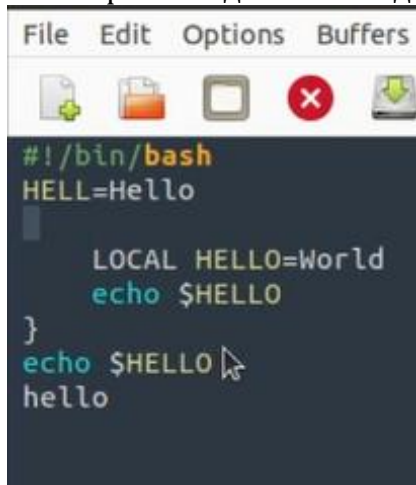


4. В открывшемся буфере наберем необходимый текст (Рисунок 4)



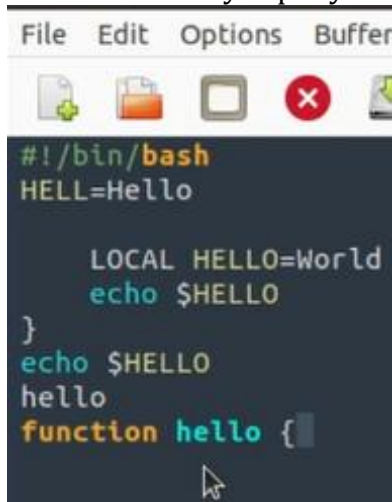
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

5. Сохраним файл с помощью комбинации «Ctrl-x» «Ctrl-s»
- 6.1. Вырежем одной командой целую строку («Ctrl-k») (Рисунок 5)



```
#!/bin/bash
HELL=Hello
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
```

- 6.2. Вставим эту строку в конец файла («Ctrl-y») (Рисунок 6)



```
#!/bin/bash
HELL=Hello
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
function hello {
```

- 6.3. Выделим область текста («Ctrl-space») (Рисунок 7)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
```

- 6.4. Скопируем область в буфер обмена («Alt-w»).
- 6.5. Вставим область в конец файла («Ctrl-y») (Рисунок 8)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 6.6. Вновь выделим эту область («Ctrl-space») (Рисунок 9) и на этот раз вырежем её («Ctrl-w»)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
hello
function hello {
echo $HELLO
```

- 6.7. Отменим последнее действие («Ctrl-/») (Рисунок 10)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 7.1. Переместим курсор в начало строки («Ctrl-a») (Рисунок 11)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 7.2. Переместим курсор в конец строки («Ctrl-e») (Рисунок 12)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 7.3. Переместим курсор в начало буфера («Alt-<») (Рисунок 13)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 7.4. Переместим курсор в конец буфера («Alt->») (Рисунок 14)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 8.1. Выведем список активных буферов на экран («Ctrl-x» «Ctrl-b») (Рисунок

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

CRM	Buffer	Size	Mode	File
U:***-	lab07.sh	All L10	(Shell-script[sh])	
1	* lab07.sh	110	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	920	Fundamental	
	*scratch*	145	Lisp Interaction	
%*	*Messages*	792	Messages	

15)

- 8.2. Переместимся во вновь открытое окно («Ctrl-x o») со списком открытых буферов (Рисунок 16) и переключимся на другой буфер (для этого

необходимо нажать на «enter» после выбора необходимого буфера)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

U:\*\*\*- lab07.sh All L9 (Shell-script[sh])

CRM	Buffer	Size	Mode	File
.	* lab07.sh	110	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	920	Fundamental	
	*scratch*	145	Lisp Interaction	
***	*Messages*	792	Messages	

- 8.3. Закроем это окно («Ctrl-x 0») (Рисунок 17)

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

- 8.4.Теперь вновь переключимся между буферами, но уже без вывода их списка на экран («Ctrl-x b») (Рисунок 19)

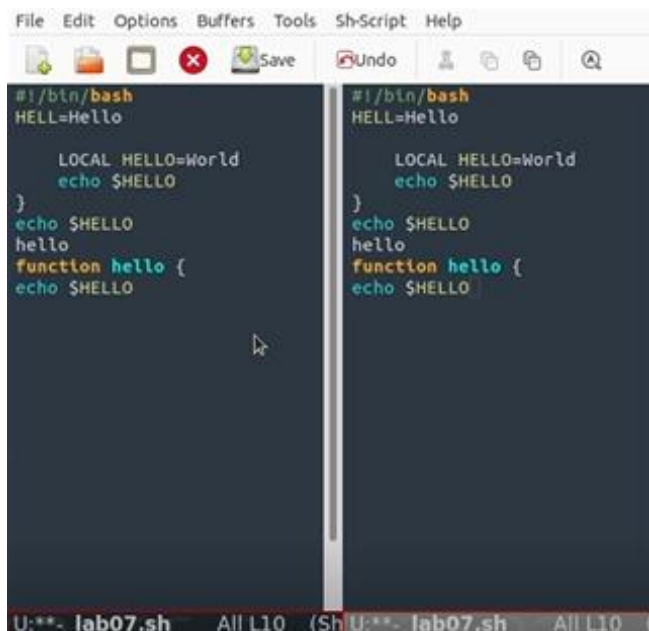
```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
```

U:\*\*\*- lab07.sh All L9 (Shell-script[sh])

Switch to buffer (default \*Buffer List\*): scratch\*

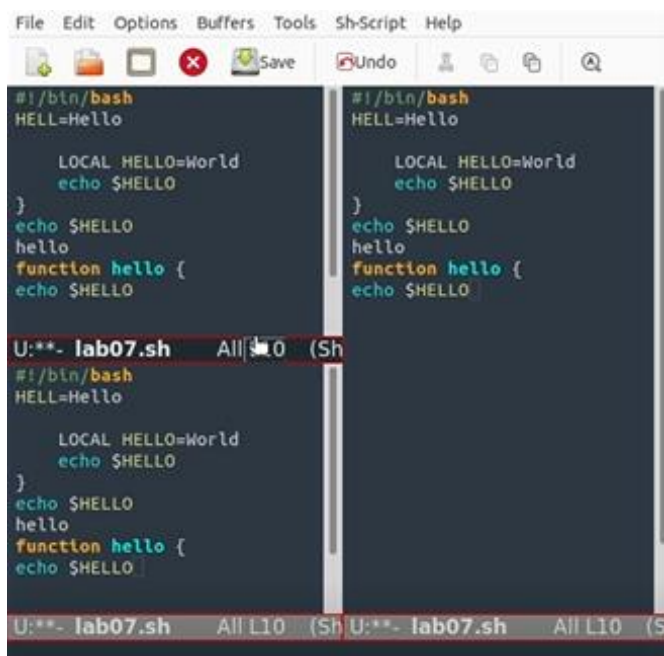
- 9.1. Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали («Ctrl-x 3»), а затем каждое из этих окон на две части по горизонтали («Ctrl-x 2») (Рисунки 20, 21, 22)



```
#!/bin/bash
HELL=Hello

LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
}
```

U:\*\*\*- lab07.sh All L10 (Sh U:\*\*\*- lab07.sh All L10



```
#!/bin/bash
HELL=Hello

LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
}
```

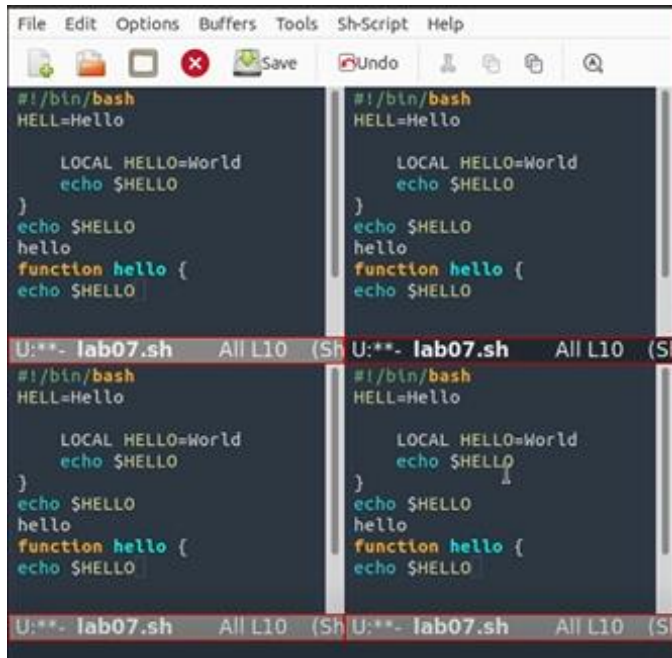
U:\*\*\*- lab07.sh All L10 (Sh

```
#!/bin/bash
HELL=Hello

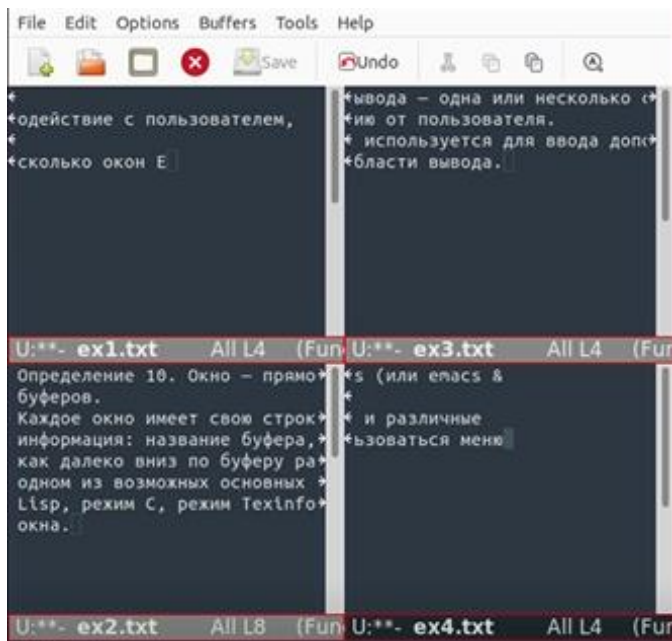
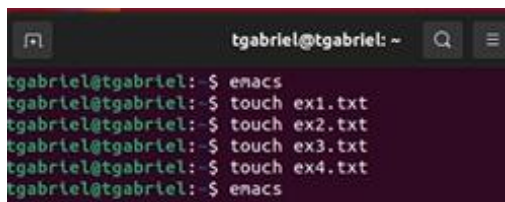
LOCAL HELLO=World
echo $HELLO
}
echo $HELLO
hello
function hello {
echo $HELLO
}
```

U:\*\*\*- lab07.sh All L10 (Sh U:\*\*\*- lab07.sh All L10 (S



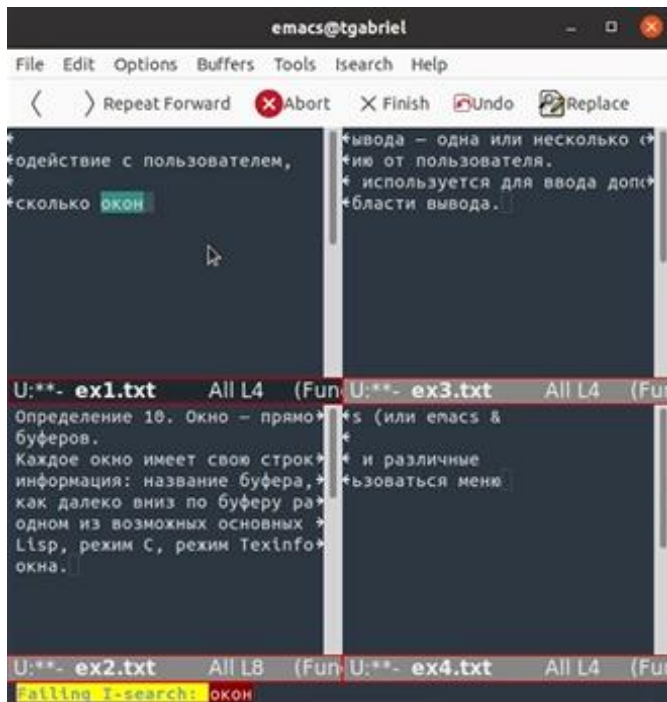


- 9.2. В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста. Для этого предварительно создадим эти файлы с помощью команд «touch ex1.txt», «touch ex2.txt», «touch ex3.txt», «touch ex4.txt» (Рисунки 23,24)



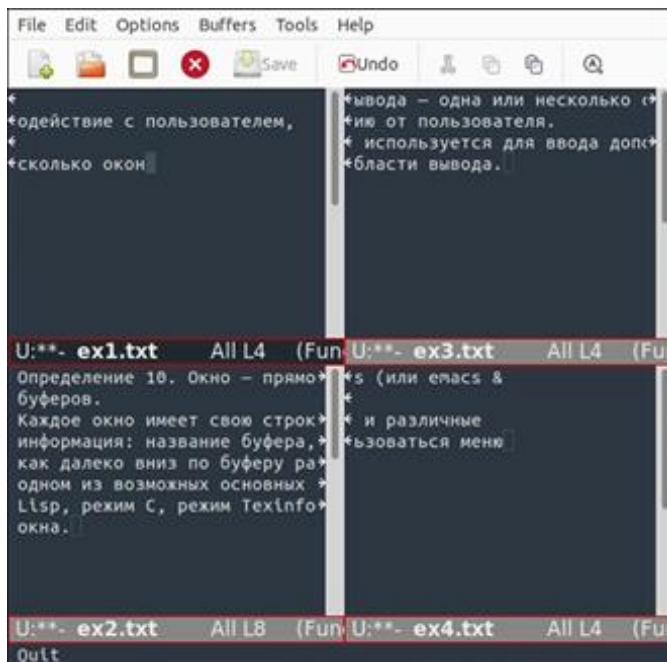


- 10.1. Переключимся в режим поиска («Ctrl-s») и найдем несколько слов, присутствующих в тексте (Рисунок 25)



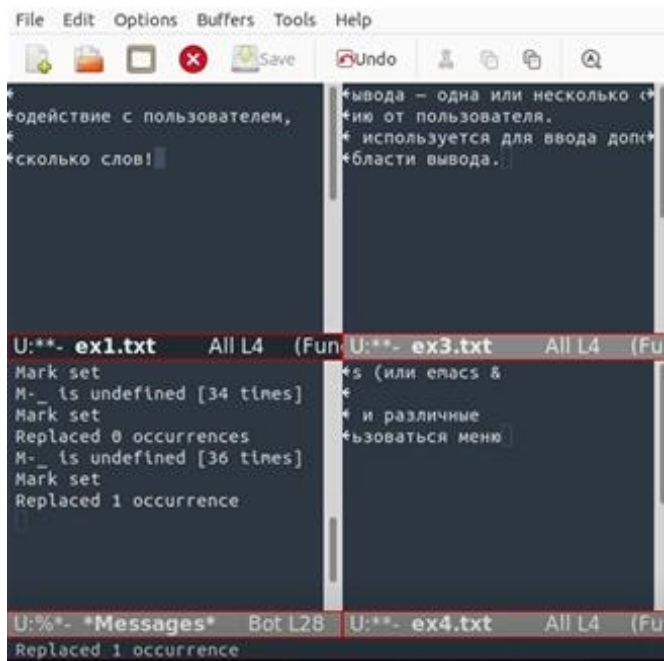
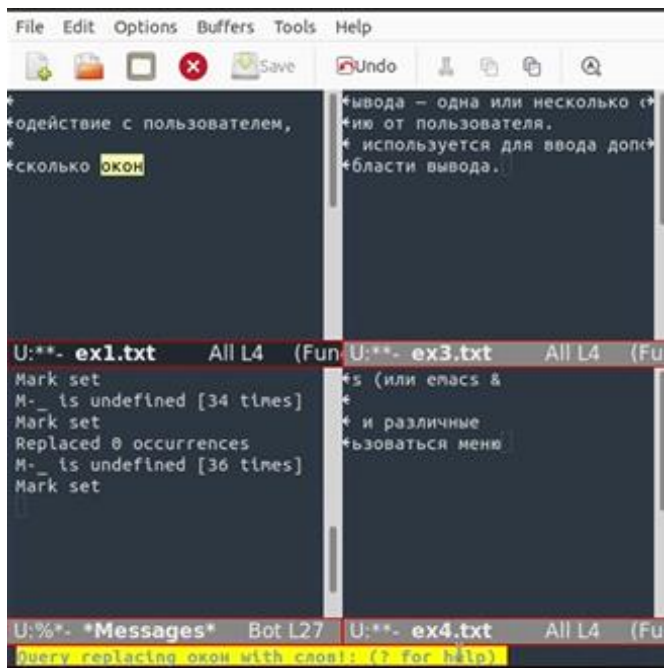
- 10.2. Переключимся между результатами поиска, нажимая «Ctrl-s»

-10.3. Выйдем из режима поиска, нажав «Ctrl-g» (Рисунок 26)

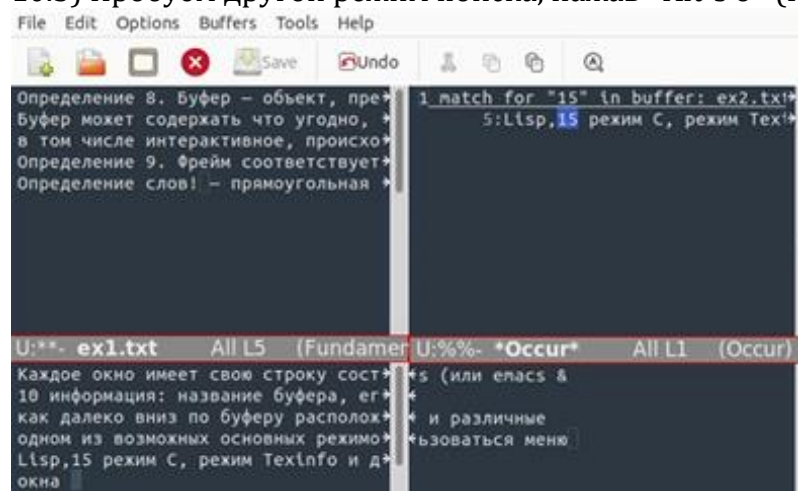


- 10.4. Перейдем в режим поиска и замены («Alt-%»), введем текст, который следует найти и заменить, нажмем «enter», затем введем текст для замены.

После того как будут подсвечены результаты поиска, нажмем «!» для подтверждения замены (Рисунки 27, 28). Важно, чтобы курсор находился в начале текста.



### 10.5) Пробуем другой режим поиска, нажав «Alt-s o» (Рисунок 29)



Данный вид поиска отличается от обычного тем, что тут считывается строка поиска, которая трактуется как регулярное выражение, и не осуществляется поиск точного совпадения в тексте буфера. Регулярное выражение – это образец, который обозначает набор строк, возможно, и неограниченный набор.

**Вывод:** >В ходе выполнения данной лабораторной работы я познакомился с операционной системой Linux и получил практические навыки работы с редактором Emacs.

### Контрольные вопросы:

1. Emacs – один из наиболее мощных и широко распространённых редакторов, используемых в мире Unix. По популярности он соперничает с редактором vi и его клонами. В зависимости от ситуации, Emacs может быть:
  - текстовым редактором;
  - программой для чтения почты и новостей Usenet;
  - интегрированной средой разработки (IDE);
  - операционной системой и т.д. Всё это разнообразие достигается благодаря архитектуре Emacs, которая позволяет расширять возможности редактора при помощи языка Emacs Lisp. На языке C написаны лишь самые базовые и низкоуровневые части Emacs, включая полнофункциональный интерпретатор языка Lisp. Таким образом, Emacs имеет встроенный язык программирования, который может использоваться для настройки, расширения и изменения поведения редактора. В действительности, большая часть того редактора, с которым пользователи Emacs работают в наши дни, написана на языке Lisp.
2. Основную трудность для новичков при освоении данного редактора могут составлять большое количество команд, комбинаций клавиш, которые не получится все запомнить с первого раза и поэтому придется часто обращаться к справочным материалам.

3. Буфер – это объект, представляющий собой текст. Если имеется несколько буферов, то редактировать можно только один. Обычно буфер считывает данные из файла или записывает в файл данные из буфера. Окно – это область экрана, отображающая буфер. При запуске редактора отображается одно окно, но при обращении к некоторым функциям могут открыться дополнительные окна. Окна Emacs и окна графической среды X Window – разные вещи. Одно окно X Window может быть разбито на несколько окон в смысле Emacs, в каждом из которых отображается отдельный буфер.
4. Да, можно.
5. При запуске Emacs по умолчанию создаются следующие буферы:
  - «scratch» (буфер для несохраненного текста)
  - «Messages» (журнал ошибок, включающий также информацию, которая появляется в области EchoArea)
  - «GNU Emacs» (справочный буфер о редакторе)
6. C-c | сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и нажимаю «|» C-c C-| сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и, удерживая «ctrl», нажимаю «|»
7. Чтобы поделить окно на две части необходимо воспользоваться комбинацией «Ctrl-x 3» (по вертикали) или «Ctrl-x 2» (по горизонтали).
8. Настройки Emacs хранятся в файле .emacs.
9. По умолчанию клавиша «←» удаляет символ перед курсором, но в редакторе её можно переназначить. Для этого необходимо изменить конфигурацию файла .emacs.
10. Более удобным я считаю редактор emacs, потому что в нем проще открывать другие файлы, можно использовать сразу несколько окон, нет «Командного режима», «Режима ввода», «Режима командной строки», которые являются немного непривычными и в какой-то степени неудобными