

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

дисциплина: Операционные системы

Студент: Ким Реачна

Группа: НПИбд-02-20

Москва

2021г.

Цель работы

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

Теоретические сведения:

Emacs представляет собой мощный экраный редактор текста, написанный на языке высокого уровня Lisp.

Основные термины Emacs

Буфер — объект, представляющий какой-либо текст.

Фрейм соответствует окну в обычном понимании этого слова. Каждый фрейм содержит область вывода и одно или несколько окно Emacs.

Окно — прямоугольная область фрейма, отображающая один из буферов.

Область вывода — одна или несколько строк внизу фрейма, в которой Emacs выводит различные сообщения, а также запрашивает подтверждения и дополнительную информацию от пользователя.

Минибуфер используется для ввода дополнительной информации и всегда отображается в области вывода.

Точка вставки — место вставки (удаления) данных в буфере.

Основы работы в Emacs

Для запуска Emacs необходимо в командной строке набрать `emacs` (или `emacs &` для работы в фоновом режиме относительно консоли).

Для работы с Emacs можно использовать как элементы меню, так и различные сочетания клавиш. Например, для выхода из Emacs можно воспользоваться меню `File` и выбрать пункт `Quit`, а можно нажать последовательно `Ctrl-x Ctrl-c` (в обозначениях Emacs: `C-x C-c`).

По назначению префиксные сочетания клавиш различаются следующим образом:

- `C-x` — префикс ввода основных команд редактора (например, открытия, закрытия, сохранения файла и т.д.);
- `C-c` — префикс вызова функций, зависящих от используемого режима.

Режим — пакет расширений, изменяющий поведение буфера Emacs при редактировании и просмотре текста (например, для редактирования исходного текста программ на языках C или Perl).

Более подробная ссылка на материал [\[1\]](#) [\[2\]](#)

Выполнение работы:

1. Открыть `emacs`: Чтобы открыть Emacs, мы используем команду `emacs`, так как у меня нет Emacs, поэтому мне нужно сначала установить его с помощью команды `sudo snap install emacs --classic` (*Рисунок 1*), после завершения установки снова запустите команду `emacs`, мы увидим редактор окон (*Рисунок 2*).

Рисунок 1: Установка emacs

```

kim@kim-VirtualBox:~$ emacs

Command 'emacs' not found, but can be installed with:

sudo snap install emacs      # version 27.2, or
sudo apt install e3          # version 1:2.71-2
sudo apt install emacs-gtk   # version 1:26.3+1-1ubuntu2
sudo apt install emacs-lucid # version 1:26.3+1-1ubuntu2
sudo apt install emacs-nox   # version 1:26.3+1-1ubuntu2
sudo apt install jove        # version 4.17.2.7-1

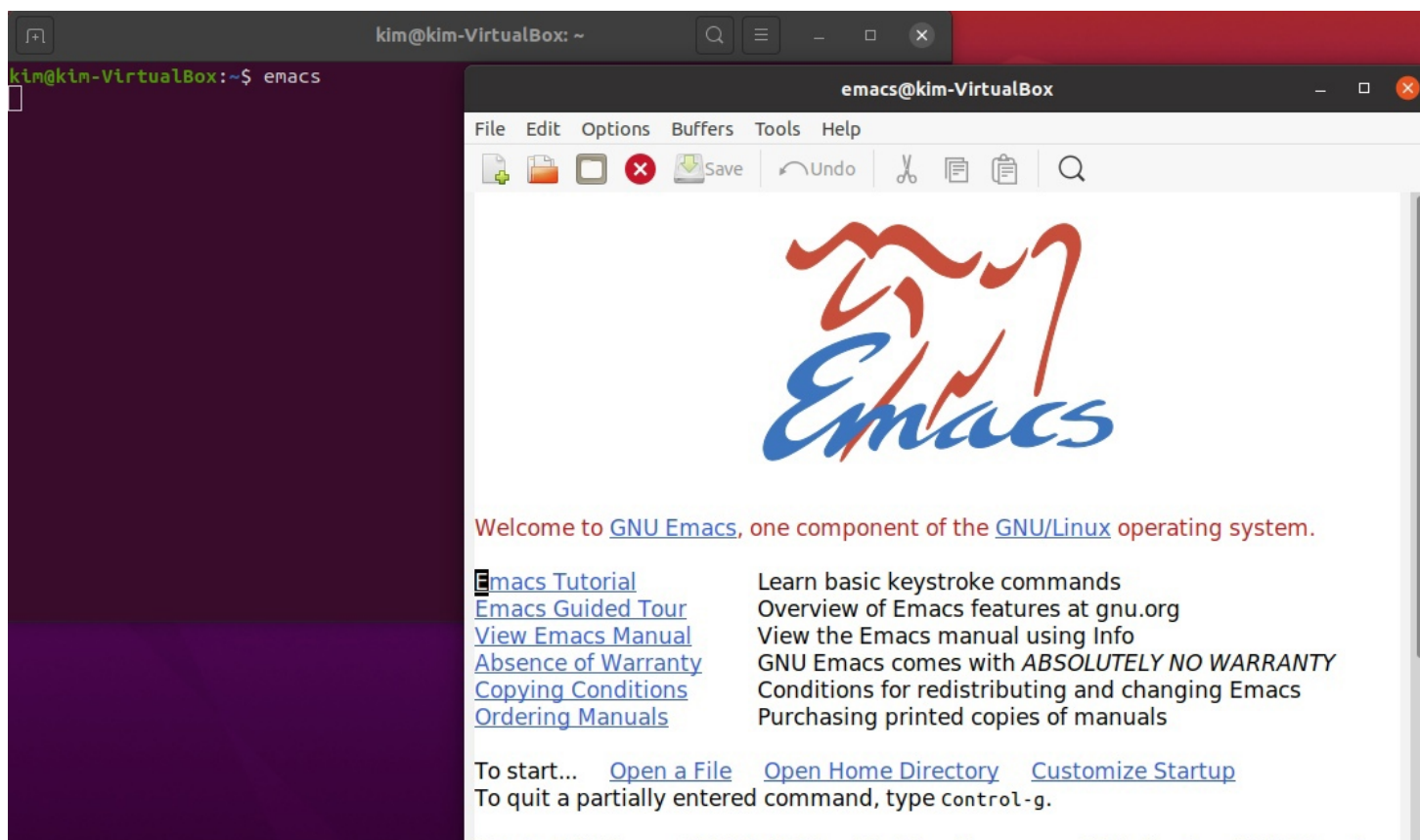
See 'snap info emacs' for additional versions.

kim@kim-VirtualBox:~$ sudo snap install emacs
[sudo] password for kim:
error: This revision of snap "emacs" was published using classic confinement
and thus may perform arbitrary system changes outside of the security
sandbox that snaps are usually confined to, which may put your system at
risk.

If you understand and want to proceed repeat the command including
--classic.
kim@kim-VirtualBox:~$ sudo snap install emacs --classic
emacs 27.2 from Alex Murray (alexmurray) installed

```

Рисунок 2: открываем редактор Emacs



2. Создать файл `lab10.sh` с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).

Рисунок 3: Создать файл `lab10.sh`

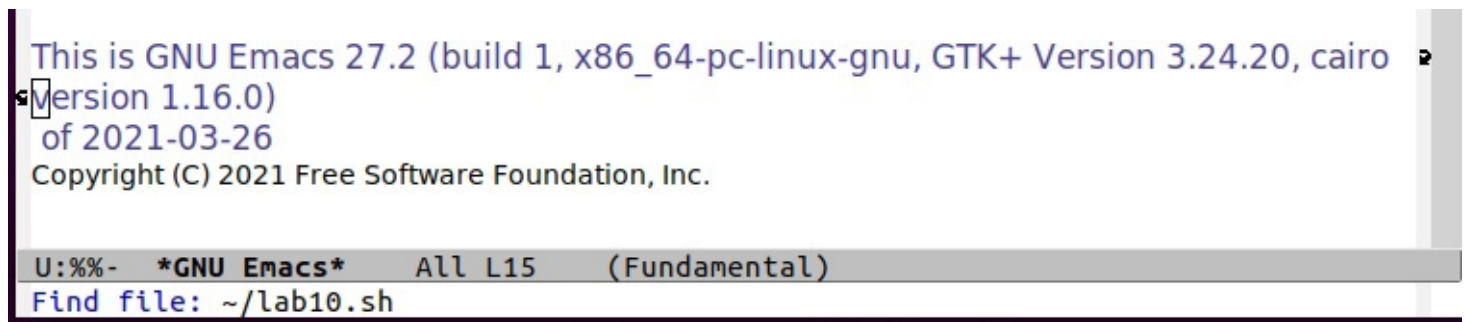
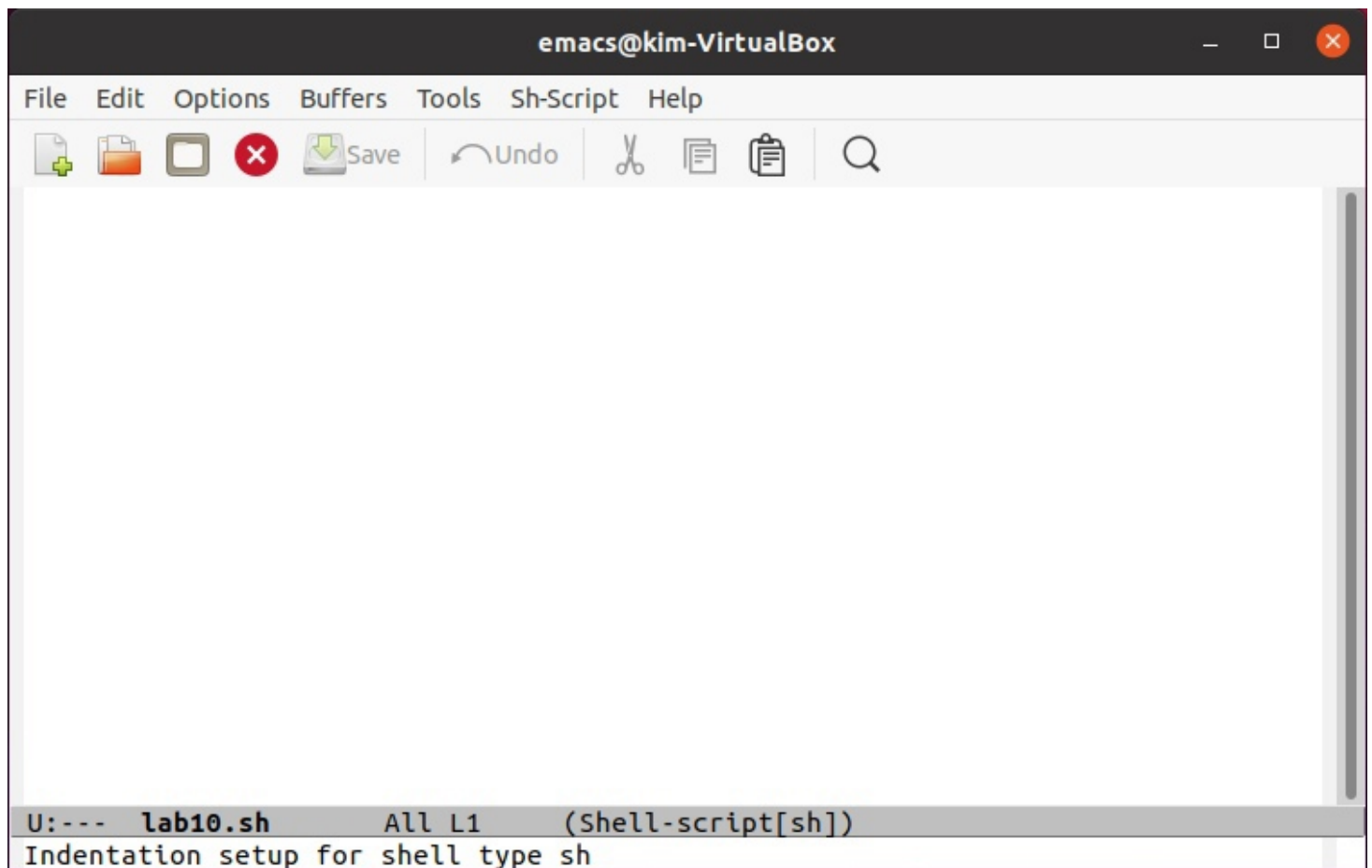


Рисунок 4: Файл `lab10.sh`



3. Наберите текст:

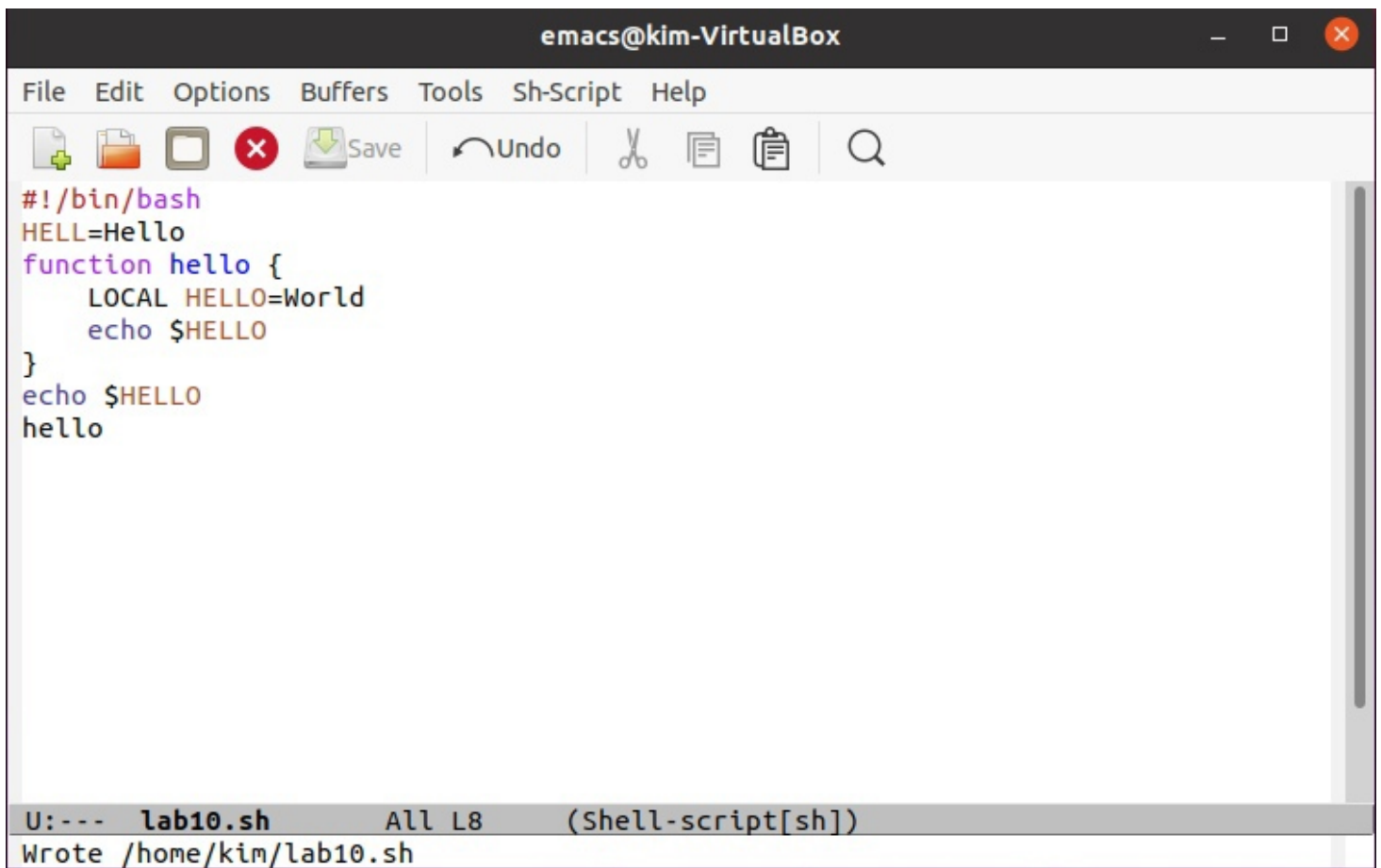
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Рисунок 5: набор текста



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

Рисунок 6: Сохранить файл



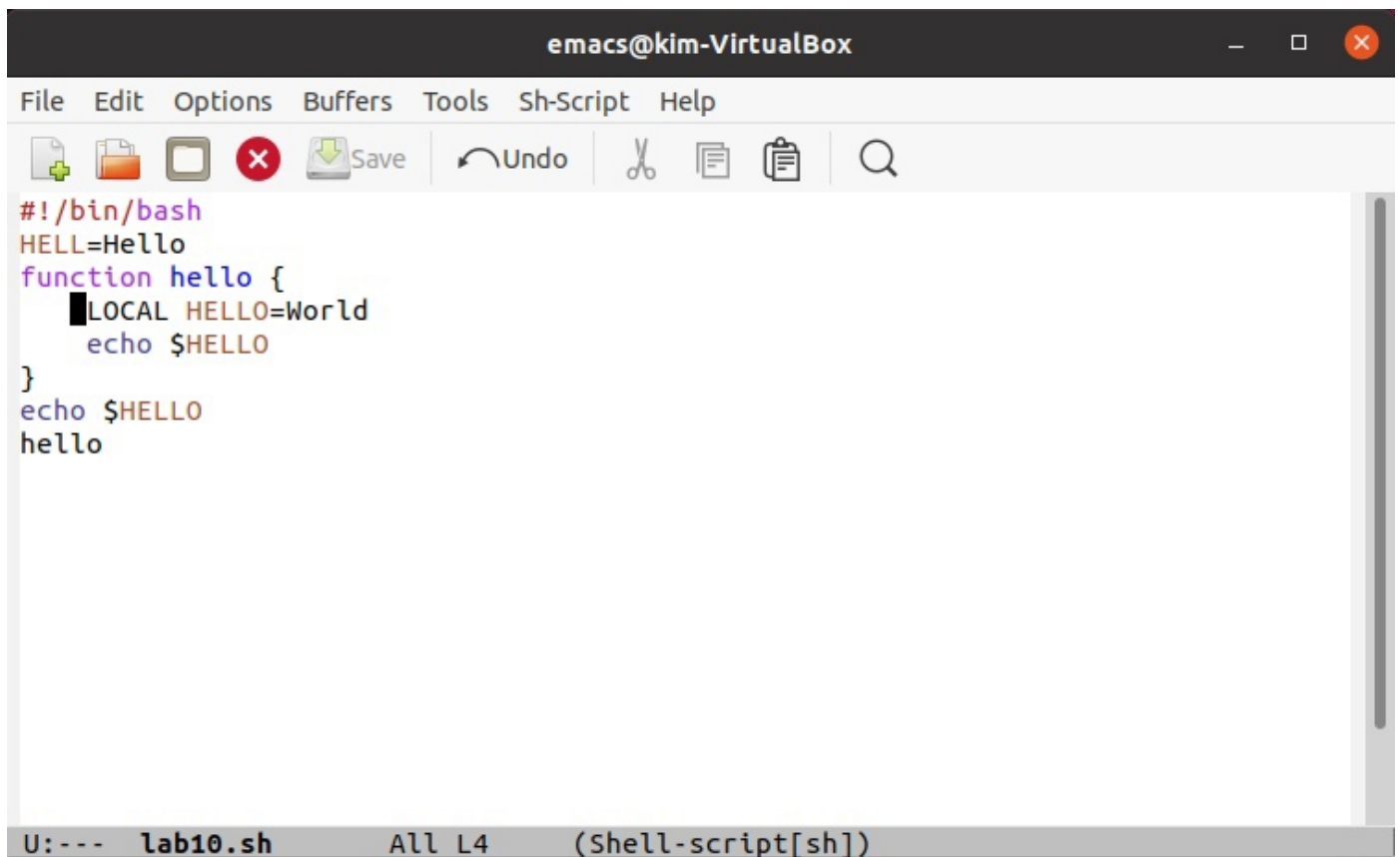
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

U:--- lab10.sh All L8 (Shell-script[sh])
Wrote /home/kim/lab10.sh

5. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш:

5.1. Вырезать одной командой целую строку с-к : сначала мы выбираем строку , которую хотим выстроить, выбираем с самого начала (Рисунок 7) , а затем используем команду с - к , чтобы вырезать строку (Рисунок 8).

Рисунок 7: Выберите строку, которую мы хотим вырезать



The screenshot shows the Emacs editor window titled "emacs@kim-VirtualBox". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for file operations (new, open, save, close), editing (undo, redo, cut, copy, paste), and search. The code in the buffer is a shell script:

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

The status bar at the bottom indicates the file is "lab10.sh", the cursor is at line 4, column 14, and the buffer is "(Shell-script[sh])".

Рисунок 8: используйте `C-k`, чтобы обрезать строку



The screenshot shows the same Emacs editor window after the `C-k` command has been executed. The code is now:

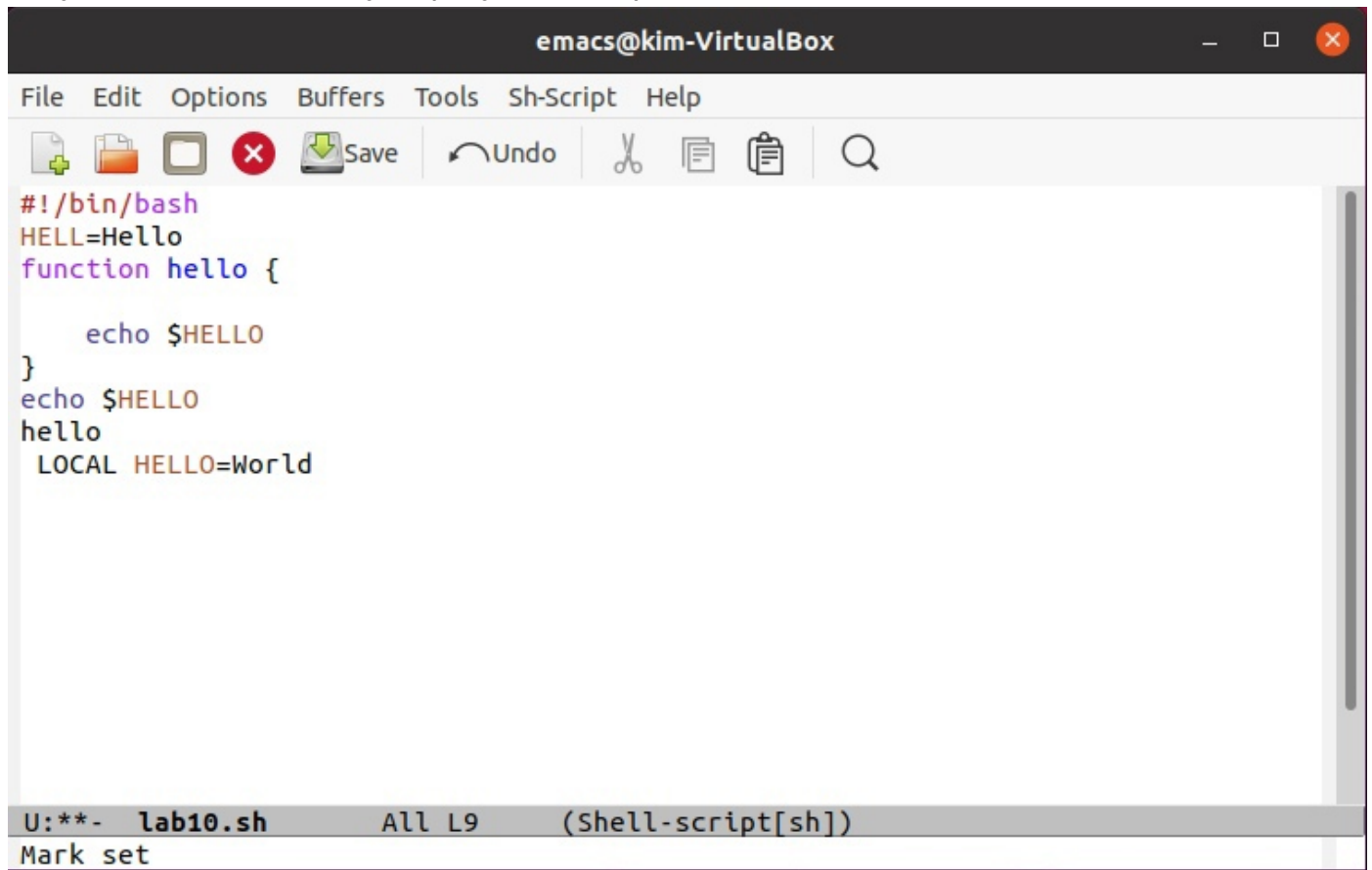
```
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
```

The status bar remains the same, indicating the file is "lab10.sh", the cursor is at line 4, column 14, and the buffer is "(Shell-script[sh])".

Как мы видим, четвертая строка успешно обрезана.

5.2. Вставить эту строку в конец файла `с-у` : для этого мы переходим в конец файла, а затем используем команду `с - у` , чтобы вставить эту строку (Рисунок 9)

Рисунок 9: Вставить эту строку в конец файла



5.3. Выделить область текста `с-space` :В работе я выбираю третью строку и использую `с - space` , а затем использую клавишу со стрелкой до конца строки, а затем использую `м - w` , чтобы скопировать выбранную нами строку (Рисунок 10).

Рисунок 10: Выделить область текста (`с-space`) и `М-w` для копирования



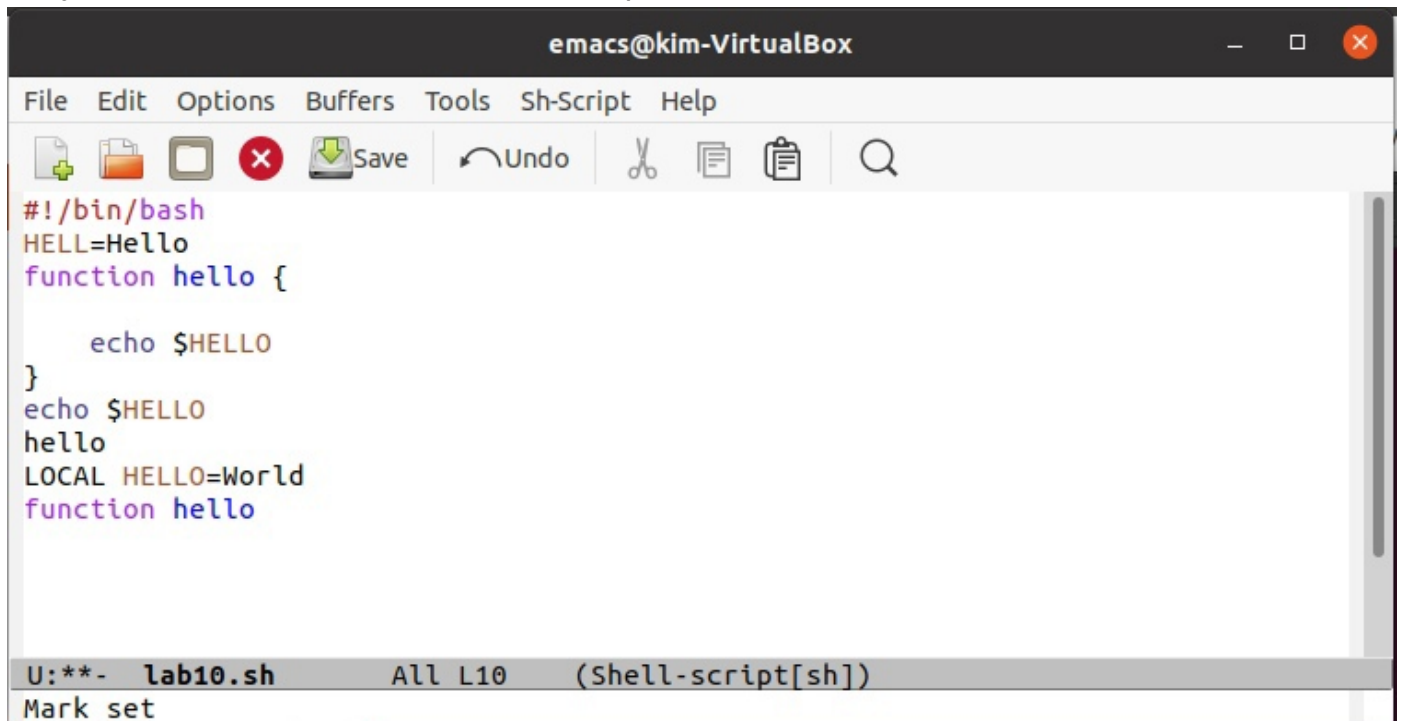
```
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
```

U:**- lab10.sh All L7 (Shell-script[sh])
Mark activated

5.4. Скопировать область в буфер обмена м-w (Рисунок 10).

5.5. Вставить область в конец файла: для этого мы снова переходим к концу файла, а затем используем с-у для вставки (Рисунок 11).

Рисунок 11: Вставить область в конец файла

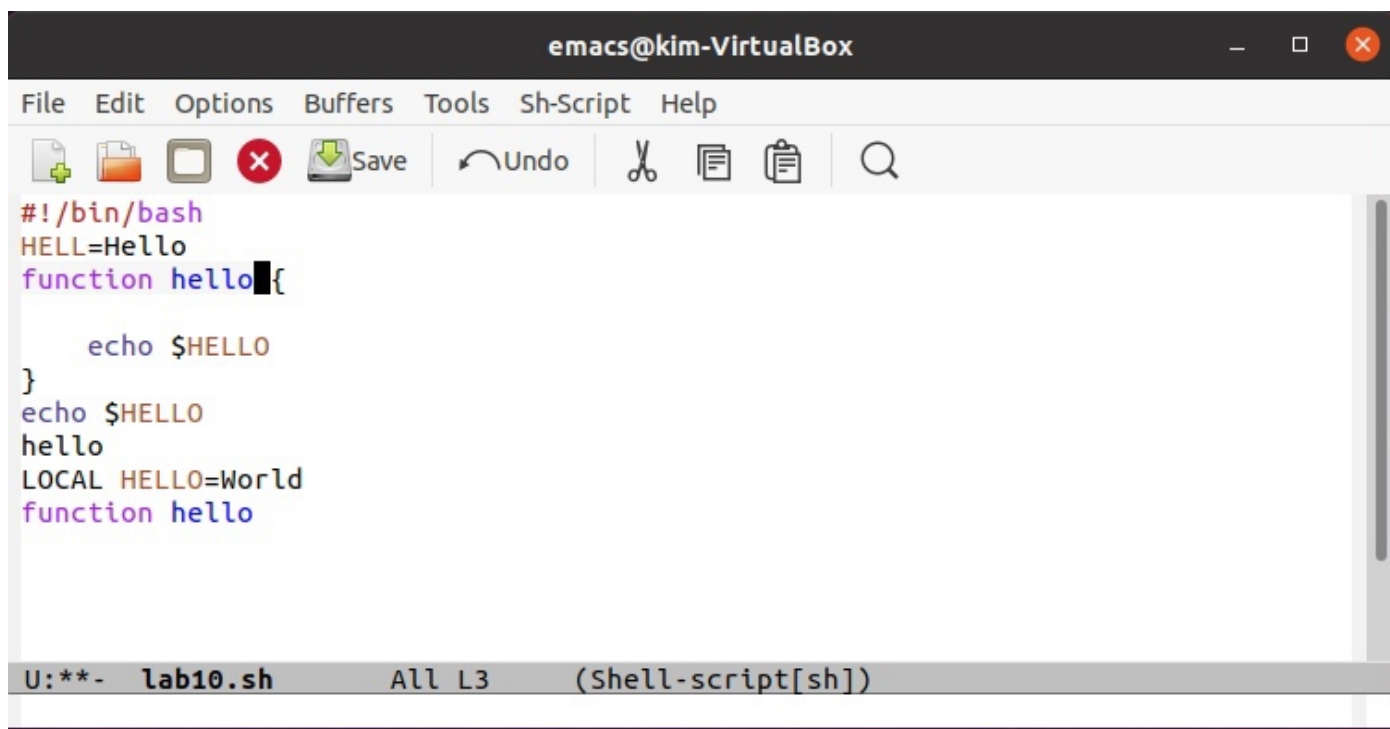


```
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
```

U:**- lab10.sh All L10 (Shell-script[sh])
Mark set

5.6. Вновь выделить эту область с - space (Рисунок 12) и на этот раз вырезать её с - w (Рисунок 13)

Рисунок 12: выделить область

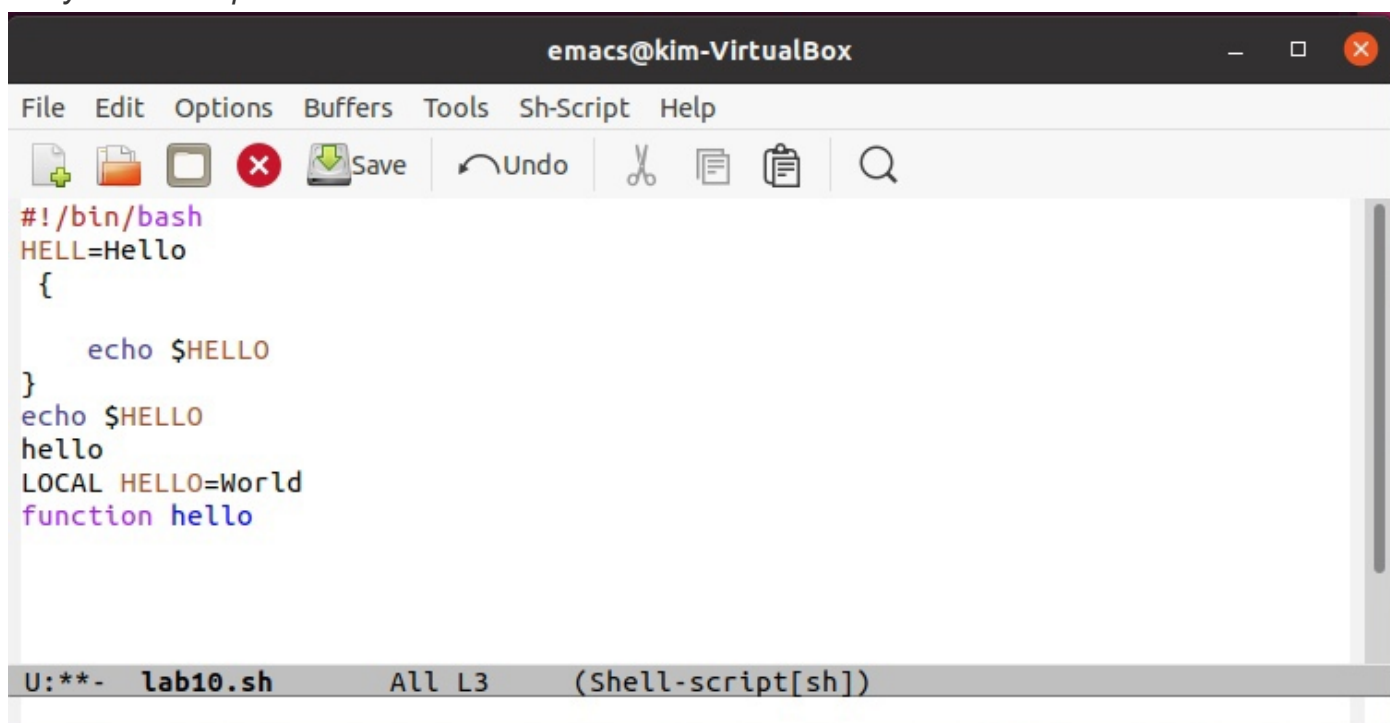


The screenshot shows the Emacs editor window titled "emacs@kim-VirtualBox". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for file operations (new, open, save, undo, redo, copy, paste, search) and a search icon. The main text area contains a shell script snippet:

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

The status bar at the bottom indicates the current file is "lab10.sh", the cursor is at line 13, and the buffer is "(Shell-script[sh])".

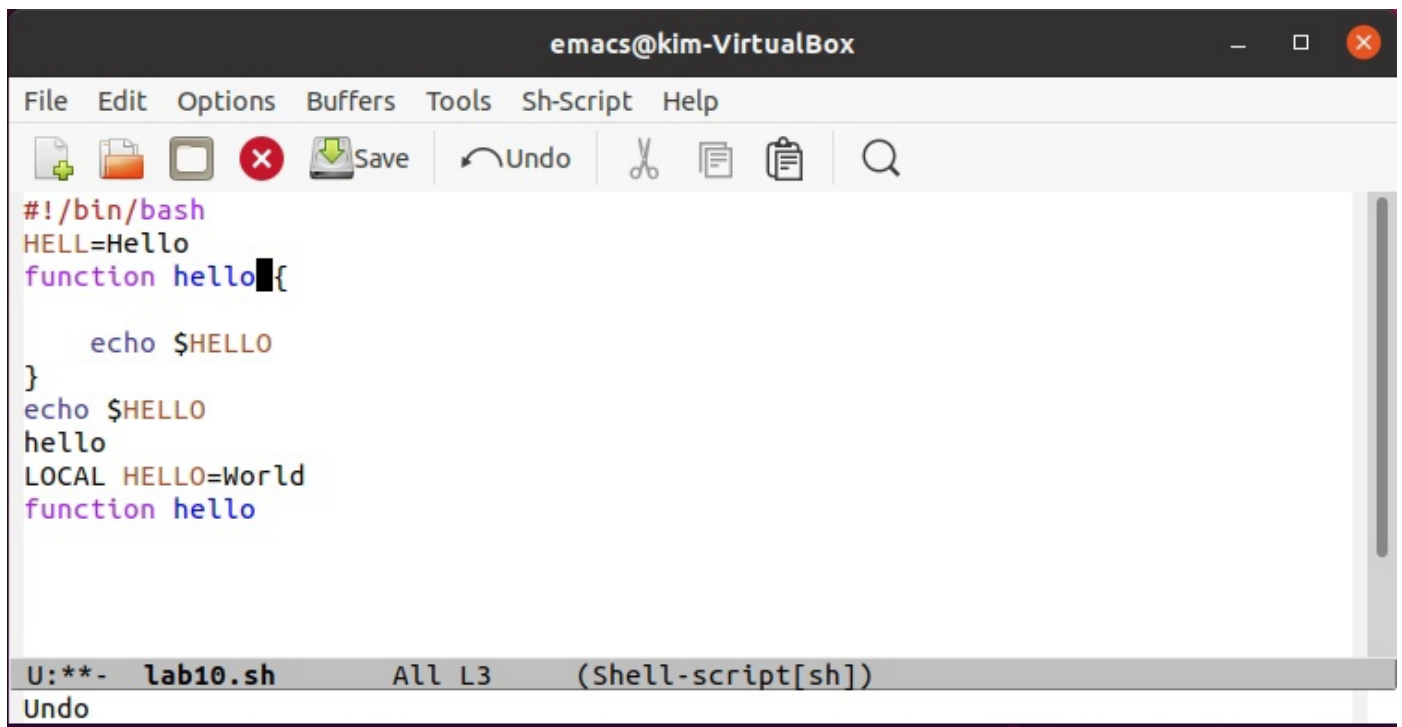
Рисунок 13: вырезка текста



This screenshot is identical to the one in Figure 13, showing the same Emacs editor window with the same shell script snippet and status bar information.

5.7. Отмените последнее действие с - / (Рисунок 14)

Рисунок 14: Отмените последнее действи



```
emacs@kim-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
HELL=Hello
function hello{
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
```

U:*** lab10.sh All L3 (Shell-script[sh])

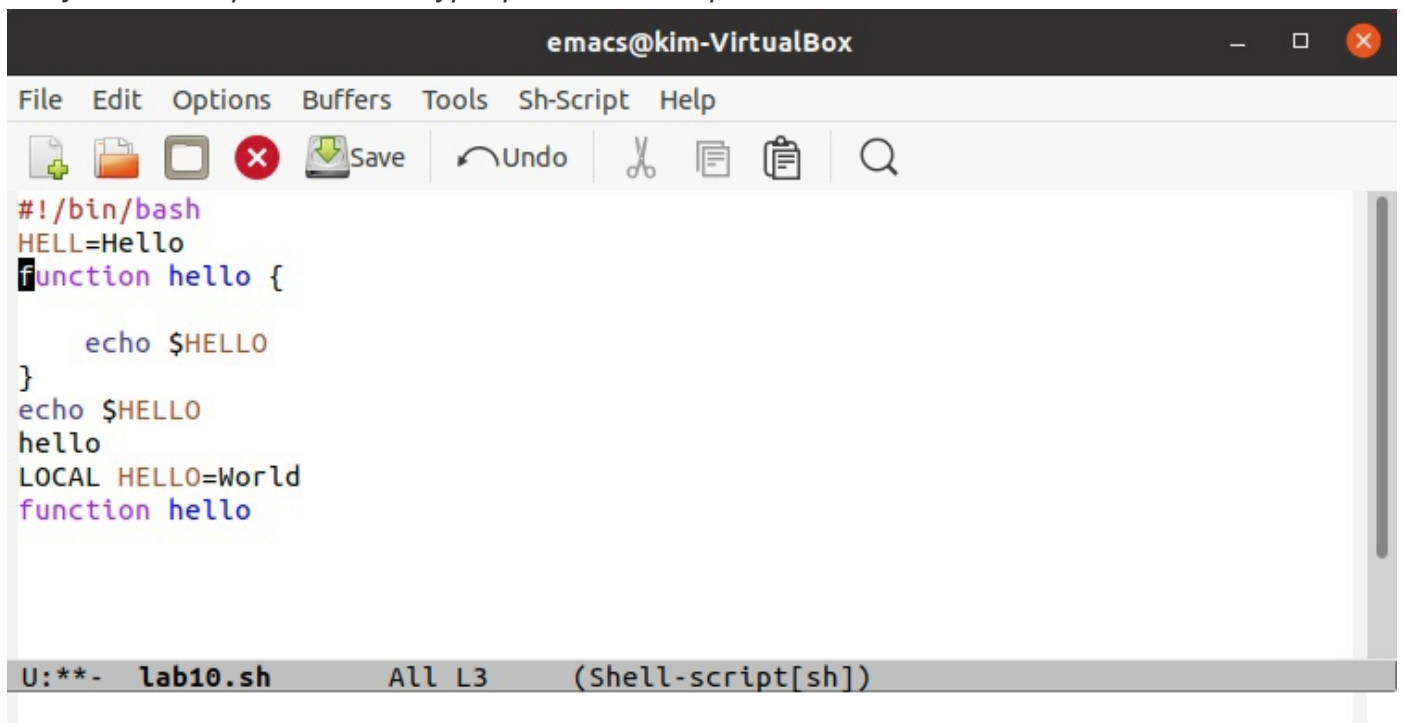
Undo

Как мы видим, текст возвращается обратно.

6. Научитесь использовать команды по перемещению курсора.

6.1. Переместите курсор в начало строки с - а (Рисунок 15) : как мы видим начальное положение курсора на (рисунке 14).

Рисунок 15: Переместите курсор в начало строки

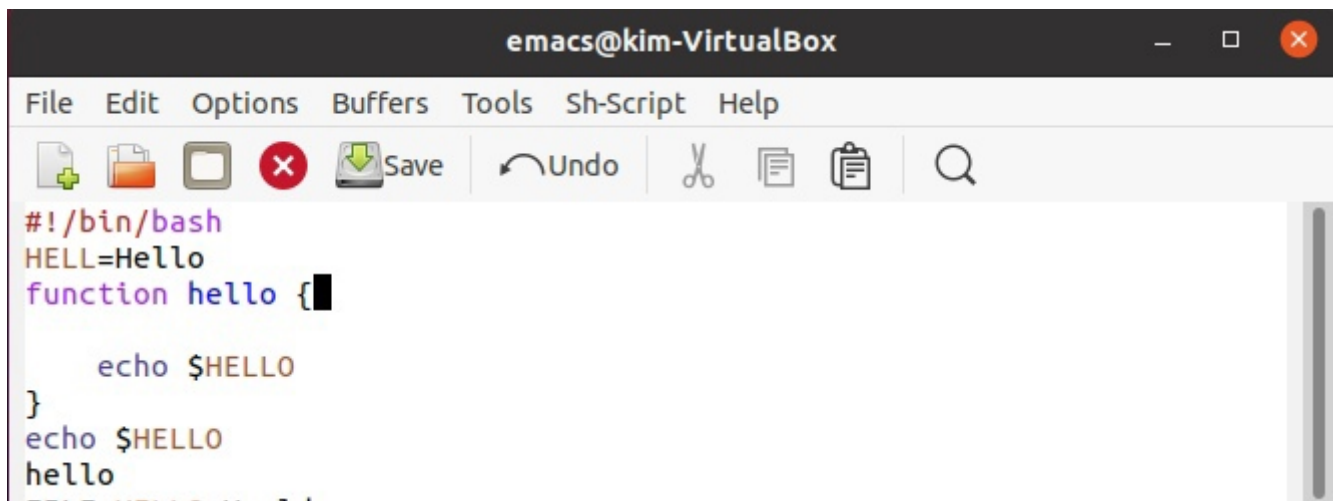


```
emacs@kim-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
```

U:*** lab10.sh All L3 (Shell-script[sh])

6.2. Переместите курсор в конец строки с-е (Рисунок 16)

Рисунок 16: Переместите курсор в конец строки

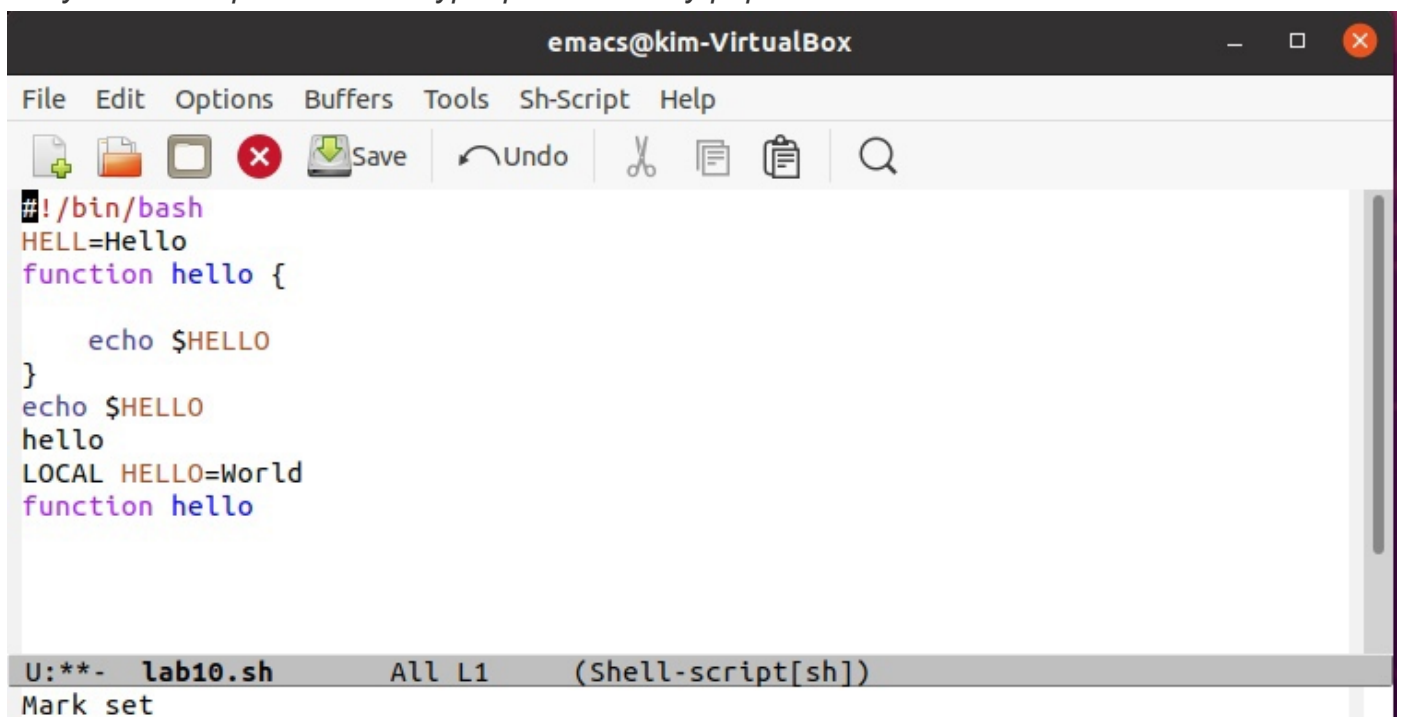
A screenshot of the Emacs editor window titled 'emacs@kim-VirtualBox'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for file operations (new, open, save, close), editing (undo, redo, cut, copy, paste), and search. The main text area shows a shell script buffer with the following content:

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

 The cursor is positioned at the end of the 'hello' command on the last line.

6.3. Переместите курсор в начало буфера м - < . (Рисунок 17)

Рисунок 17: Переместите курсор в начало буфера

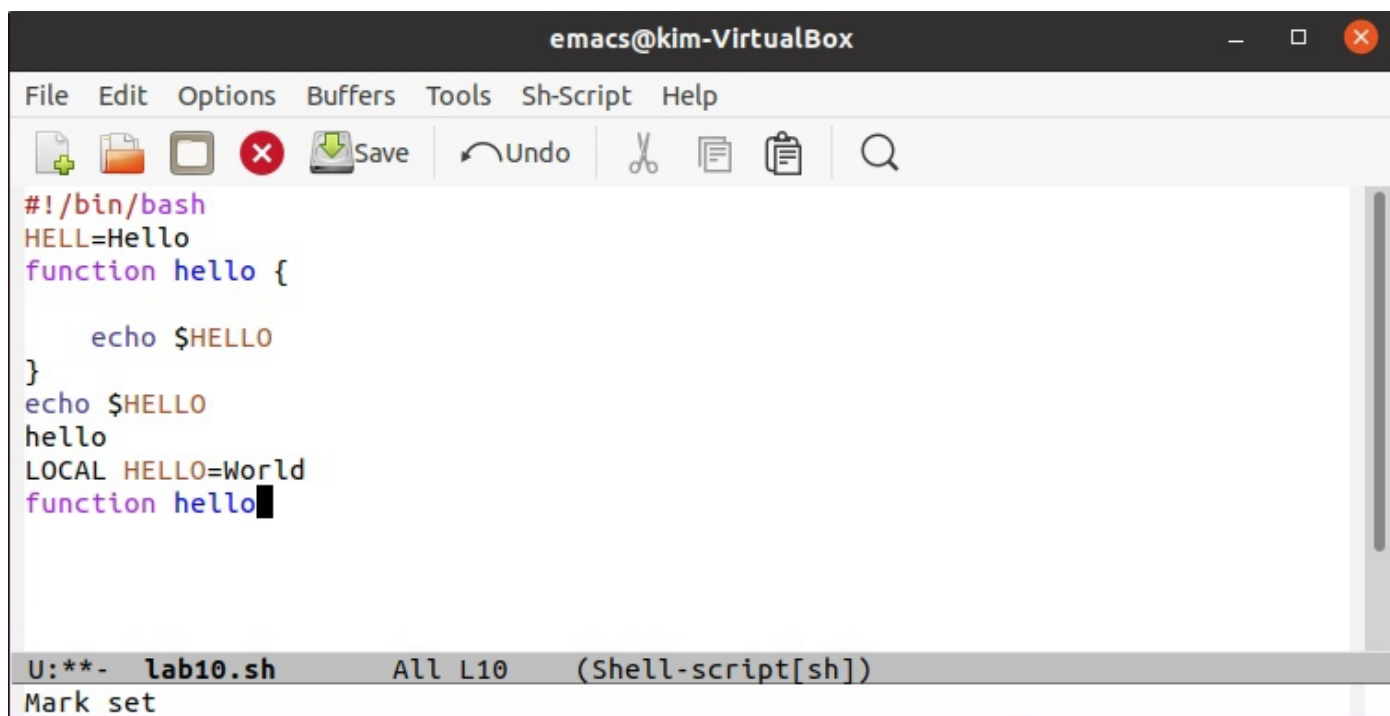
A screenshot of the Emacs editor window titled 'emacs@kim-VirtualBox'. The menu bar and toolbar are the same as in the previous image. The main text area shows the same shell script buffer, but with an additional line added:

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

 The cursor is now at the end of the 'function hello' line on the last line. The status bar at the bottom shows 'U:**- lab10.sh All L1 (Shell-script[sh])' and 'Mark set'.

6.4. Переместите курсор в конец буфера м - > . (Рисунок 18)

Рисунок 18: Переместите курсор в конец буфера



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

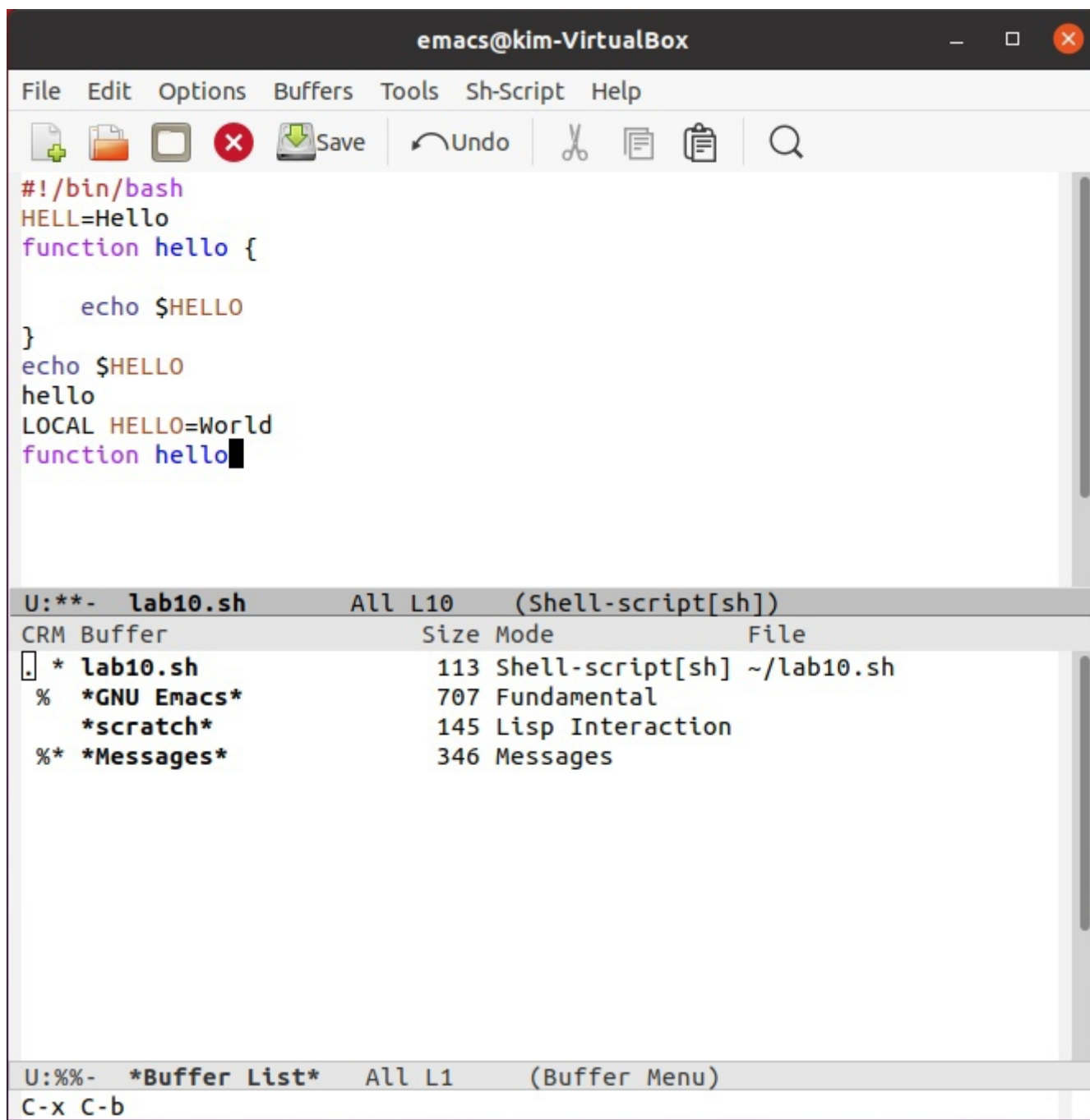
U:**- lab10.sh All L10 (Shell-script[sh])

Mark set

7. Управление буферами.

7.1. Вывести список активных буферов на экран с-х с-б (Рисунок 19)

Рисунок 19: Вывести список активных буферов



7.2. Переместитесь во вновь открытое окно `с-х о` (Рисунок 20) со списком открытых буферов и переключитесь на другой буфер `с-х в` (Рисунок 21)

Рисунок 20: Переместитесь во вновь открытое окно

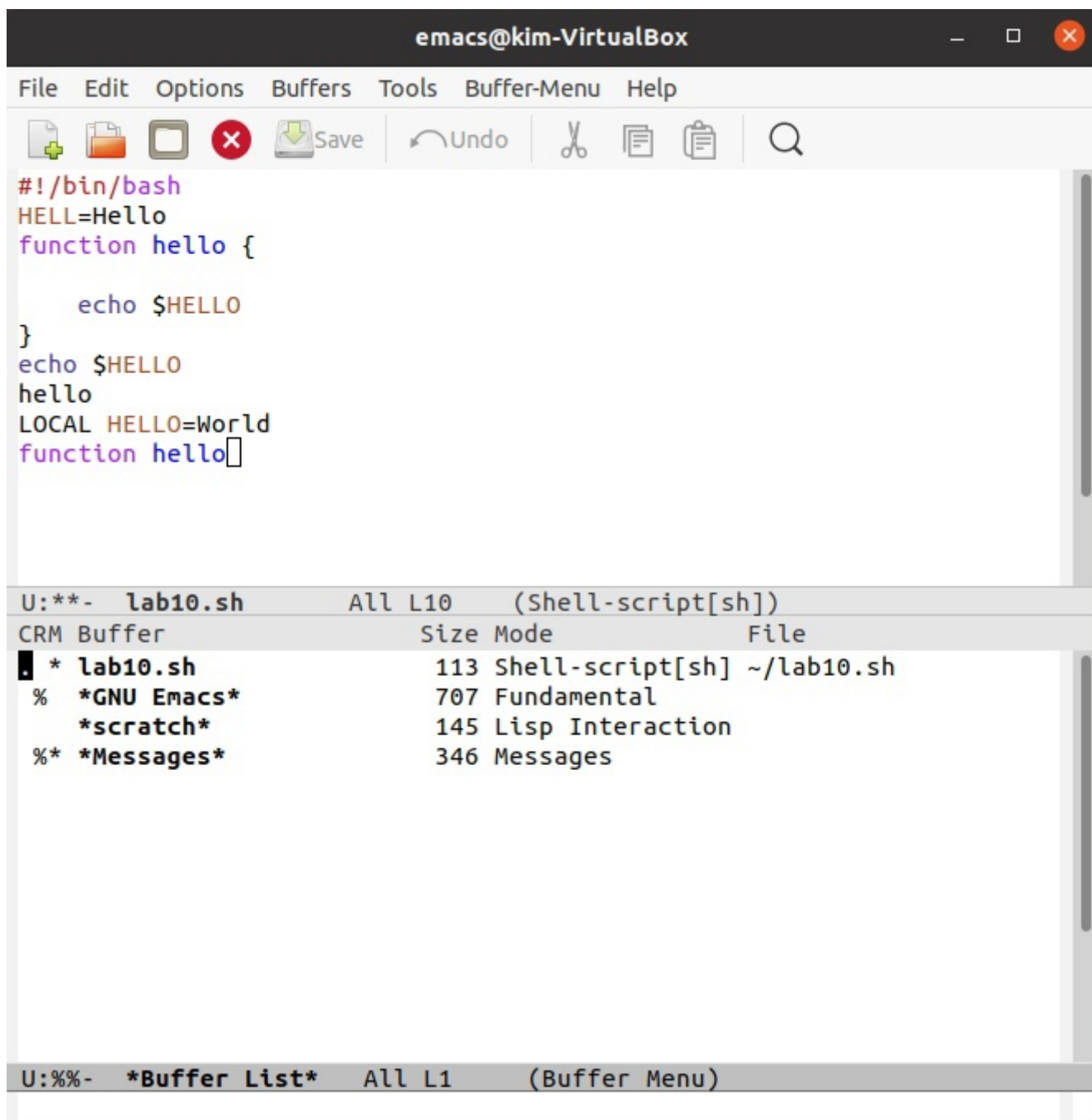
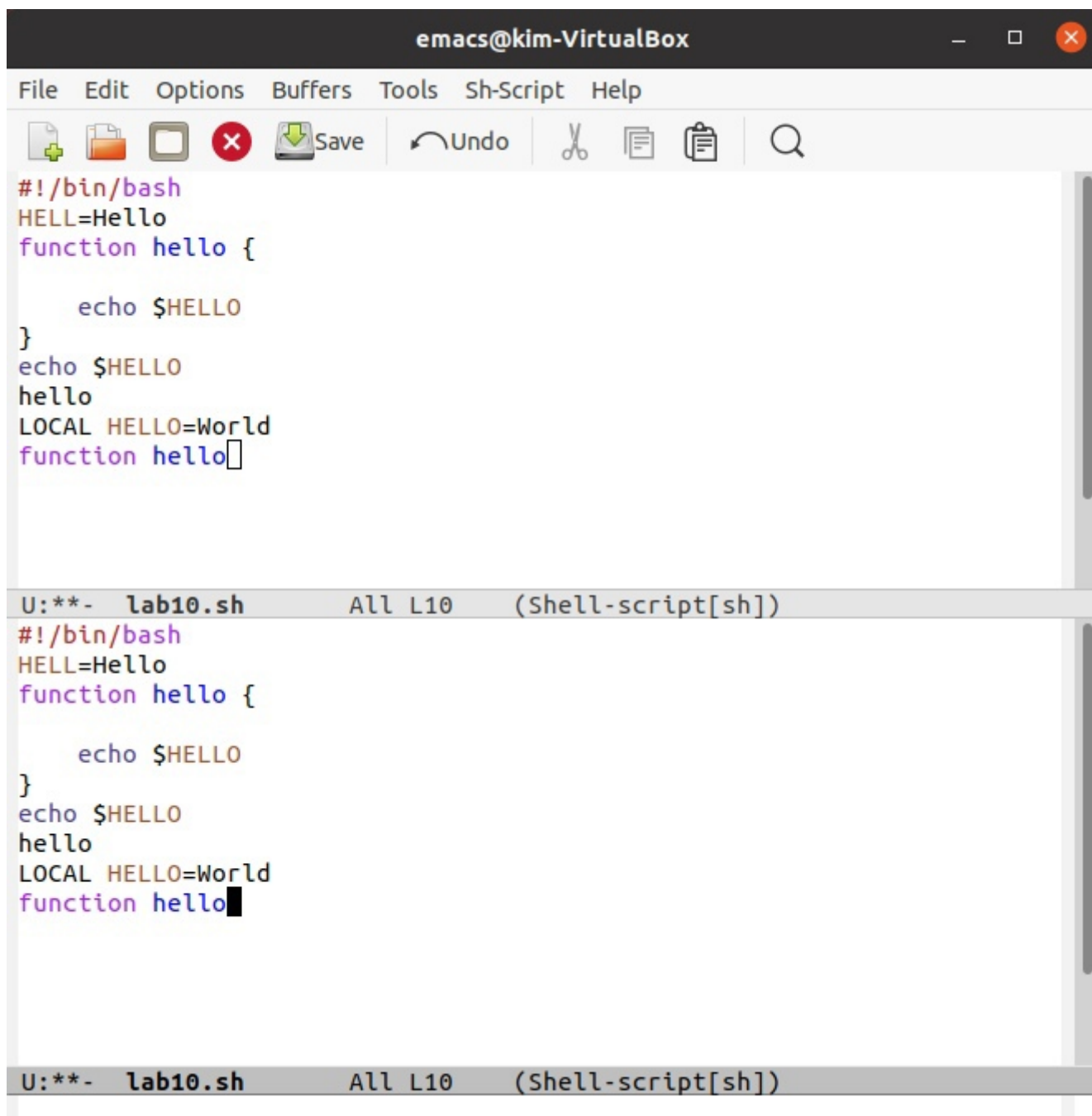
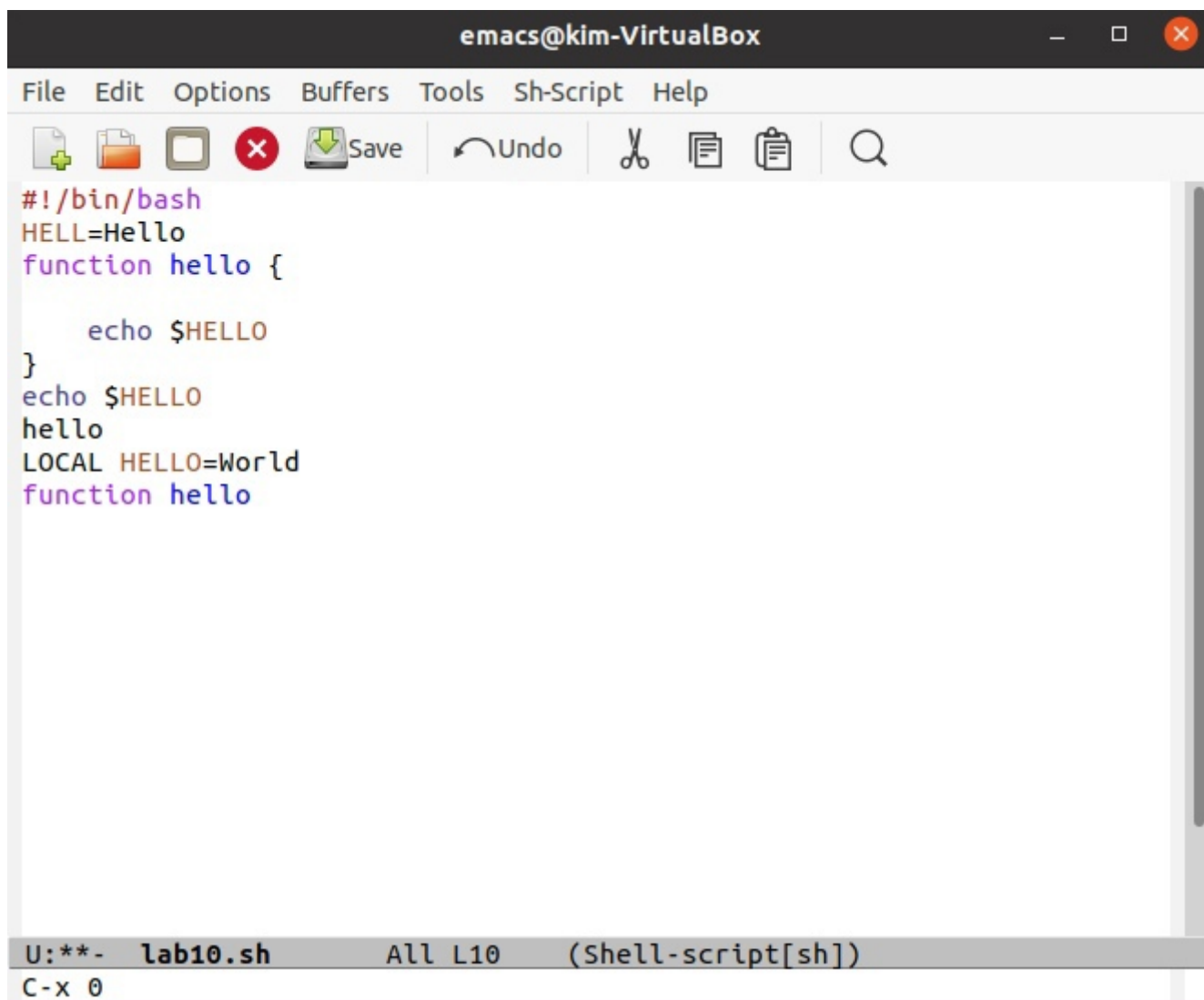


Рисунок 21: переключитесь на другой буфер



7.3. Закройте это окно с-х 0 (Рисунок 22).

Рисунок 22: Закройте окно

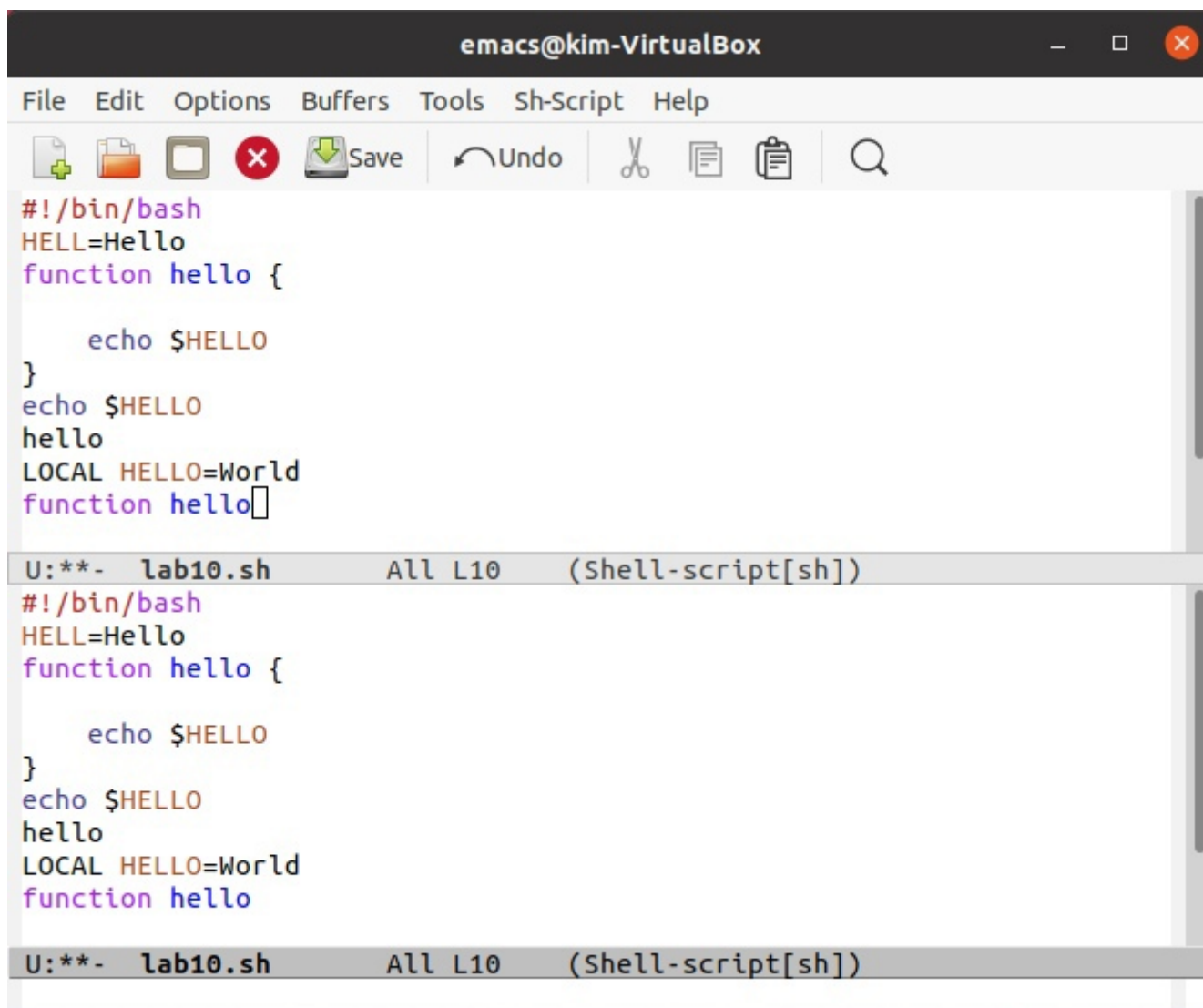


```
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
```

U:**- lab10.sh All L10 (Shell-script[sh])
C-x 0

7.4. Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран `C-x b` (Рисунок 23).

Рисунок 23: вновь переключайтесь между буферами



8. Управление окнами.

8.1. Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали `с-х 3` (Рисунок 24), а затем каждое из этих окон на две части по горизонтали `с-х 2` (Рисунок 25).

Рисунок 24: деление окна по вертикали

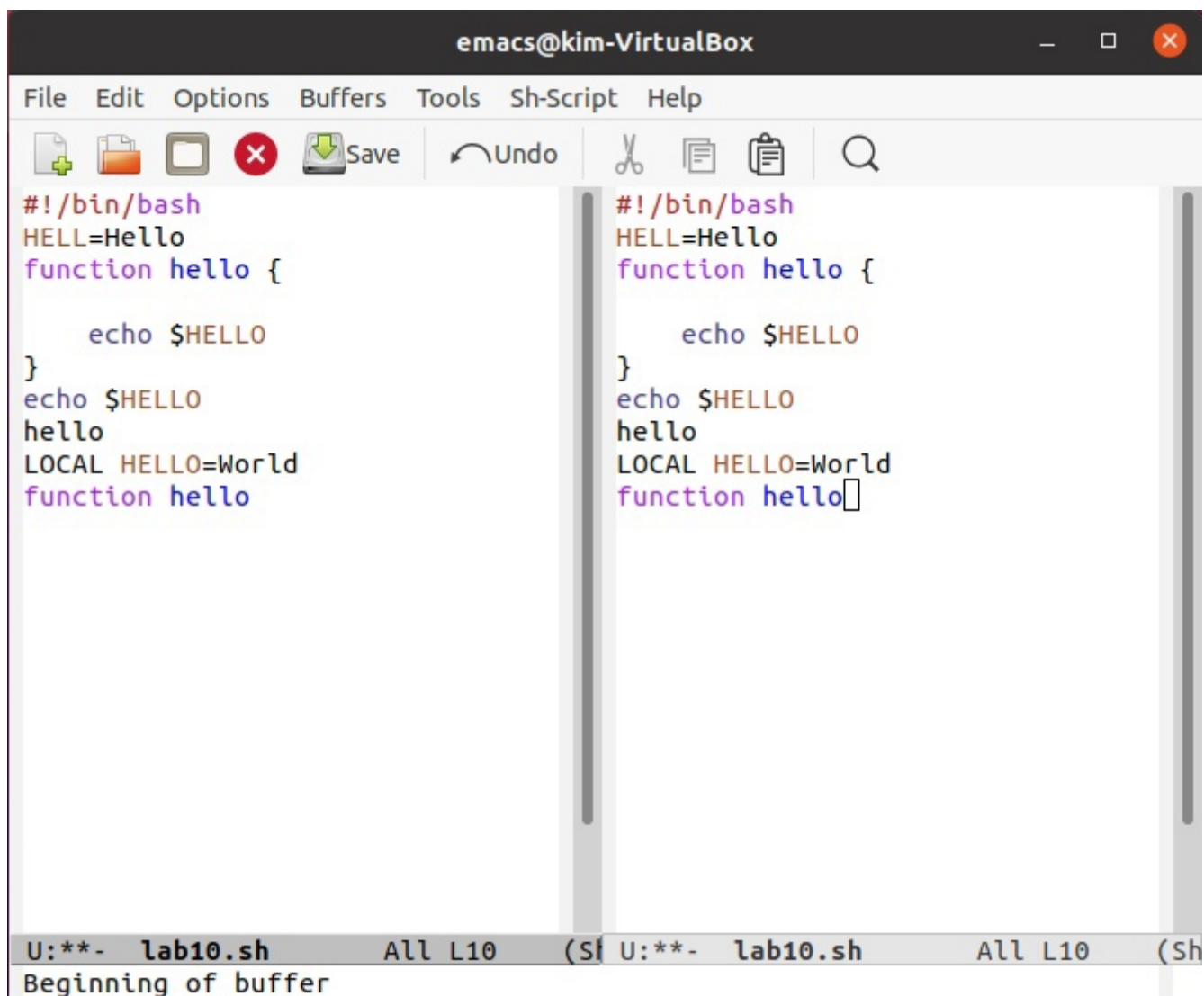
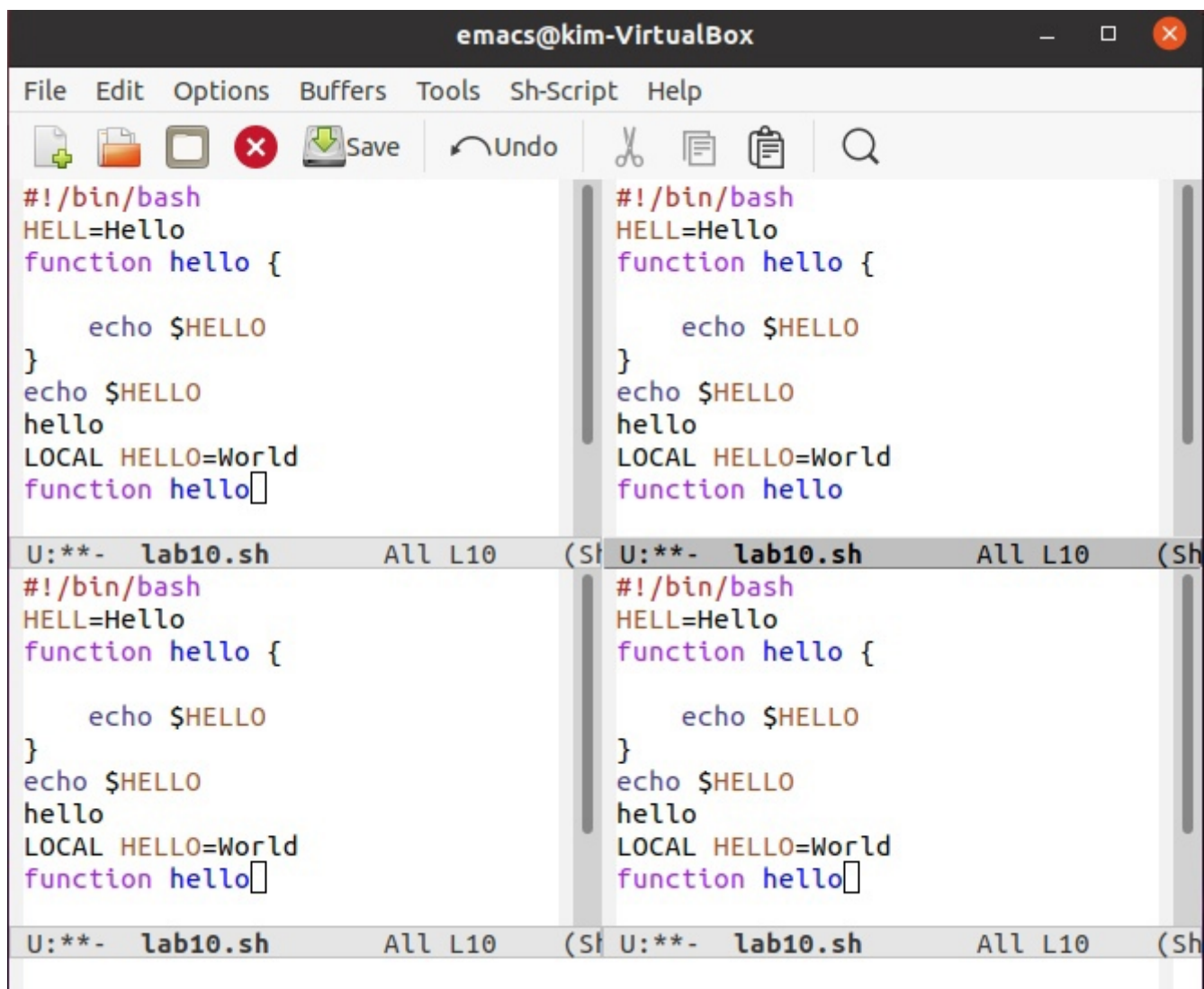


Рисунок 25: деление окна по вертикали



8.2. В каждом из четырёх созданных окон откройте новый буфер (файл) `new_file.txt` `c-x c-f` (Рисунок 26-27) и введите несколько строк текста (Рисунок 28)

Рисунок 26: открытие файла во всех окнах

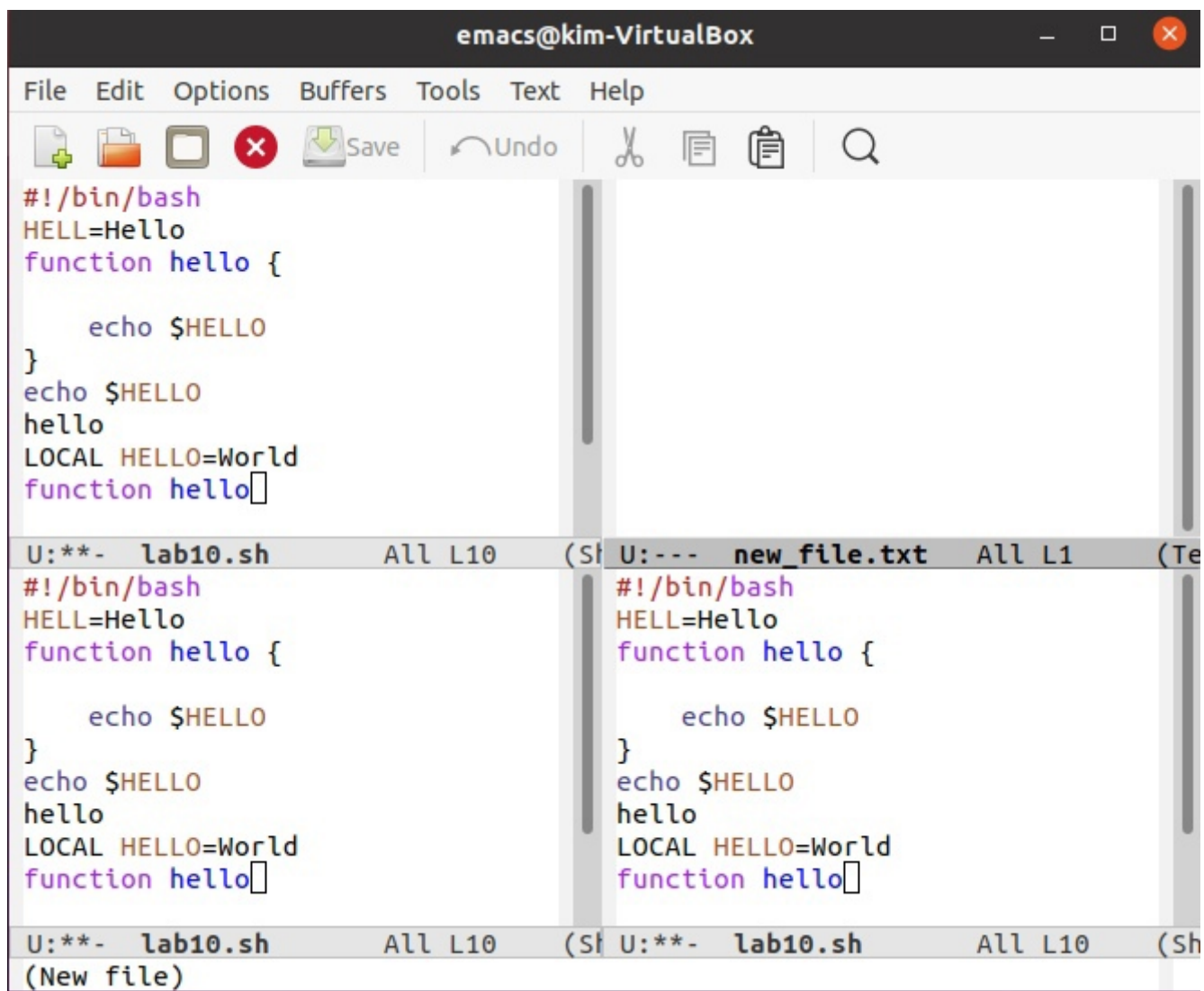


Рисунок 27: открытие файла во всех окнах

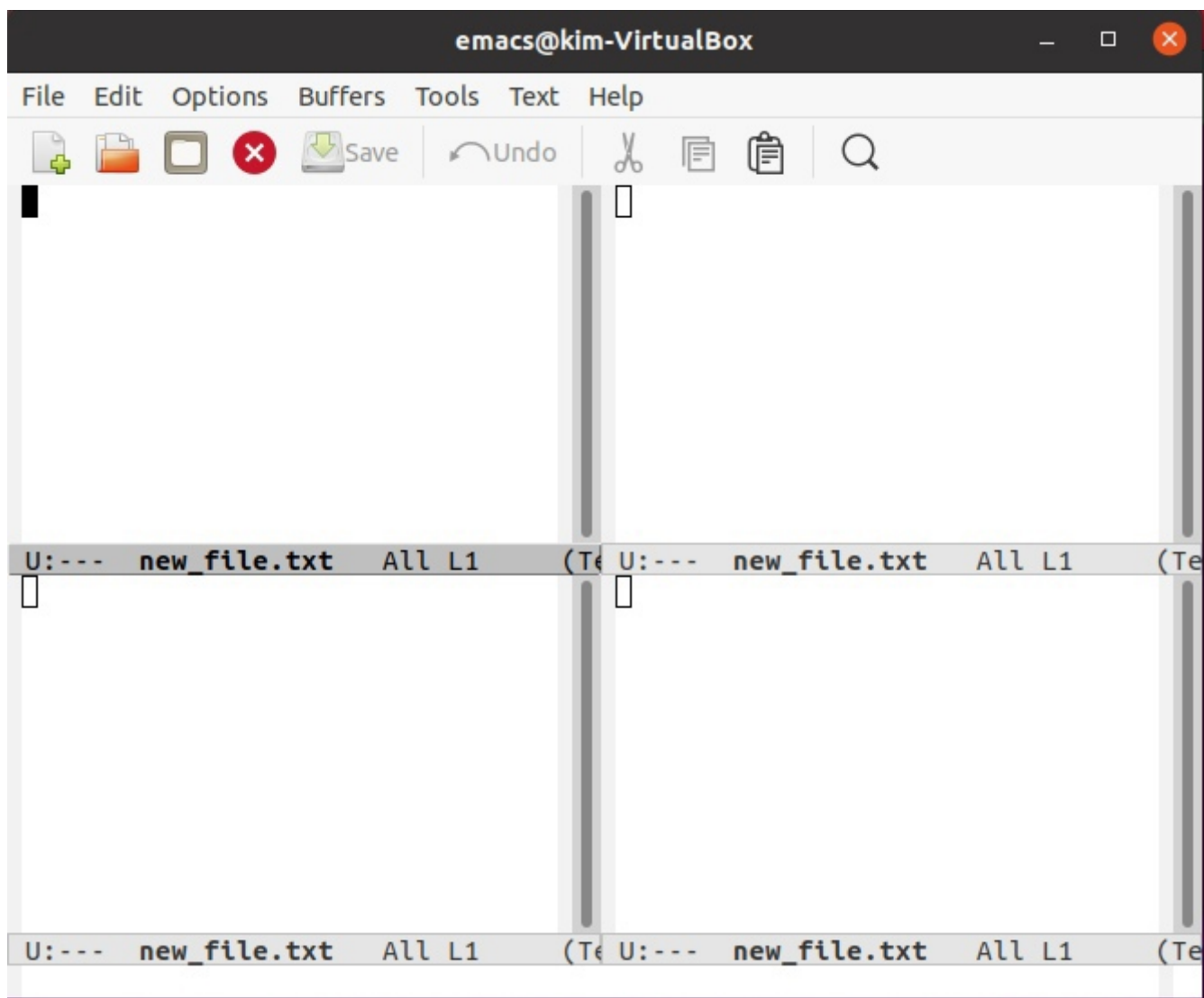
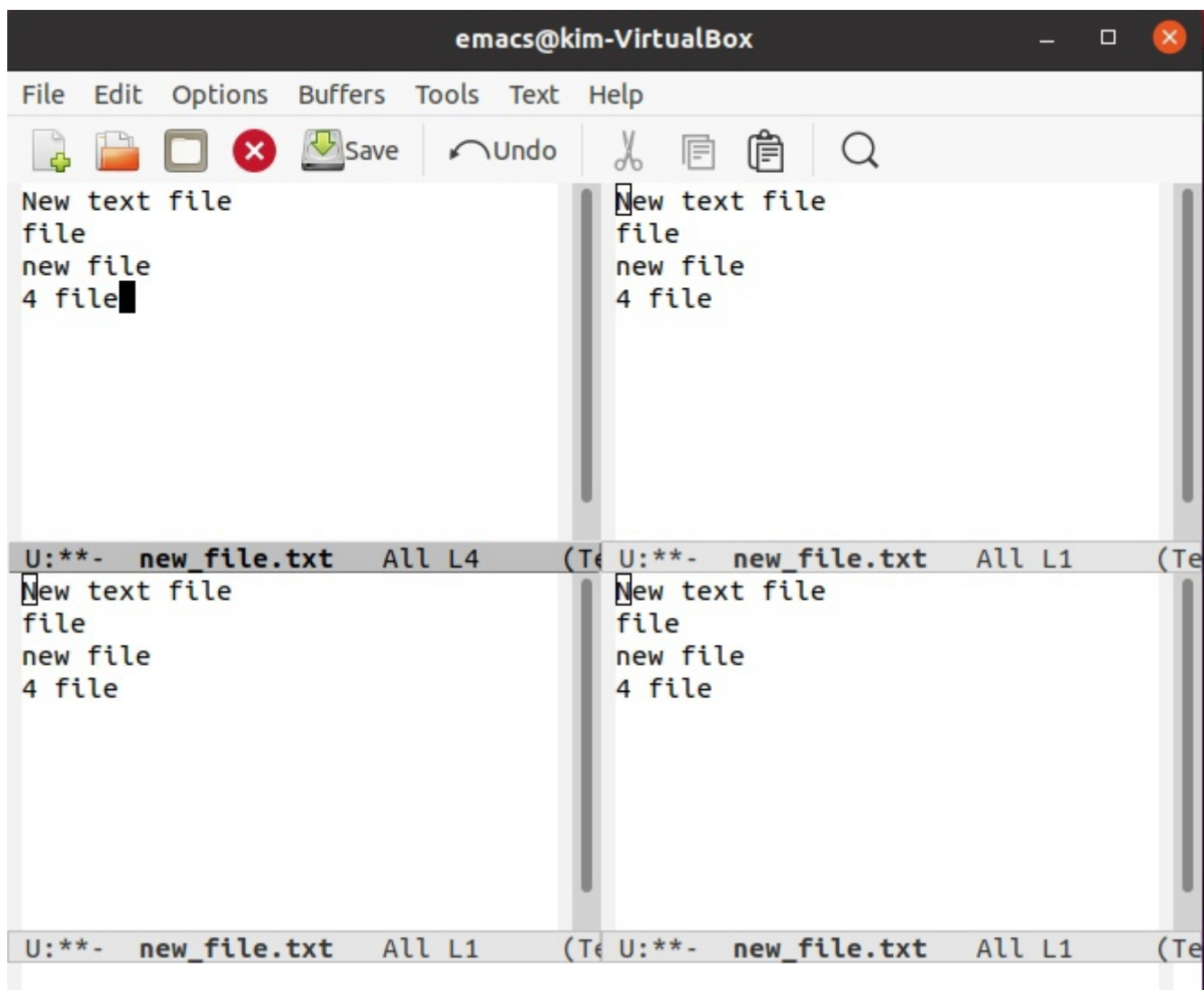


Рисунок 28: ввод текста



9. Режим поиска

9.1. Переключитесь в режим поиска `c-s` (Рисунок 29) и найдите несколько слов (Рисунок 30), присутствующих в тексте.

Рисунок 29: Переключитесь в режим поиска `c-s`

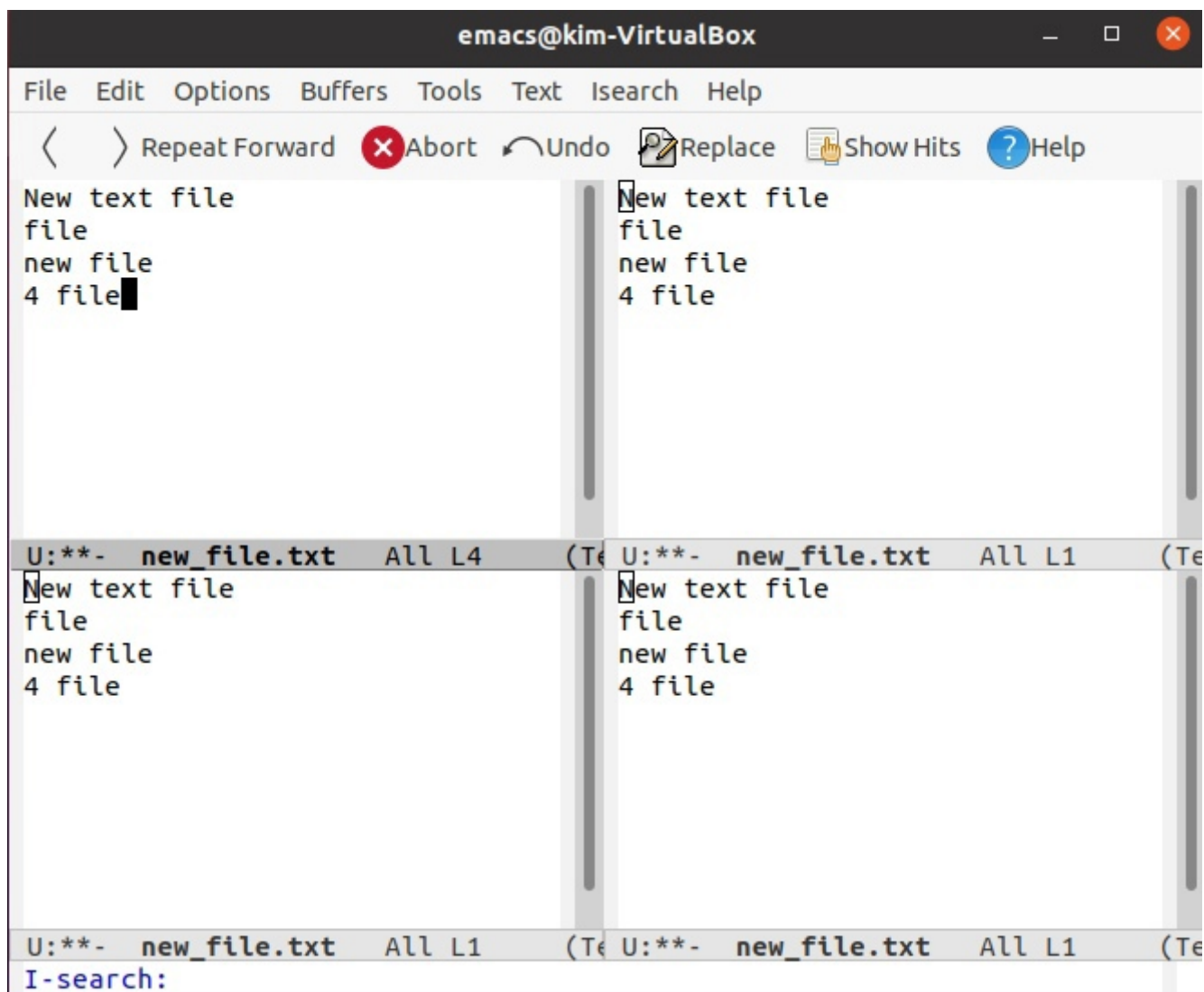
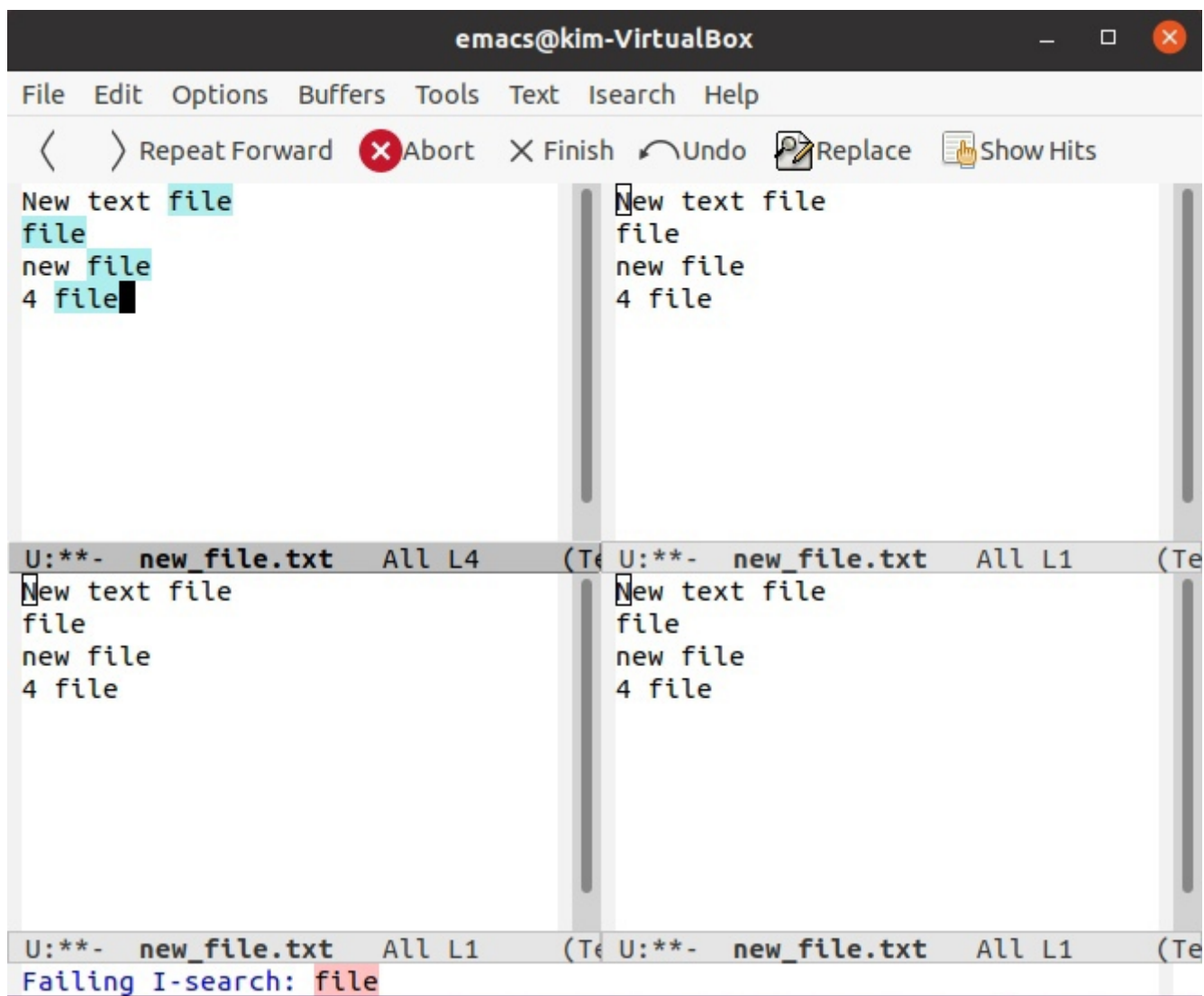


Рисунок 30: найдите несколько слов (file)



9.2. Переключайтесь между результатами поиска, нажимая `c-s` (Рисунок 31-32)

Рисунок 31: Переключайтесь между результатами поиска

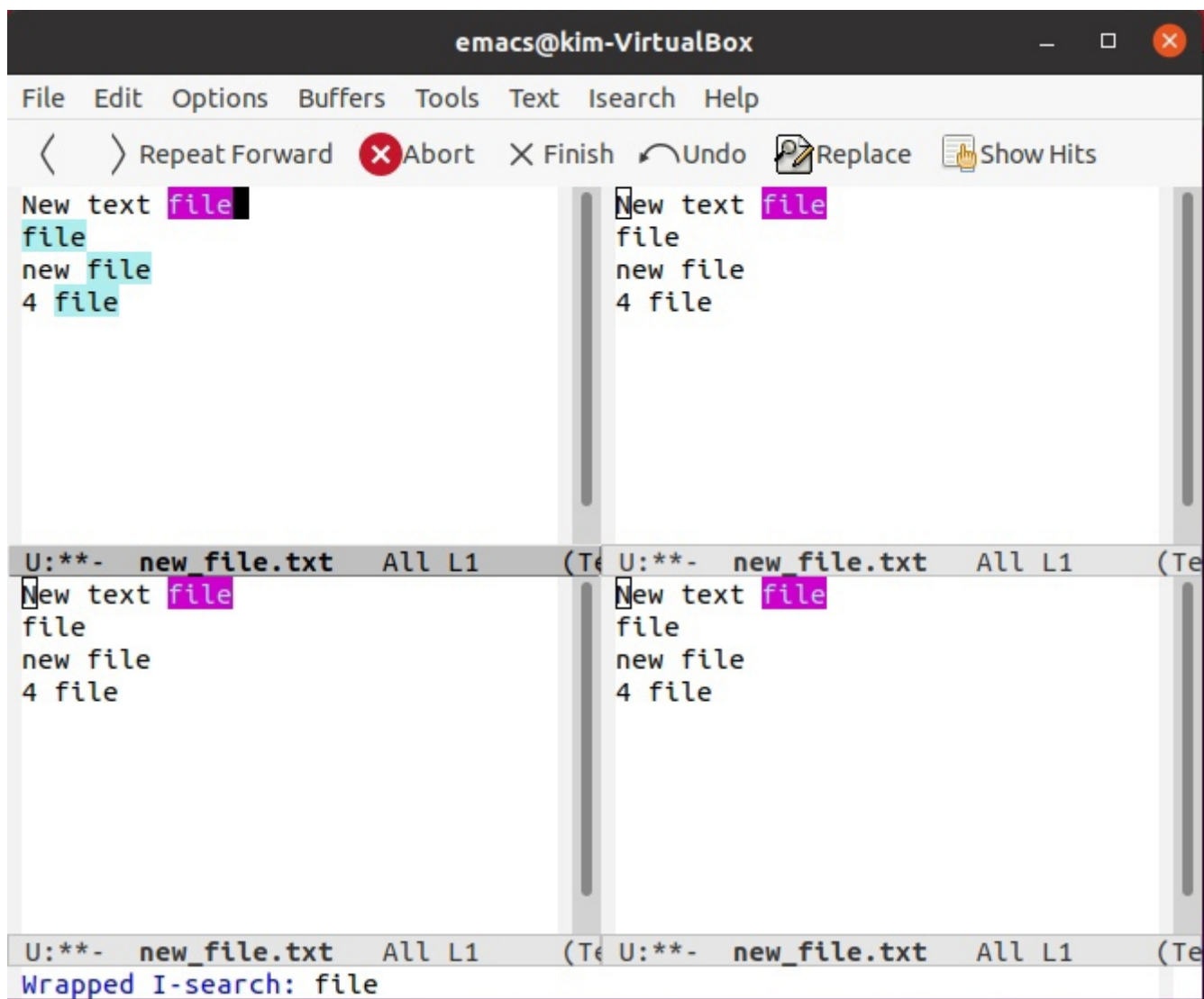
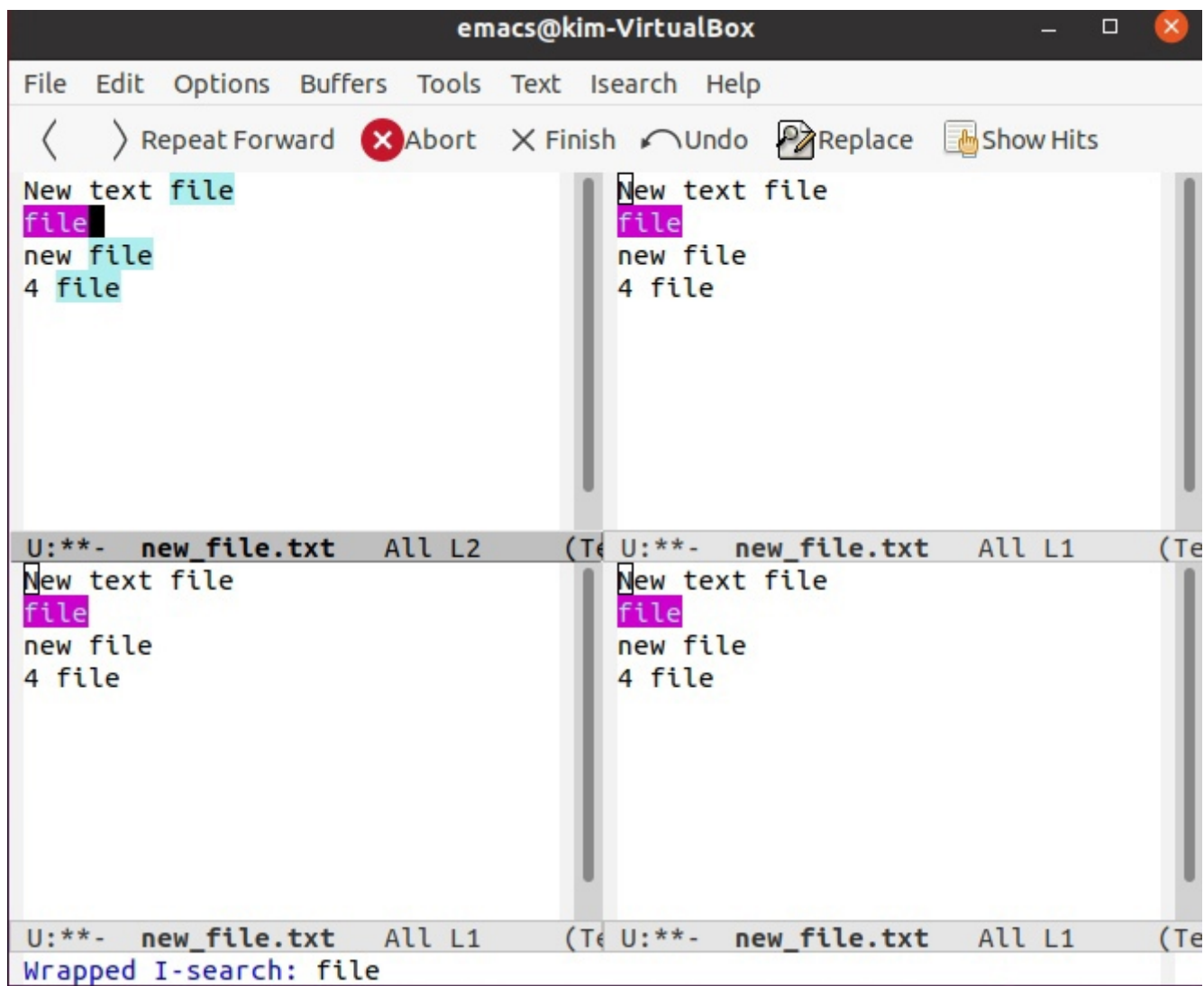
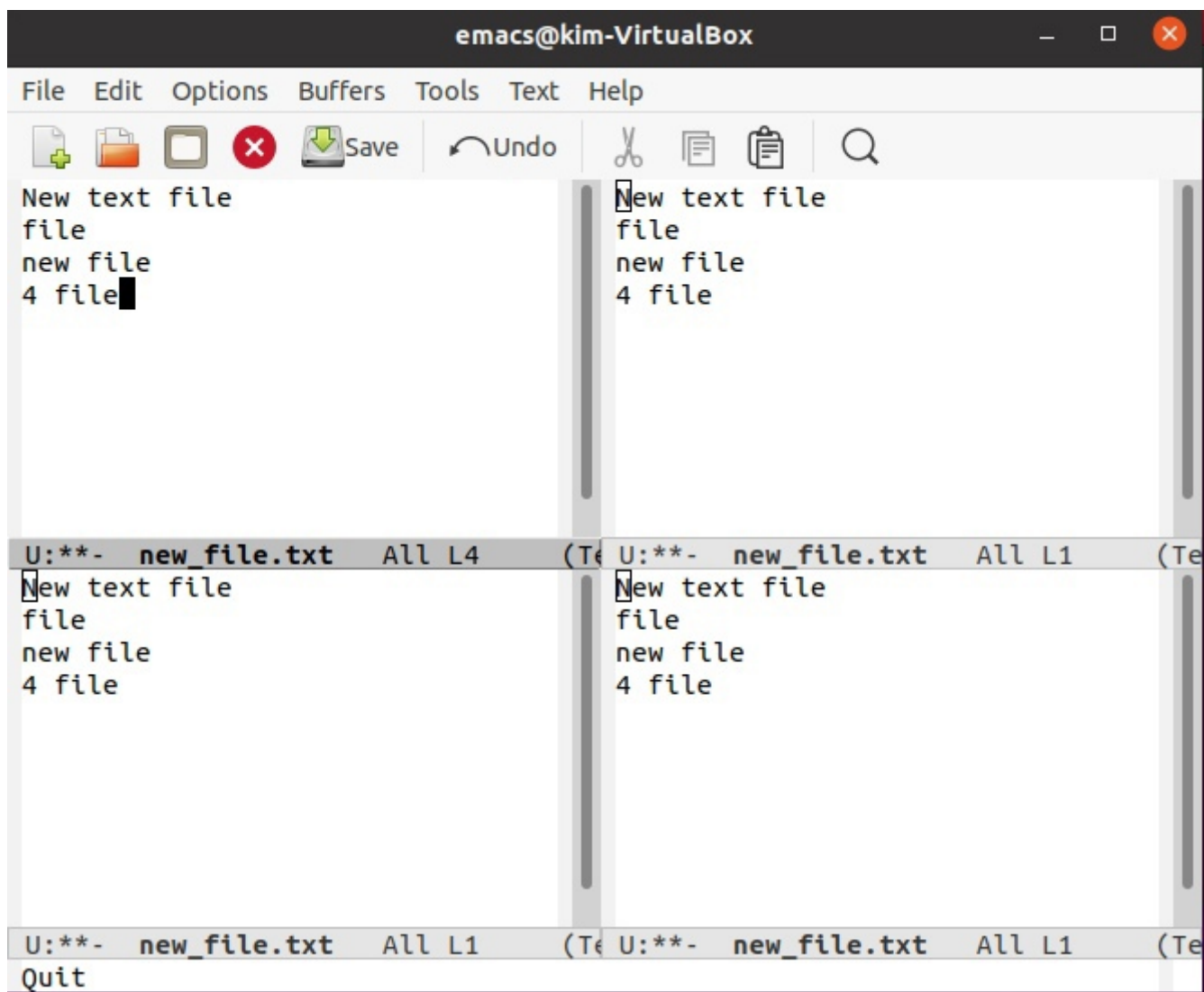


Рисунок 32: Переключайтесь между результатами поиска



9.3. Выйдите из режима поиска, нажав `с-g` (Рисунок 33)

Рисунок 33: Выйдите из режима поиска



9.4. Перейдите в режим поиска и замены `M-%`, введите текст, который следует найти и заменить, нажмите `Enter`, затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите `!` для подтверждения замены

Рисунок 34: Замените слово LOCAL на FILE и результаты

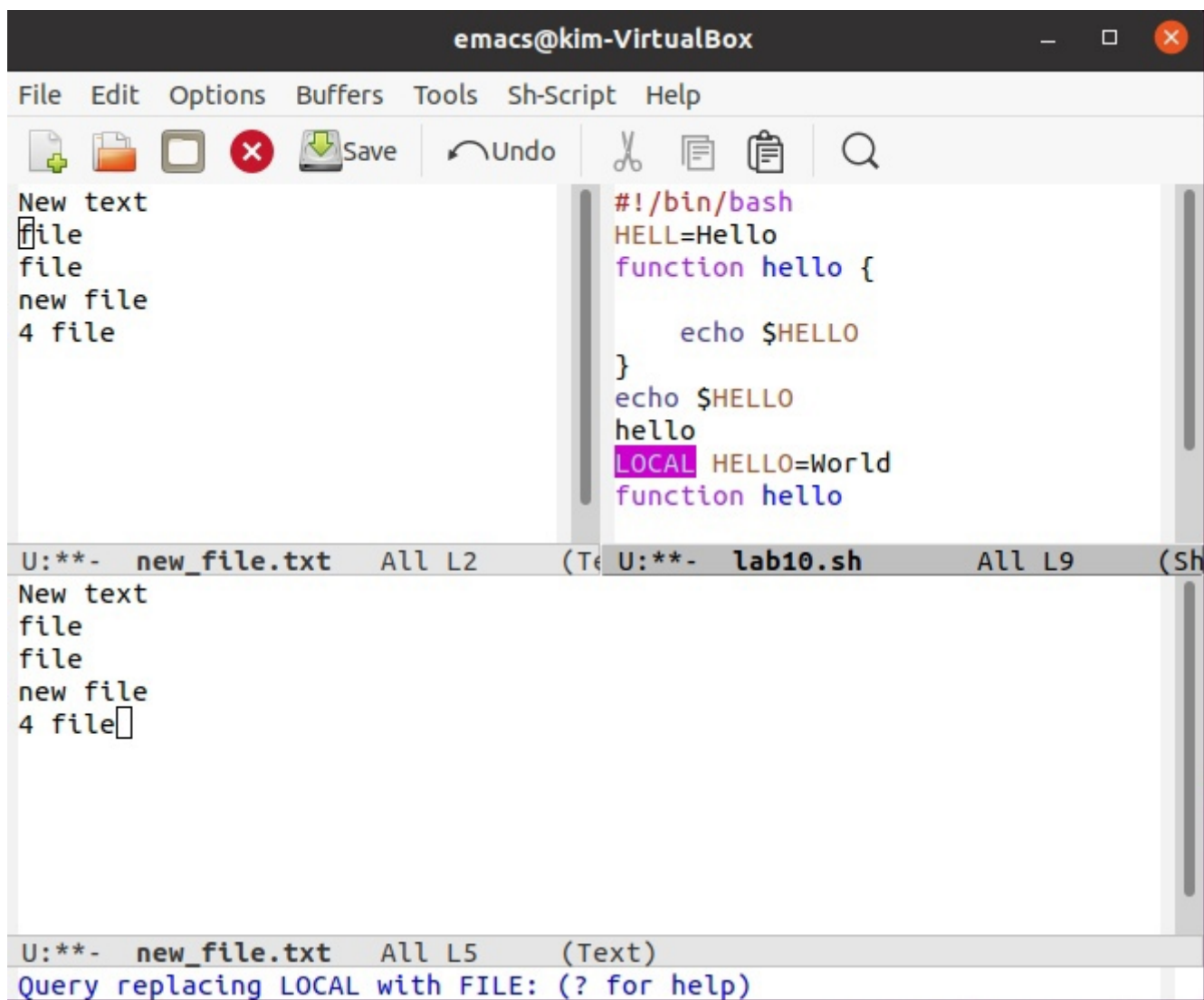
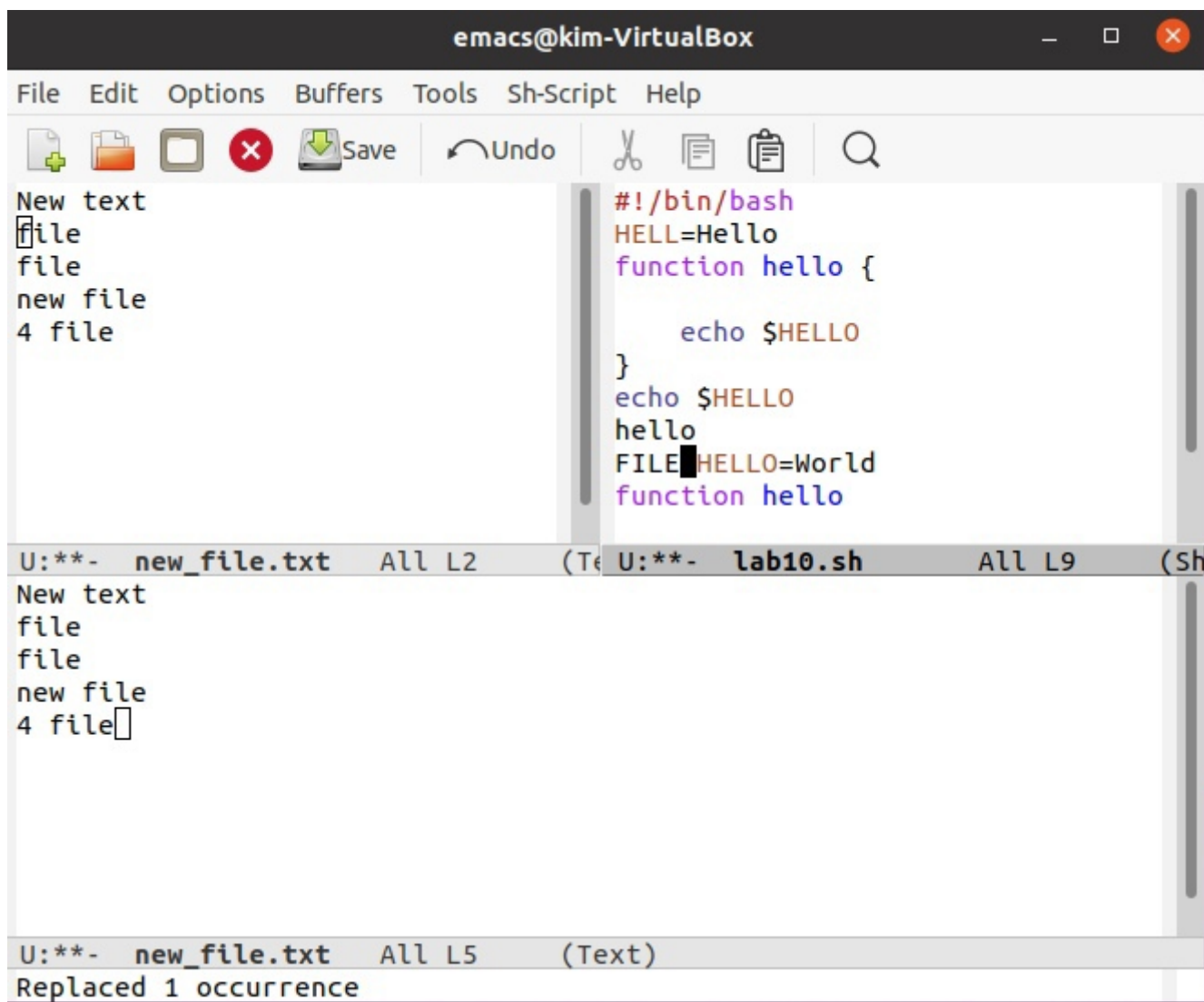


Рисунок 35: Замена



9.5. Испробуйте другой режим поиска, нажав `M-s o` в этом мы ищем слово `hello`, так как видим, что оно ищет не введенное слово, а строку, в которой оно встречается (Рисунок 36).
Рисунок 36:Испробуйте другой режим поиска `M-s o`

```
emacs@kim-VirtualBox
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
FILE HELLO=World
function hello

U:**- lab10.sh All L9 (Shell-script[sh])
7 matches for "hello" in buffer: lab10.sh
2:HELL=Hello
3:function hello {
5:    echo $HELLO
7:echo $HELLO
8:hello
9:FILE HELLO=World
10:function hello

U:%%- *Occur* All L1 (Occur)
Searched 1 buffer; 7 matches for "hello"
```

Вывод:

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

Библиография:

- [1] [Лабораторная №10](#)
- [2] [Термин Emacs](#)