

Лабораторная работа N 9

Текстовый редактор emacs

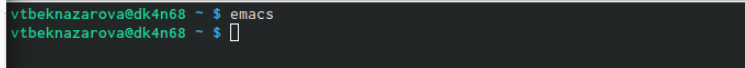
Бекназарова Виктория Тиграновна

06 апреля 2023

Российский университет дружбы народов, Москва, Россия

Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs

1. Открываем emacs



```
vtbknazarova@dk4n68 ~ $ emacs  
vtbknazarova@dk4n68 ~ $
```

Рис. 1: Команда в терминале

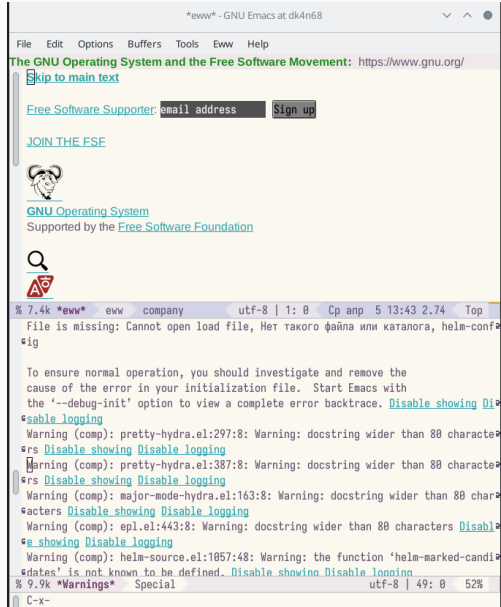
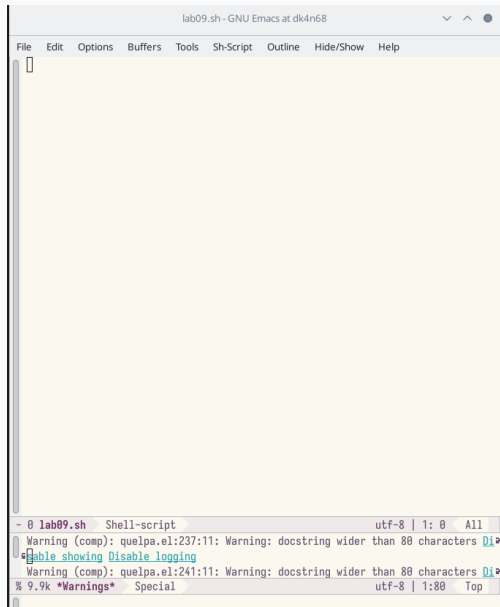


Рис. 2: Emacs

2. Создаем файл lab09.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).



lab09.sh - GNU Emacs at dk4n68

File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help

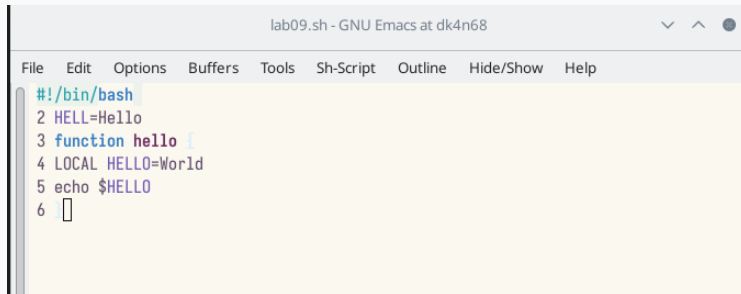
- 0 lab09.sh Shell-script utf-8 | 1: 0 All

Warning (comp): quelpa.el:237:11: Warning: docstring wider than 80 characters [Disable showing](#) [Disable logging](#)

Warning (comp): quelpa.el:241:11: Warning: docstring wider than 80 characters [Disable showing](#) [Disable logging](#)

% 9.9k *Warnings* Special utf-8 | 1:80 Top

3. Набираем текст

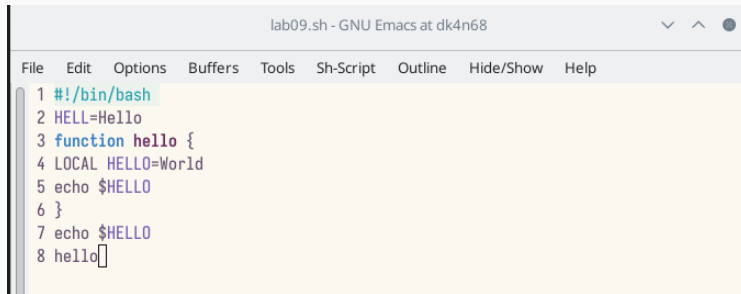
A screenshot of a GNU Emacs editor window. The title bar at the top reads "lab09.sh - GNU Emacs at dk4n68". Below the title bar is a menu bar with the following items: "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", "Outline", "Hide/Show", and "Help". The main editing area has a light yellow background and contains a shell script with the following lines:

```
#!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
```

The text is color-coded: the shebang is green, line numbers are purple, keywords like "function" and "echo" are blue, and variable names like "HELL" and "HELLO" are purple. The cursor is at the end of line 6.

Рис. 4: Набранный текст

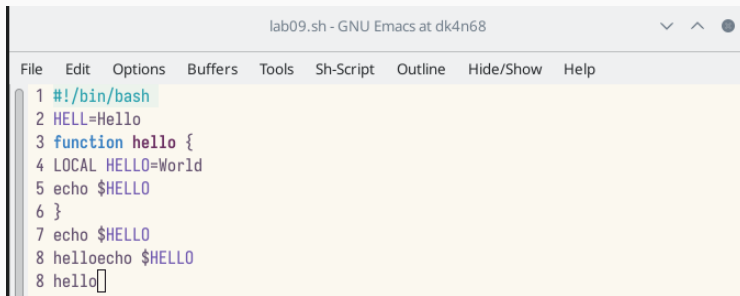
4. Сохраняем файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).



```
lab09.sh - GNU Emacs at dk4n68
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 hello
```

Рис. 5: Сохраняем файл

5. Прodelываем с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш. Вырезаем одной командой целую строку (C-k). Вставляем эту строку в конец файла (C-y). Выделяем область текста (C-space). Копируем область в буфер обмена (M-w). Вставляем область в конец файла. Вновь выделяем эту область и на этот раз вырезать её (C-w). Отменяем последнее действие (C-/).

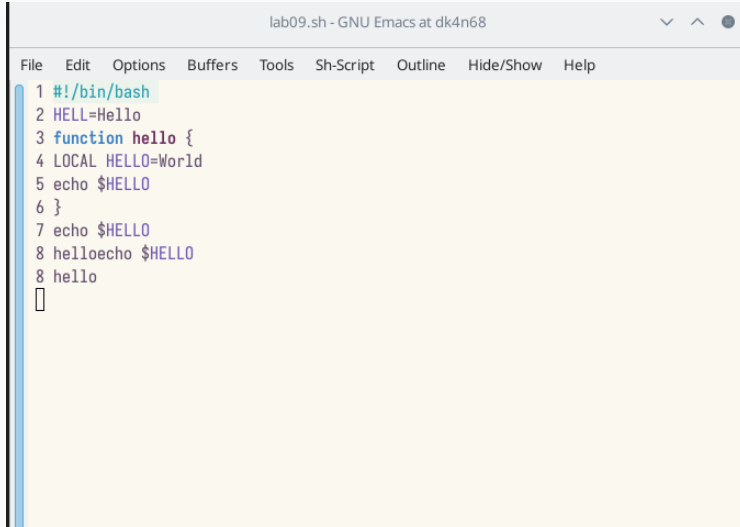


```
lab09.sh - GNU Emacs at dk4n68
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
1 #!/bin/bash
2 HELLO=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
```

Рис. 6: Команда C-k


```
lab09.sh - GNU Emacs at dk4n68
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4 LOCAL HELLO=World
5 echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4 LOCAL HELLO=World
5 echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
```

Рис. 7: Команда C-y



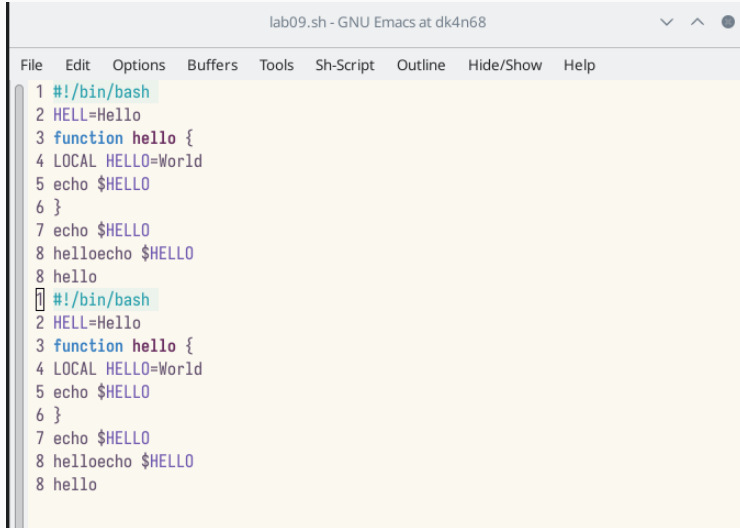
The image shows a screenshot of a GNU Emacs editor window. The title bar at the top reads "lab09.sh - GNU Emacs at dk4n68". Below the title bar is a menu bar with the following items: "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", "Outline", "Hide/Show", and "Help". The main editing area has a light yellow background and contains a shell script with the following lines:

```
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello

```

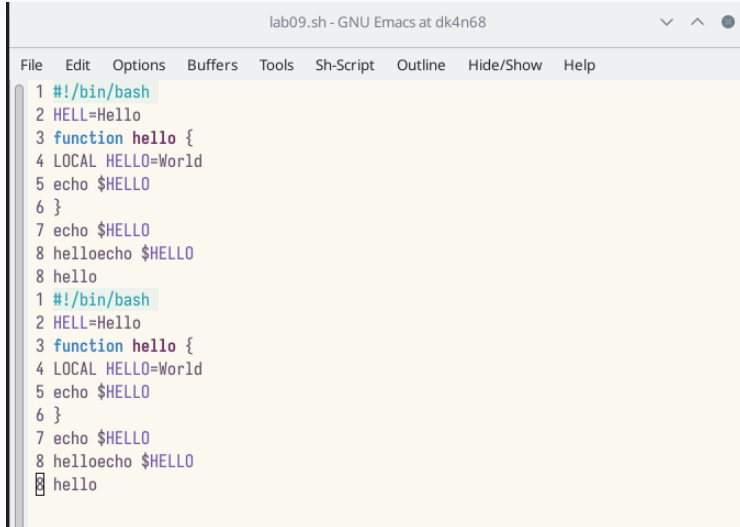
The cursor is positioned at the end of the last line, on the line "8 hello".

Рис. 8: Команда C-space



```
lab09.sh - GNU Emacs at dk4n68
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
```

Рис. 9: Команда M-w



lab09.sh - GNU Emacs at dk4n68

File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help

```
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
```

Рис. 10: Вставляем область в конец файла

The image shows a GNU Emacs window titled 'lab09.sh - GNU Emacs at dk4n68'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', 'Outline', 'Hide/Show', and 'Help'. The main text area contains a shell script with the following content:

```
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
```

The script defines a function named 'hello' that sets a local variable 'HELLO' to 'World' and echoes its value. It then echoes the global variable 'HELL' (which is 'Hello') and calls the 'hello' function twice. The first call is on line 8, and the second call is on line 8, with a cursor visible at the end of the line.

Рис. 11: Команда C-w

```
lab09.sh - GNU Emacs at dk4n68
File Edit Options Buffers Tools Sh-Script Isearch Outline Hide/Show Help
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4 LOCAL HELLO=World
5 echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4 LOCAL HELLO=World
5 echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO
8 hello
```

Рис. 12: Команда C-/

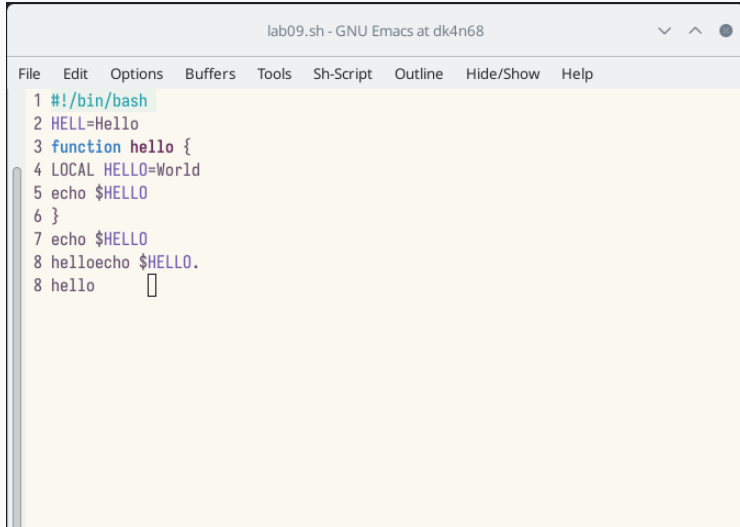
6. Научимся использовать команды по перемещению курсора. Перемещаем курсор в начало строки (C-a). Перемещаем курсор в конец строки (C-e). Перемещаем курсор в начало буфера (M-<). Перемещаем курсор в конец буфера (M->).

```
* 256 lab09.sh Shell-script utf-8 | 18:13 Bottom
```

CRM Buffer	Size	Mode	File
* lab09.sh	256	Shell-script[sh]	~/lab09.sh
%* *eww*	7437	eww	https://www.gnu.org/
% *GNU Emacs*	723	Fundamental	
scratch	145	Lisp Interaction	
%* *Messages*	68329	Messages	
%* *Compile-Log*	30039	Compilation	
%* *Async-native-compi...	598019	Fundamental	
%* *Native-compile-Log*	5500	LIMPLE	
* *quelpa-build-checkou...	4955	Fundamental	
%* *Warnings*	9932	Special	
% *Quail Completions*	0	Fundamental	
%* *package update results*	19	Fundamental	

```
* 256 lab09.sh Shell-script utf-8 | 18:13 Bottom  
;; This buffer is for text that is not saved, and for Lisp evaluation.  
;; To create a file, visit it with C-x C-f and enter text in its buffer.  
  
|
```

Рис. 14: Команда C-e

A screenshot of a GNU Emacs editor window. The title bar at the top reads "lab09.sh - GNU Emacs at dk4n68" and includes standard window controls (minimize, maximize, close). The menu bar below the title bar contains the following items: File, Edit, Options, Buffers, Tools, Sh-Script, Outline, Hide/Show, and Help. The main editing area has a light yellow background and contains a shell script with the following lines:

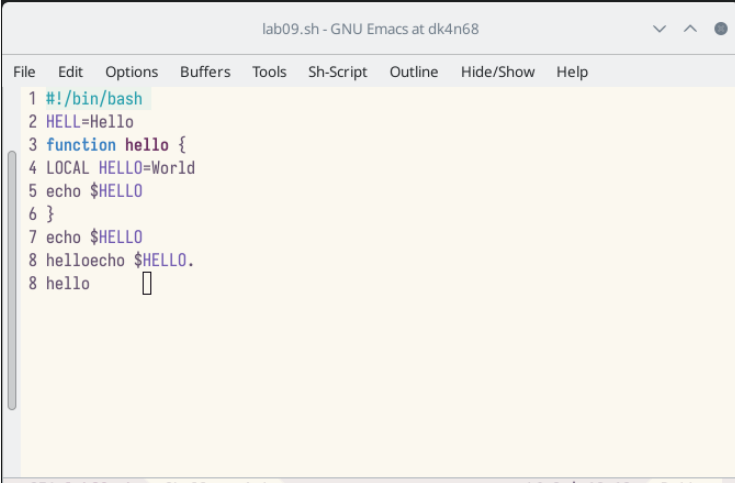
```
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO.
8 hello
```

The first line is highlighted in light blue. A vertical scrollbar is visible on the left side of the editing area, and a cursor is positioned at the end of the last line.

```
lab09.sh - GNU Emacs at dk4n68
File Edit Options Buffers Tools Sh-Script Outline Hide/Show Help
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO.
8 hello
```

Рис. 15: Команда M-<

7. Управление буферами. Выводим список активных буферов на экран (C-x C-b). Перемещаем во вновь открытое окно (C-x) о со списком открытых буферов и переключаемся на другой буфер. Закрываем это окно (C-x 0). Теперь вновь переключаемся между буферами, но уже без вывода их списка на экран (C-x b).



```
1 #!/bin/bash
2 HELL=Hello
3 function hello {
4   LOCAL HELLO=World
5   echo $HELLO
6 }
7 echo $HELLO
8 helloecho $HELLO.
8 hello
```

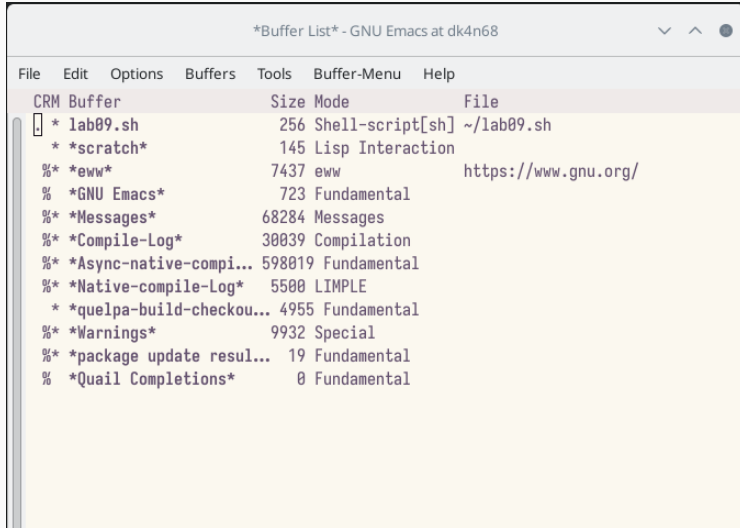


Рис. 17: Команда C-x

% 7.4k *eww* eww		utf-8 1: 0 Top	
<pre>1 #!/bin/bash 2 HELL=Hello 3 function hello { 4 LOCAL HELLO=World 5 echo \$HELLO 6 } 7 echo \$HELLO 8 helloecho \$HELLO. 8 hello</pre>		<pre>1 #!/bin/bash 2 HELL=Hello 3 function hello { 4 LOCAL HELLO=World 5 echo \$HELLO 6 } 7 echo \$HELLO 8 helloecho \$HELLO. 8 hello</pre>	
* 256 lab09.sh Bottom		* 256 lab09.sh Bottom	
<pre>1 #!/bin/bash 2 HELL=Hello 3 function hello { 4 LOCAL HELLO=World 5 echo \$HELLO 6 }</pre>		<pre>1 #!/bin/bash 2 HELL=Hello 3 function hello { 4 LOCAL HELLO=World 5 echo \$HELLO 6 }</pre>	

Рис. 18: Команда C-x 0

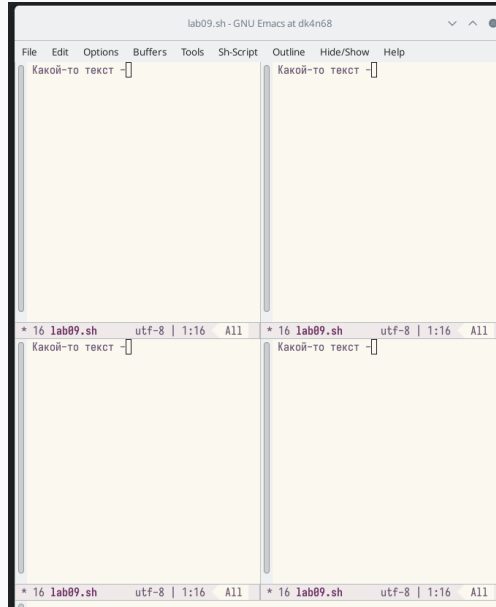
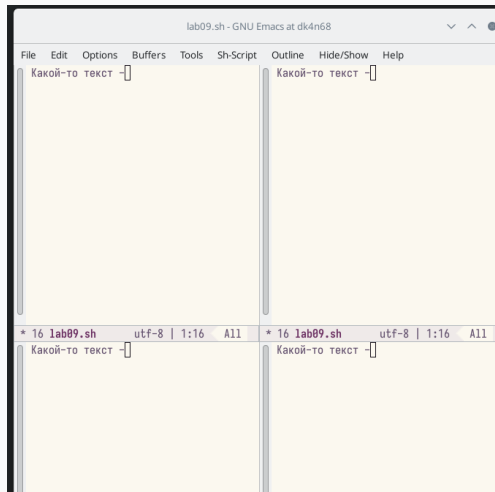
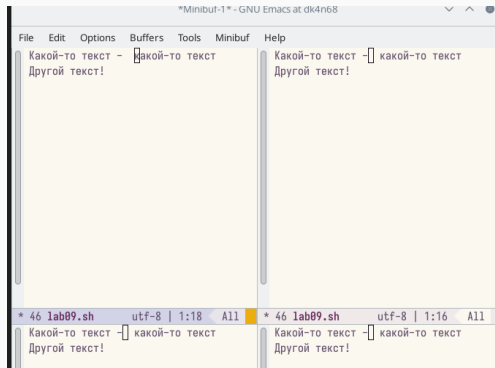


Рис. 19: Команда C-x b

8. Управление окнами. Поделит фрейм на 4 части: разделим фрейм на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2). В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста.



9. Режим поиска. Переключимся в режим поиска (C-s) и найдем несколько слов, присутствующих в тексте. Переключаемся между результатами поиска, нажимая C-s. Выйдем из режима поиска, нажав C-g. Перейдем в режим поиска и замены (M-%), введем текст, который следует найти и заменить, нажимаем Enter , затем вводим текст для замены. После того как будут подсвечены результаты поиска, нажимаем ! для подтверждения замены. Пробуем другой режим поиска, нажав M-s o. От обычного режима отличается тем, что находит не фрагмент текста, а файл.



Я познакомилась с операционной системой Linux. Получила практические навыки работы с редактором Emacs.