

# Отчёт по лабораторной работе № 10

Перелыгин Сергей Викторович

# Содержание

1 Цель работы	5
2 Выполнение лабораторной работы	6
3 Ответы на контрольные вопросы	27
4 Выводы	30

# Список иллюстраций

2.1 Редактор emacs . . . . .	6
2.2 Редактор emacs . . . . .	7
2.3 Создание файла lab07.sh . . . . .	8
2.4 Набираем текст . . . . .	8
2.5 Вырезаем строку . . . . .	9
2.6 Вставляем строку в конец файла . . . . .	9
2.7 Выделяем область текста . . . . .	10
2.8 Вставляем текст в конец файла . . . . .	10
2.9 Выделяем эту область . . . . .	11
2.10 Вырезаем эту область . . . . .	11
2.11 Отменяем последнее действие . . . . .	12
2.12 Список активных буферов . . . . .	13
2.13 Переключаемся во вновь открытое окно . . . . .	14
2.14 Закрываем окно . . . . .	15
2.15 Переключаемся между буферами . . . . .	16
2.16 Переключаемся между буферами . . . . .	16
2.17 Переключаемся между буферами . . . . .	17
2.18 Поделим фрейм на 4 части . . . . .	18
2.19 Создаем файлы . . . . .	19
2.20 Печатаем текст . . . . .	19
2.21 Ищем слово Emacs . . . . .	20
2.22 Выходим из режима поиска . . . . .	21

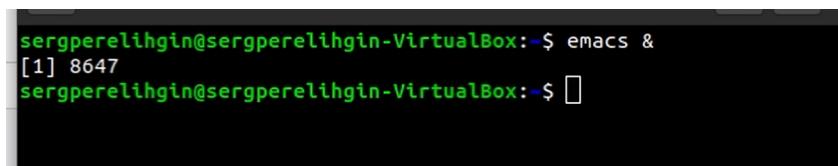
2.23 Заменяем слово в тексте . . . . .	22
2.24 Заменяем слово в тексте . . . . .	23
2.25 Заменяем слово в тексте . . . . .	24
2.26 Заменяем слово в тексте . . . . .	25
2.27 Заменяем слово в тексте . . . . .	26

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Откроем редактор Emacs с помощью команды «emacs &».



```
sergperelihgin@sergperelihgin-VirtualBox:~$ emacs &
[1] 8647
sergperelihgin@sergperelihgin-VirtualBox:~$ █
```

A screenshot of a terminal window with a black background and white text. The text shows a user named 'sergperelihgin' at a host 'sergperelihgin-VirtualBox'. The user runs the command 'emacs &'. The command is followed by a process ID '[1] 8647'. The prompt '\$' is visible again, indicating the session continues.

Рис. 2.1: Редактор emacs

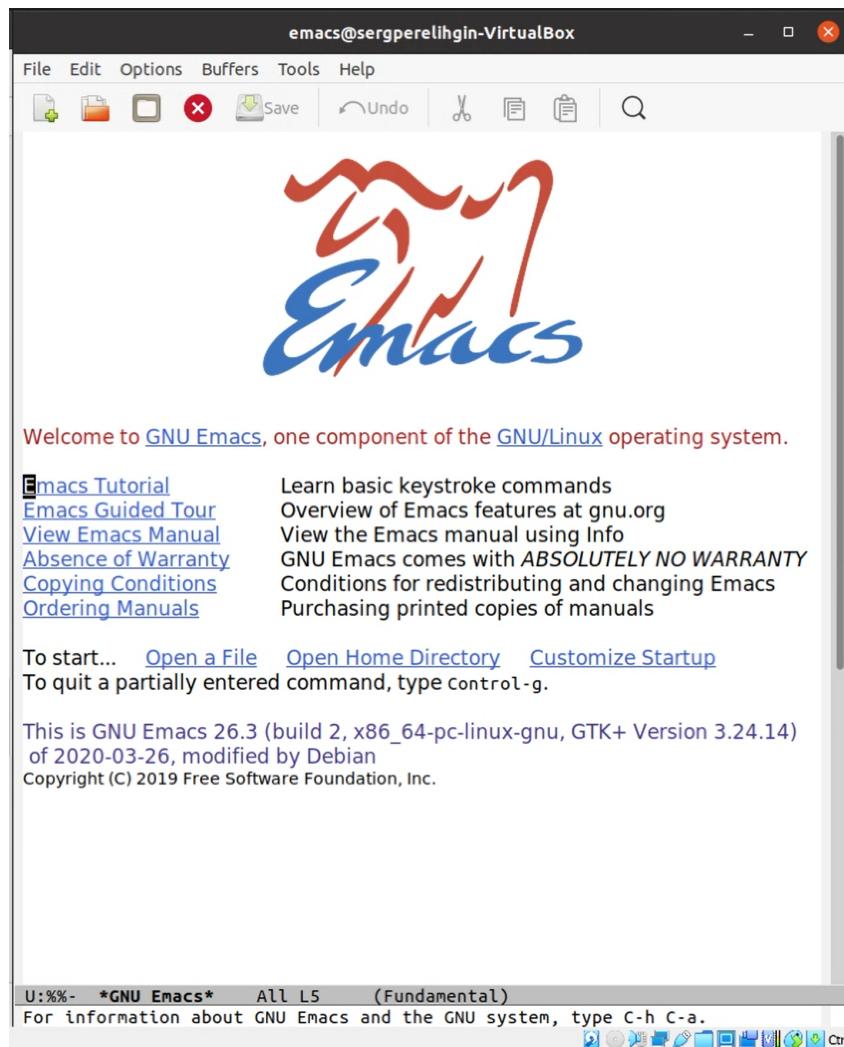


Рис. 2.2: Редактор emacs

2. Создадим файл lab07.sh с помощью комбинации «Ctrl-x» «Ctrl-f».

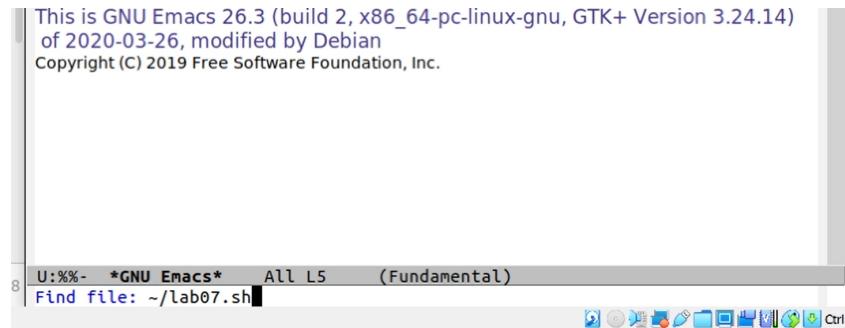


Рис. 2.3: Создание файла lab07.sh

3. В открывшемся буфере наберем необходимый текст.

A screenshot of the Sh-Script editor. The menu bar includes File, Edit, Options, Buffers, Tools, Sh-Script, and Help. The toolbar below the menu bar includes icons for new file, open file, save, undo, and redo. The main code editor window displays the same script as in Figure 2.3:

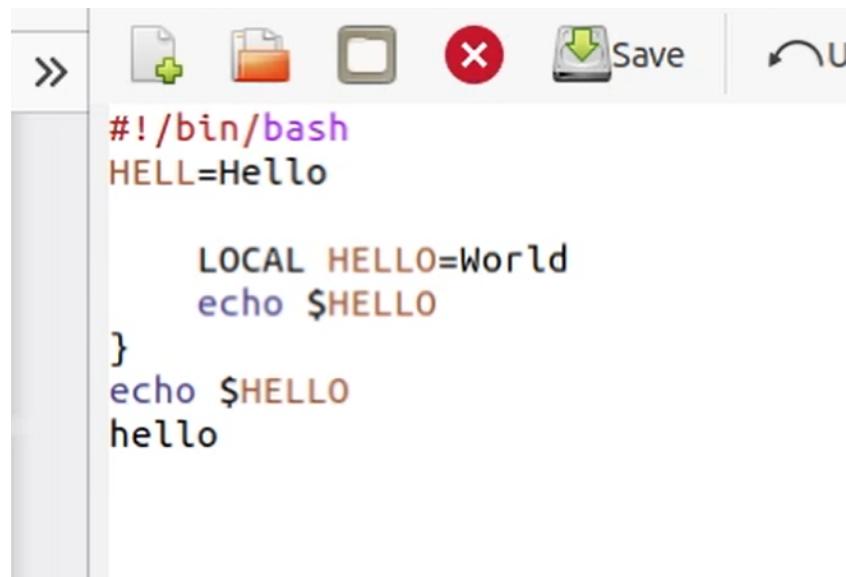
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELL
}
echo $HELL
hello
```

Рис. 2.4: Набираем текст

4. Сохраним файл с помощью комбинации «Ctrl-x»«Ctrl-s».

5.

1) Вырежем одной командой целую строку («Ctrl-k»).

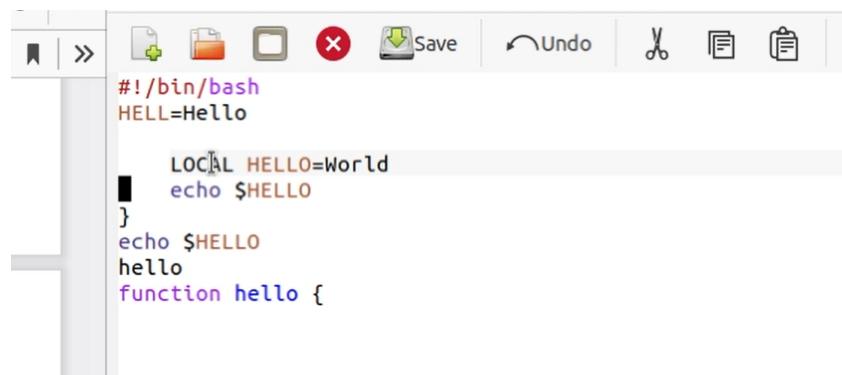


```
#!/bin/bash
HELL=Hello

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

Рис. 2.5: Вырезаем строку

- 2) Вставим эту строку в конец файла («Ctrl-y»).

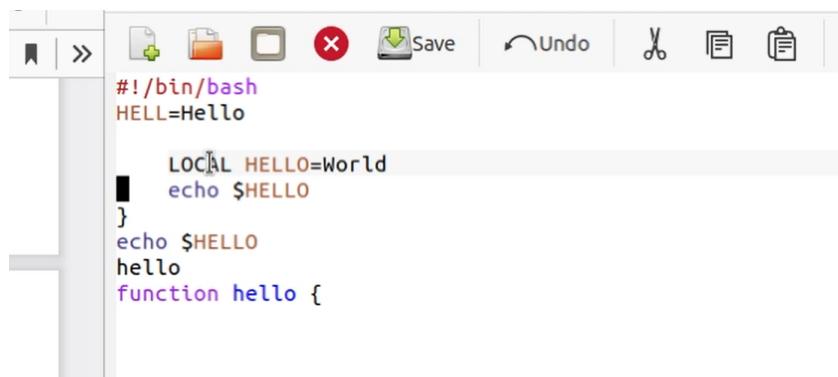


```
#!/bin/bash
HELL=Hello

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

Рис. 2.6: Вставляем строку в конец файла

- 3) Выделим область текста («Ctrl-space»).



A screenshot of a terminal window showing a Bash script. The script content is:

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

LOCAL HELLO=World
echo $HELLO
}

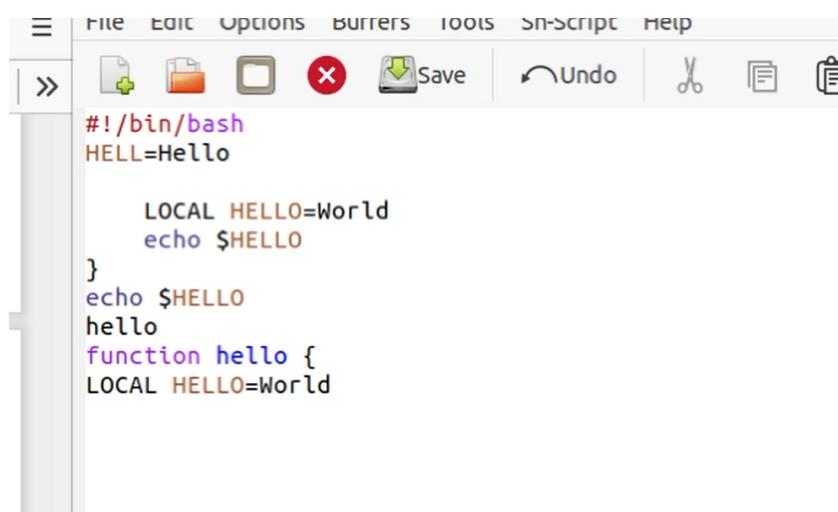
echo $HELLO
hello
function hello {
```

The line "LOCAL HELLO=World" and the line "echo \$HELLO" are highlighted with a light gray selection bar above them.

Рис. 2.7: Выделяем область текста

4) Скопируем область в буфер обмена («Alt-w»).

5) Вставим область в конец файла («Ctrl-y»).



A screenshot of a terminal window showing the same Bash script as in Figure 2.7. A new line has been added at the end of the script:

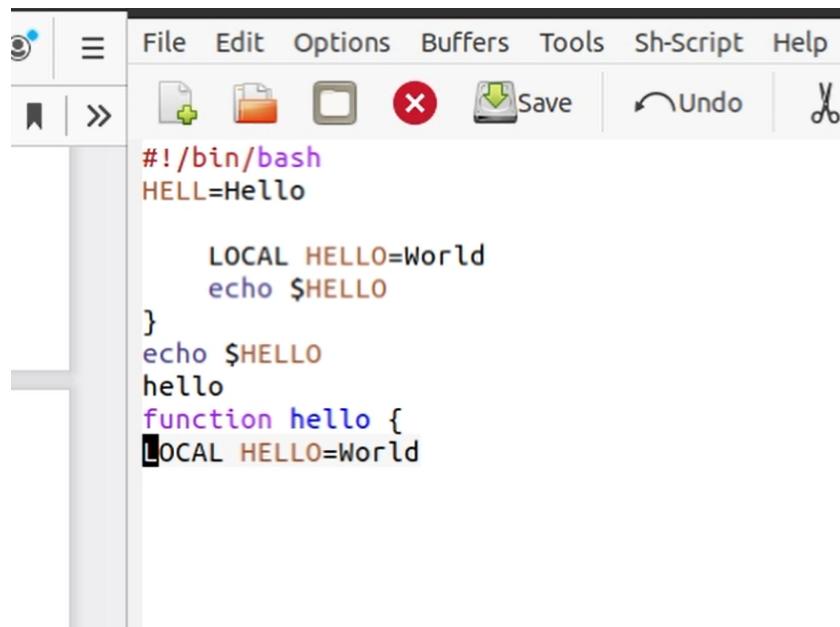
```
#!/bin/bash
HELL=Hello

LOCAL HELLO=World
echo $HELLO
}

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

Рис. 2.8: Вставляем текст в конец файла

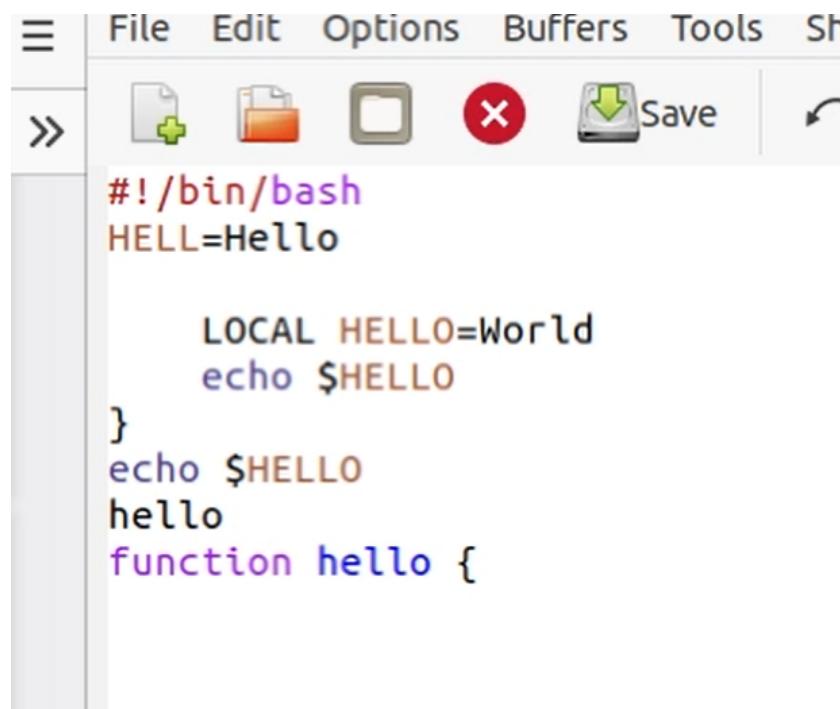
6) Вновь выделим эту область («Ctrl-space») и на этот раз вырежем её («Ctrl-w»).



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

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

Рис. 2.9: Выделяем эту область



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

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

Рис. 2.10: Вырезаем эту область

7) Отменим последнее действие («**Ctrl-/**»).

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

LOCAL HELLO=World
echo $HELLO
}

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

Рис. 2.11: Отменяем последнее действие

6.

- 1) Переместим курсор в начало строки («Ctrl-a»).
- 2) Переместим курсор в конец строки («Ctrl-e»).
- 3) Переместим курсор в начало буфера («M-<») .
- 4) Переместим курсор в конец буфера («M->»).

7.

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

The screenshot shows the Emacs interface with a buffer containing shell script code:

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

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

Below the buffer is a buffer list:

U:**- lab07.sh	All L1	(Shell-script[sh])
CRM Buffer	Size	Mode
0 * lab07.sh	116	Shell-script[sh] ~/lab07.sh
% *GNU Emacs*	705	Fundamental
*scratch*	145	Lisp Interaction
%* *Messages*	1294	Messages

At the bottom, there is a message bar:

U:%%- \*Buffer List\* All L1 (Buffer Menu)  
C-x <mouse-1> is undefined

Рис. 2.12: Список активных буферов

- 2) Переместимся во вновь открытое окно («**Ctrl-x o**») со списком открытых буферов и переключимся на другой буфер (для этого необходимо нажать на «**enter**» после выбора необходимого буфера).

The screenshot shows an Emacs window titled "emacs@sergperetin-in-VirtualBox". The menu bar includes File, Edit, Options, Buffers, Tools, Buffer-Menu, and Help. The toolbar contains icons for file operations like Open, Save, Undo, and Cut/Paste. The main buffer displays a shell script:

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

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

The status bar at the bottom shows "U:--\*- lab07.sh All L10 (Shell-script[sh])". Below the main buffer is a buffer list:

	Size	Mode	File
* lab07.sh	116	Shell-script[sh]	~/lab07.sh
% *GNU Emacs*	705	Fundamental	
*scratch*	145	Lisp Interaction	
%* *Messages*	1294	Messages	

The status bar also shows "U:%%- \*Buffer List\* All L1 (Buffer Menu)".

Рис. 2.13: Переключаемся во вновь открытое окно

3) Закроем это окно («**Ctrl-x 0**»).

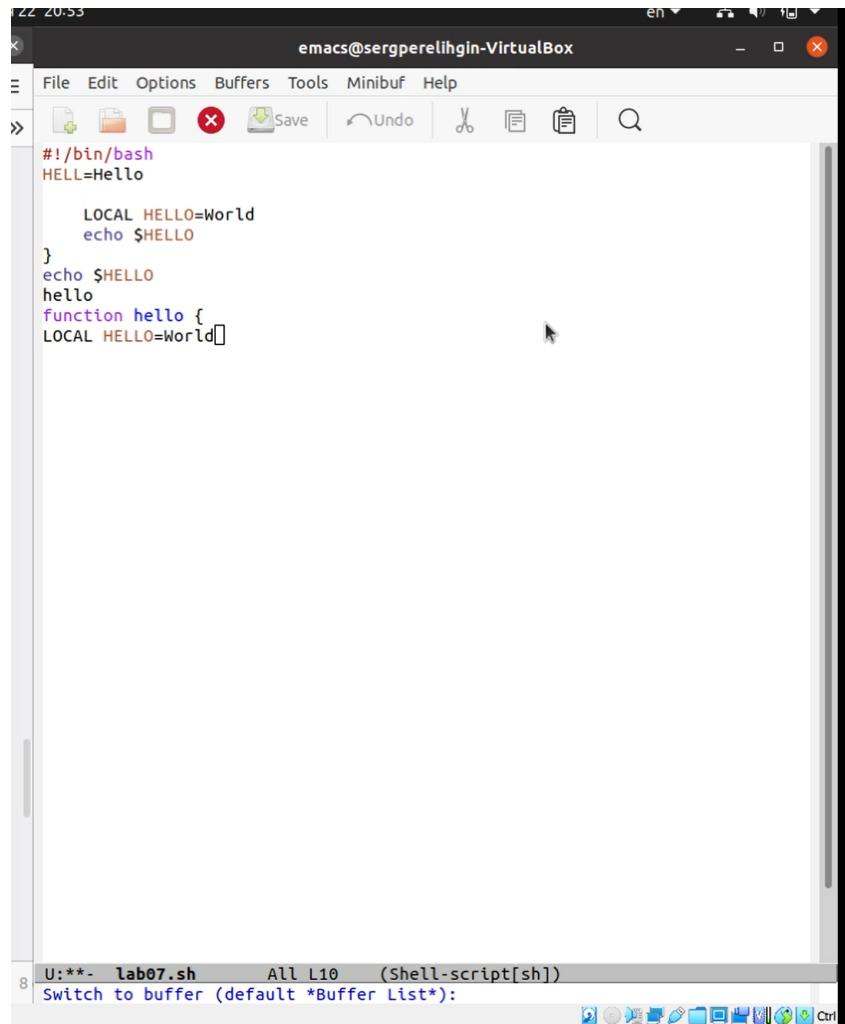


Рис. 2.14: Закрываем окно

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

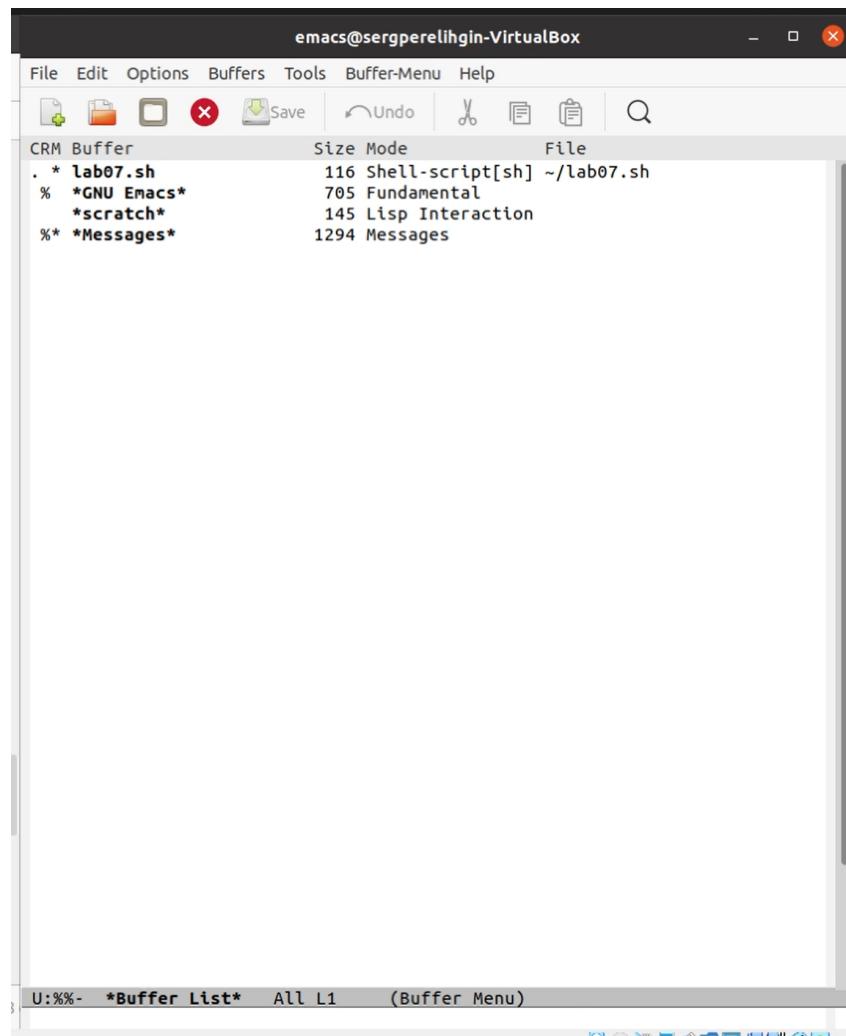


Рис. 2.15: Переключаемся между буферами

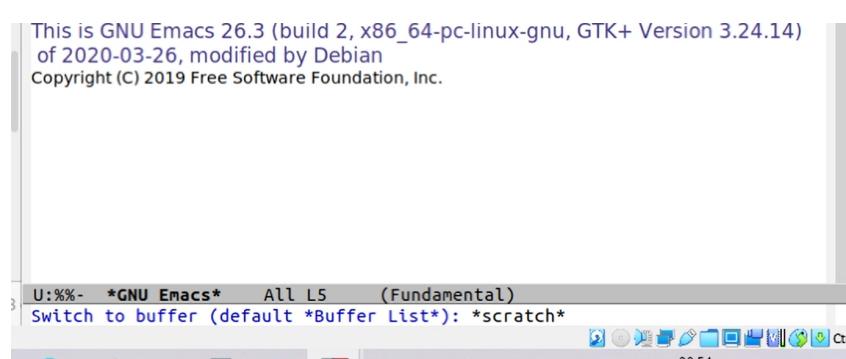


Рис. 2.16: Переключаемся между буферами

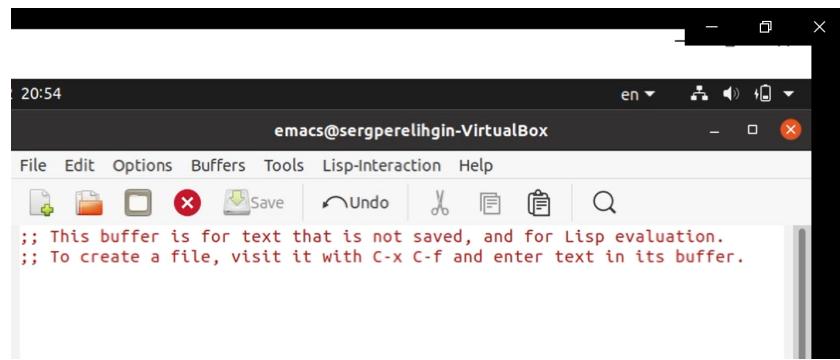


Рис. 2.17: Переключаемся между буферами

8.

- 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**»).

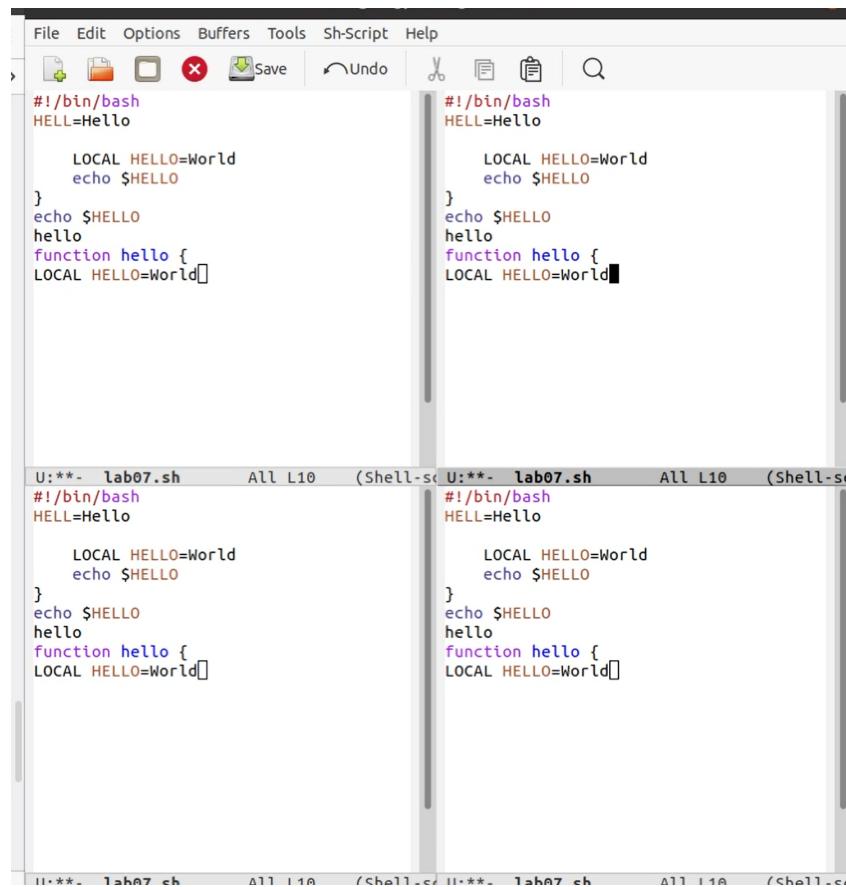


Рис. 2.18: Поделим фрейм на 4 части

- 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 file1.txt», «touch file2.txt», «touch file3.txt», «touch file4.txt».

```
2 20:56
sergperelihgin@sergperelihgin-VirtualBox: ~
sergperelihgin@sergperelihgin-VirtualBox:~$ emacs &
[1] 8647
sergperelihgin@sergperelihgin-VirtualBox:~$ touch file1.txt
sergperelihgin@sergperelihgin-VirtualBox:~$ touch file2.txt
sergperelihgin@sergperelihgin-VirtualBox:~$ touch file3.txt
sergperelihgin@sergperelihgin-VirtualBox:~$ touch file4.txt
sergperelihgin@sergperelihgin-VirtualBox:~$
```

Рис. 2.19: Создаем файлы

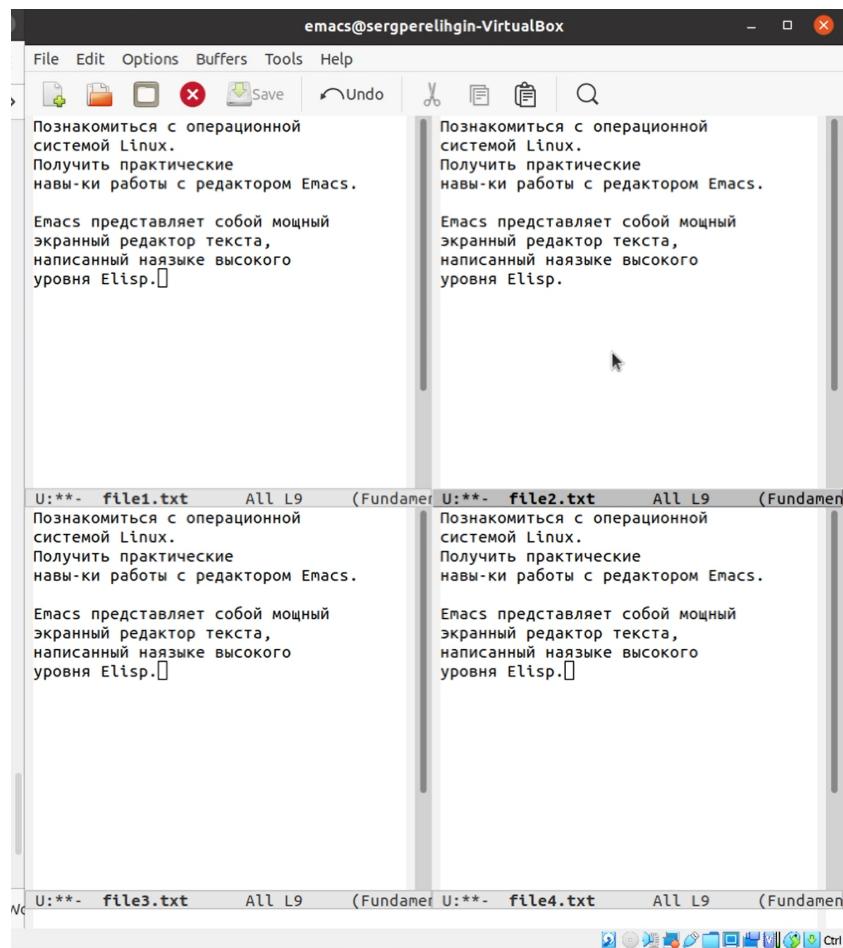


Рис. 2.20: Печатаем текст

9.

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

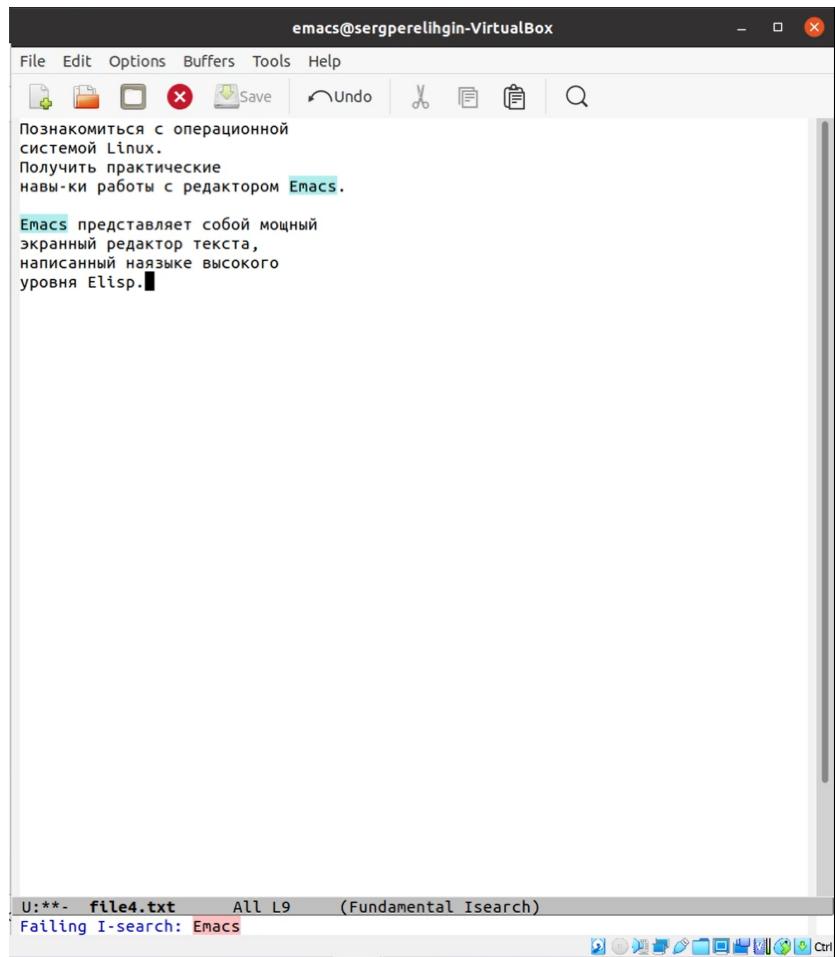


Рис. 2.21: Ищем слово Emacs

- 2) Переключимся между результатами поиска, нажимая «Ctrl-s».
- 3) Выйдем из режима поиска, нажав «Ctrl-g».

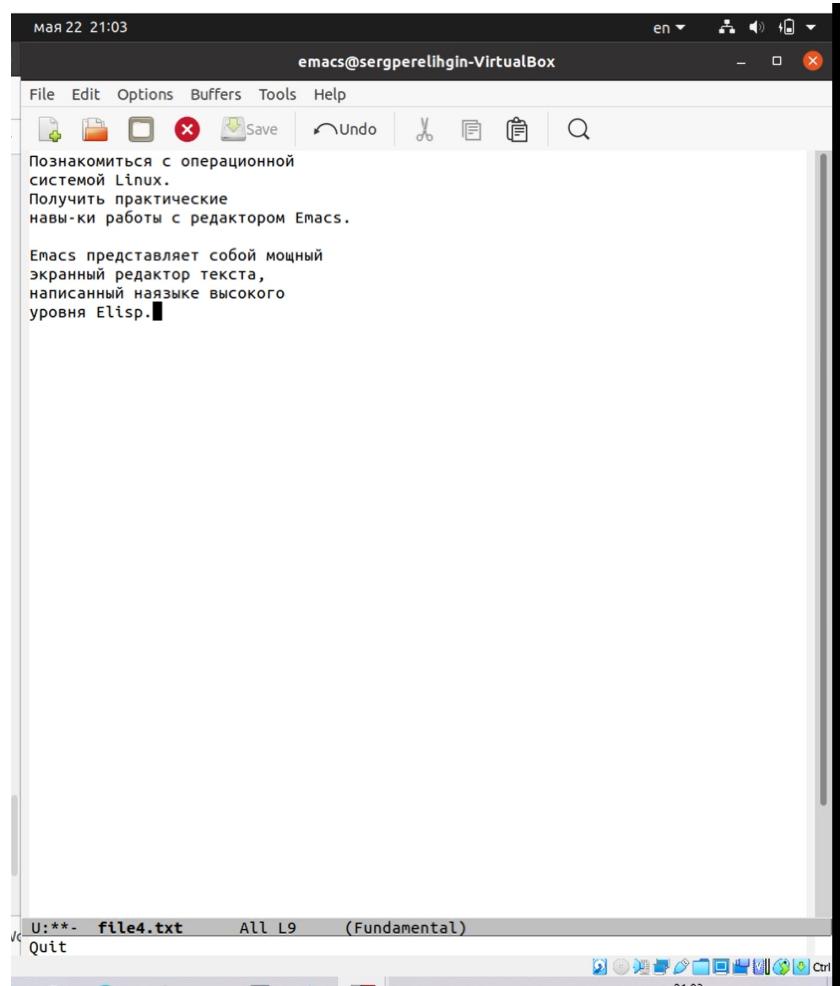


Рис. 2.22: Выходим из режима поиска

- 4) Перейдем в режим поиска и замены («M-%»), 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.

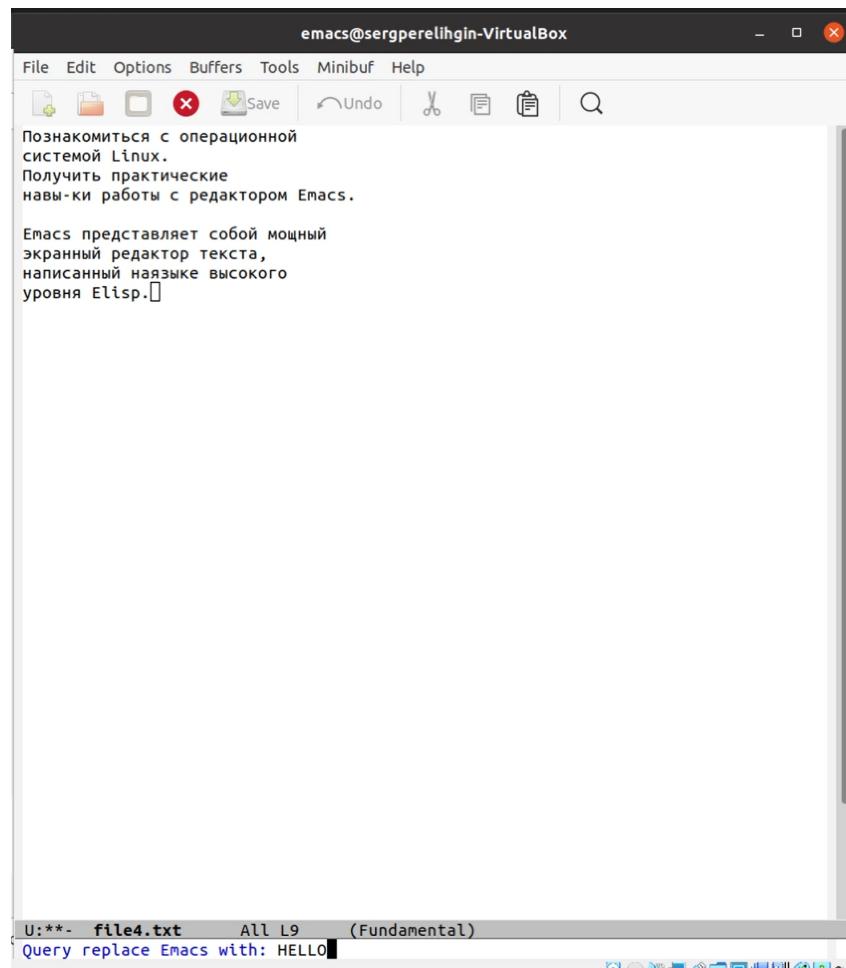


Рис. 2.23: Заменяем слово в тексте

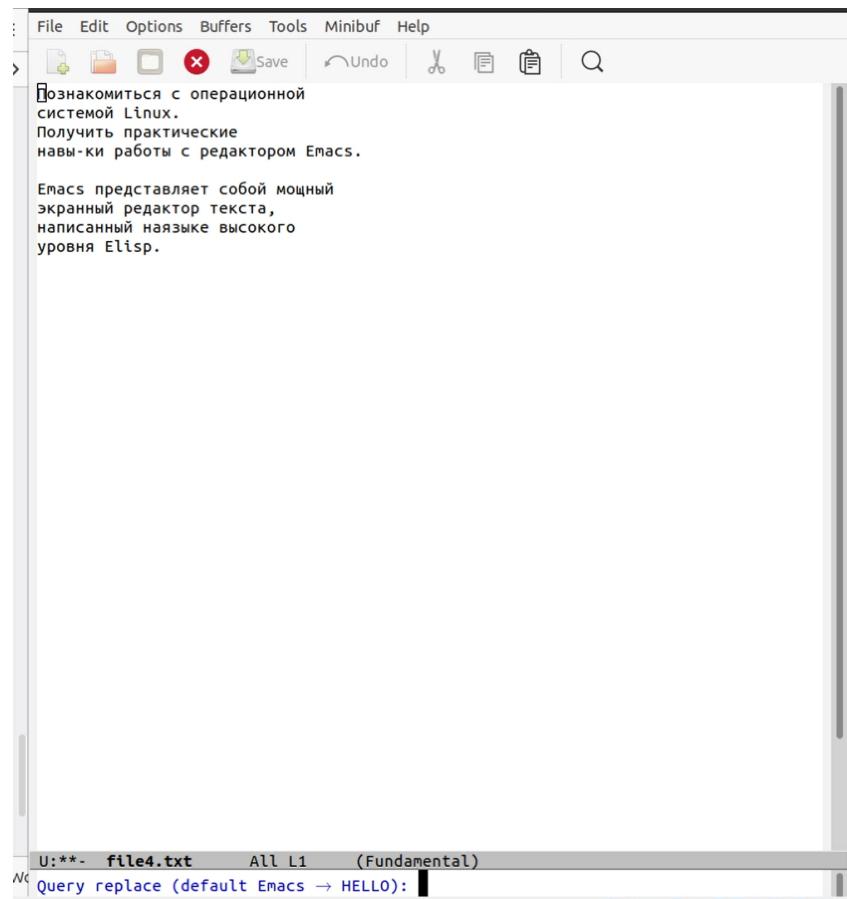


Рис. 2.24: Заменяем слово в тексте

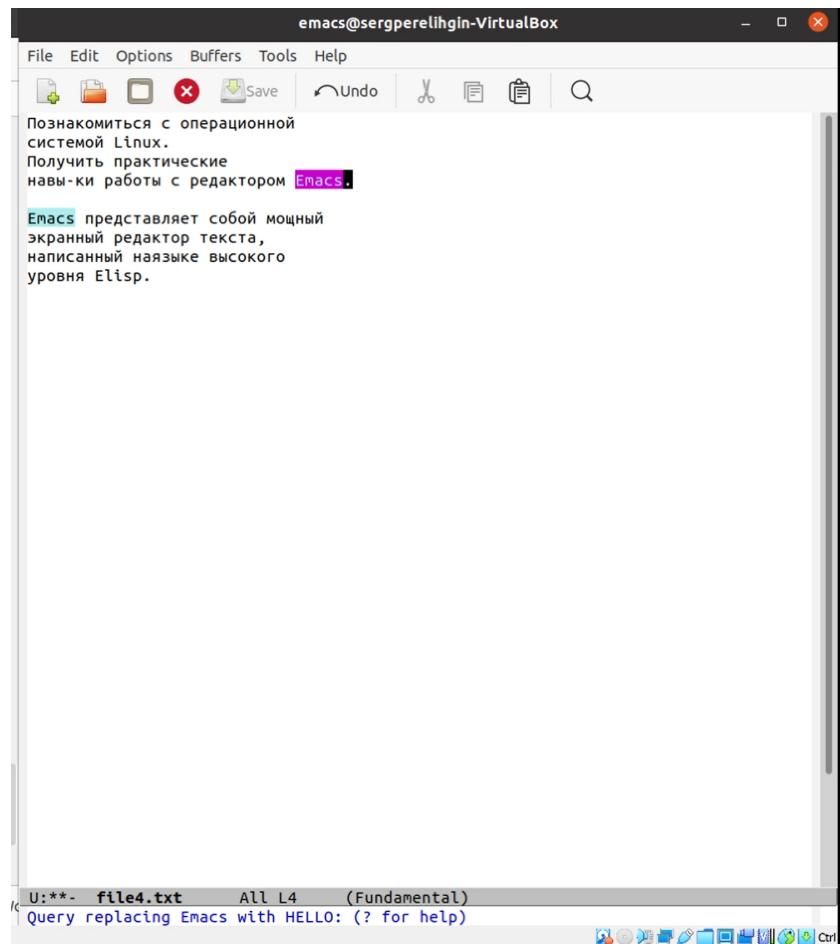


Рис. 2.25: Заменяем слово в тексте

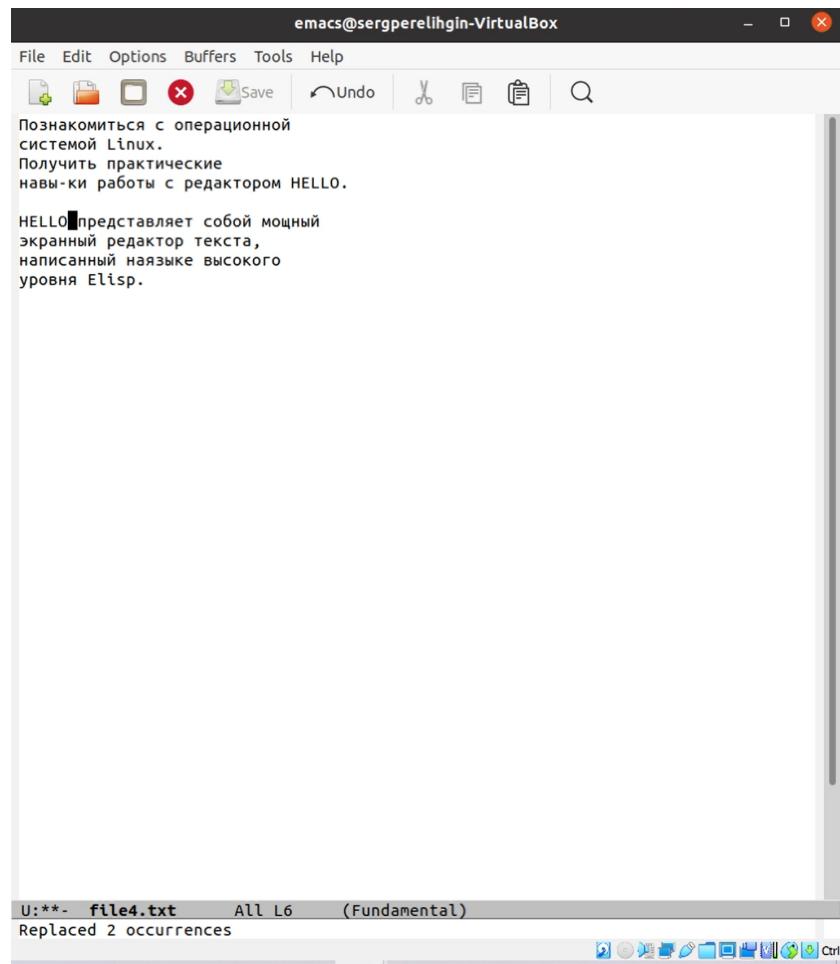


Рис. 2.26: Заменяем слово в тексте

5) Пробуем другой режим поиска, нажав «M-s o»

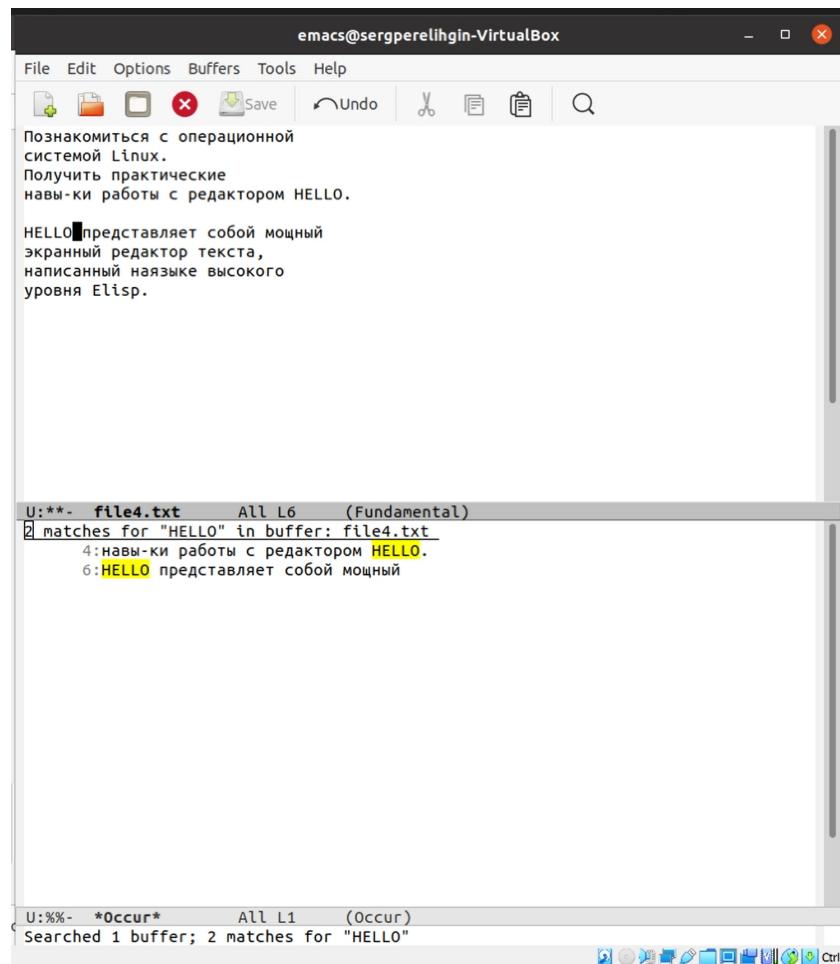


Рис. 2.27: Заменяем слово в тексте

### 3 Ответы на 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 этого 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 открываться дополнительные окна. Окна Emacs и окна графической среды XWindow – разные вещи. Одно окно XWindow 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)
- «GNUEmacs» (справочный буфер о редакторе)

6. С-с |сначала, удерживая «ctrl», нажимаю «с», после –отпускаю обе клавиши нажимаю «|» С-сС-|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 сохраняются в файле .emacs.

9. По умолчанию клавиша «←» удаляет символ перед курсором, нов 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 неудобным

## 4 Выводы

в ходе выполнения данной лабораторной работы я познакомился с операционной системой Linux и получил практические навыки работы с редактором Emacs.