

Отчет по лабораторной работе №10

Тема:

Текстовой редактор emacs

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: Операционные системы

Студент: Байрамгельдыев Довлетмурат

Группа: НФИбд-03-20

Москва, 2021г.

Цель работы

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

Введение

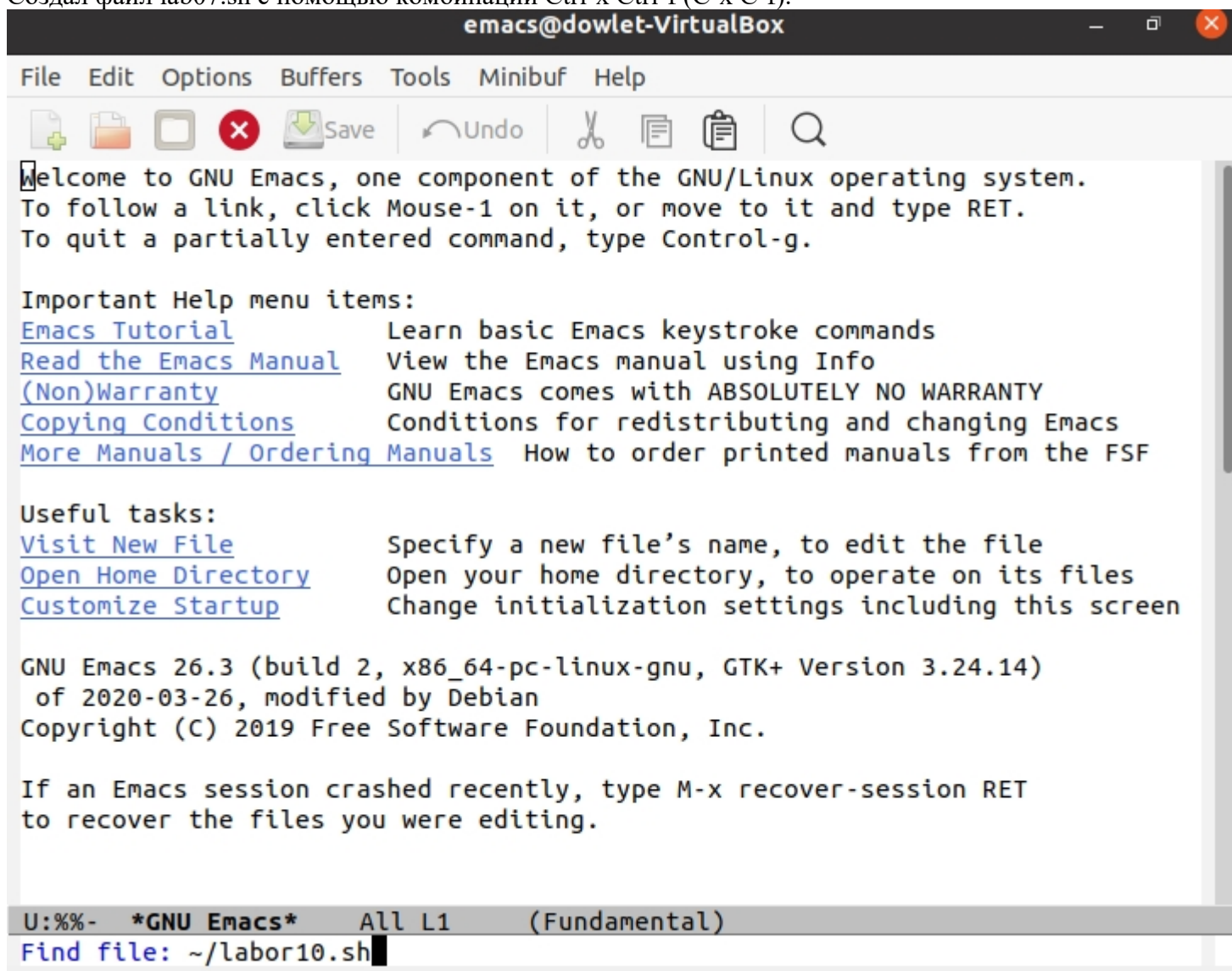
Emacs — семейство многофункциональных расширяемых текстовых редакторов. Поскольку Emacs появился давно и изначально работал только на текстовых терминалах, его терминология не совсем совпадает с таковой современных оконных систем. Основные специфические термины Emacs: Буфер — объект, представляющий какой-либо текст. Большинство буферов соответствуют открытым файлам, но в принципе буфер может содержать что угодно, например результаты компиляции программы или встроенные подсказки. Emacs в этом плане очень однороден и практически всё общение с пользователем, в том числе интерактивное, идёт через буферы. Фрейм соответствует окну в обычном понимании этого слова (как у X Window System). Каждый фрейм содержит область вывода и одно или несколько окон Emacs. Окно — это просто прямоугольная область фрейма, которая отображает один из буферов. Помимо этого, каждое окно имеет свою строку состояния (англ. mode line), где выводится название буфера, его основной режим и т. п. Область вывода (echo area) — одна или несколько (по необходимости) строк внизу фрейма, в которой Emacs выводит различные сообщения, а также запрашивает подтверждения и дополнительную информацию от пользователя. Минибуфер используется для ввода дополнительной информации. Он всегда отображается в области вывода. Это такой же буфер, как и все остальные, что даёт интересный эффект: например, при операции замены текста можно ввести многострочный текст, что невозможно в большинстве других редакторов.

Последовательность выполнения работы

1. Ознакомиться с теоретическим материалом.
 2. Ознакомиться с редактором emacs.
 3. Выполнить упражнения.
 4. Ответить на контрольные вопро
-

Ход работы:

1. Открыл emacs. `dowlet@dowlet-VirtualBox:~$ emacs`
2. Создал файл lab07.sh с помощью комбинаций Ctrl-x Ctrl-f (C-x C-f).



3. Набрал текст: `#!/bin/bash HELL=Hello function hello { LOCAL HELLO=World echo $HELLO } echo $HELLO
hello`

File Edit Options Buffers Tools Sh-Script Help



Save

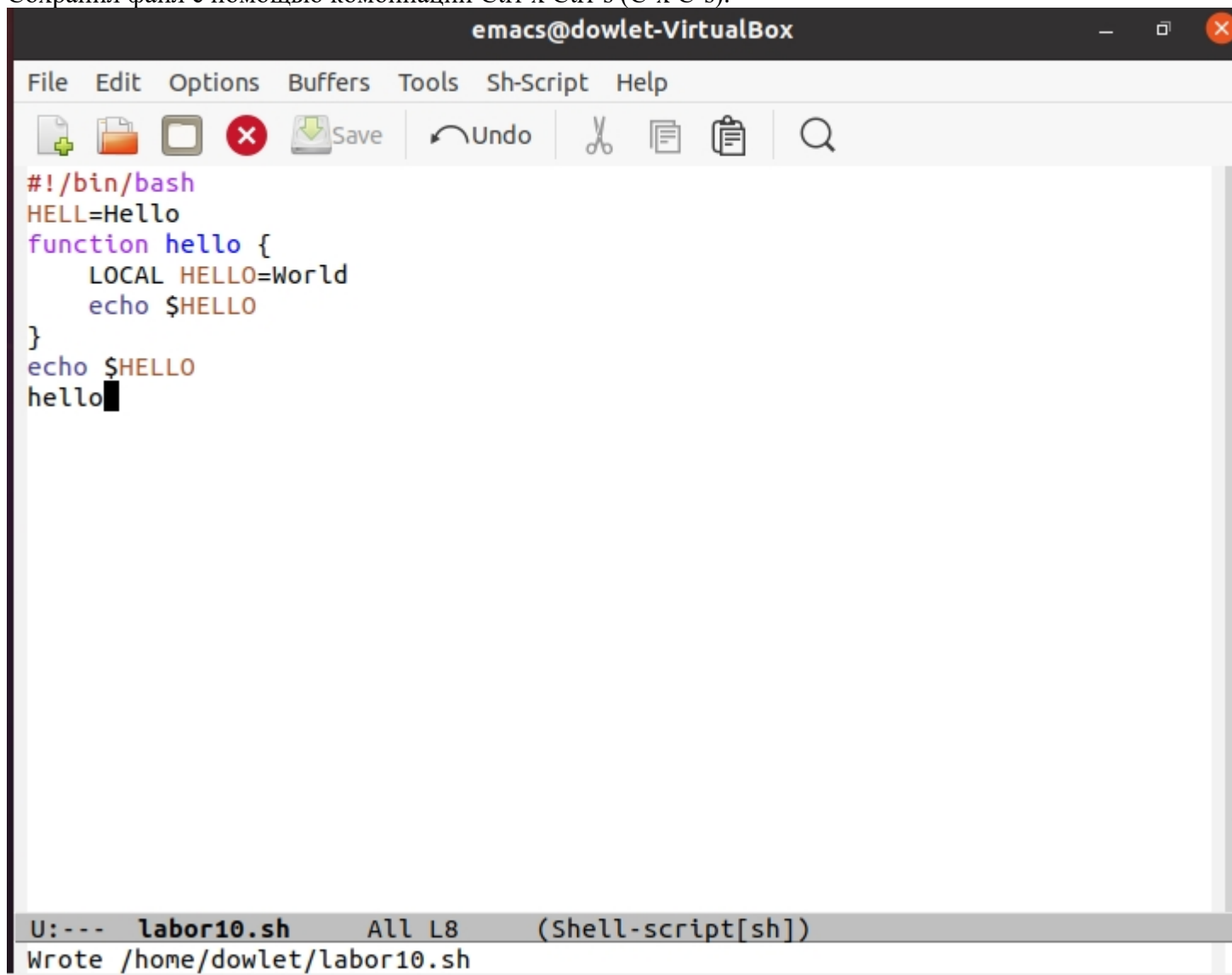
Undo



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

U:**- labor10.sh All L8 (Shell-script[sh])

4. Сохранил файл с помощью комбинаций Ctrl-x Ctrl-s (C-x C-s).



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

U: --- labor10.sh All L8 (Shell-script[sh])
Wrote /home/dowlet/labor10.sh

5. Проделал с текстом стандартные процедуры редактирования, каждое действие осуществлял комбинацией клавиш. • Вырезал одной командой целую строку (C-k).

File Edit Options Buffers Tools Sh-Script Help



Save



Undo













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

U:**- labor10.sh All L7 (Shell-script[sh])

- Вставил эту строку в конец файла (C-y).

emacs@dowlet-VirtualBox

File Edit Options Buffers Tools Sh-Script Help

 Save  Undo 

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






U:**- labor10.sh All L8 (Shell-script[sh])


Mark set





- Выделил область текста (C-space).

emacs@dowlet-VirtualBox

FileEditOptionsBuffersToolsSh-ScriptHelp

 Save

 Undo



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

Geany


-:--- labor10.shAll L8(Shell-script[bash])

Mark set

- Скопировал область в буфер обмена (M-w).
- Вставил область в конец файла.

emacs@dowlet-VirtualBox

File Edit Options Buffers Tools Sh-Script Help



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

U:**- labor10.sh All L9 (Shell-script[sh])

Mark set

Вновь выделил эту область и на этот раз вырезал её (C-w).

File Edit Options Buffers Tools Sh-Script Help



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

U:**- labor10.sh All L9 (Shell-script[sh])

Mark set

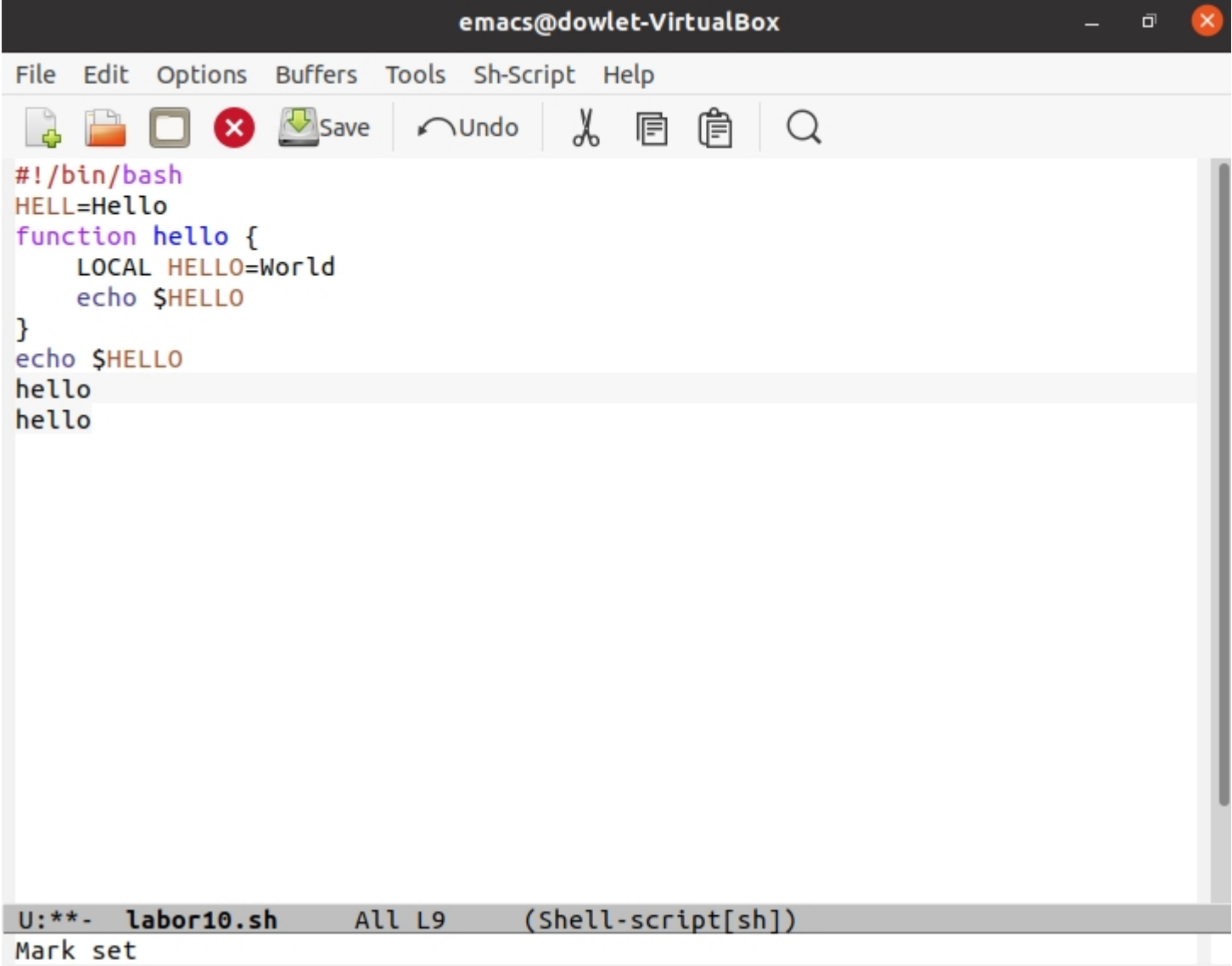
File Edit Options Buffers Tools Sh-Script Help



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

U:**- labor10.sh All L8 (Shell-script[sh])

- Отменил последнее действие (C-/).



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

U:**- labor10.sh All L9 (Shell-script[sh])
Mark set

1. Научился использовать команды по перемещению курсора. • Переместите курсор в начало строки (C-a).



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

U:**- labor10.sh All L9 (Shell-script[sh])

Переместите курсор в конец строки (C-e).

File Edit Options Buffers Tools Sh-Script Help




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

U:**- labor10.sh All L9 (Shell-script[sh])

- Переместите курсор в начало буфера (M-<).

emacs@dowlet-VirtualBox

File Edit Options Buffers Tools Sh-Script Help



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

Ubuntu Software


-:***- labor10.sh All L1 (Shell-script[bash])

Beginning of buffer

• Переместите курсор в конец буфера (M->).

emacs@dowlet-VirtualBox

File Edit Options Buffers Tools Sh-Script Help



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

Help











-:***- labor10.sh All L9 (Shell-script[bash])

End of buffer

7. Управление буферами. • Вывел список активных буферов на экран (C-x C-b).

emacs@dowlet-VirtualBox

File Edit Options Buffers Tools Sh-Script Help

 Save  Undo 

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

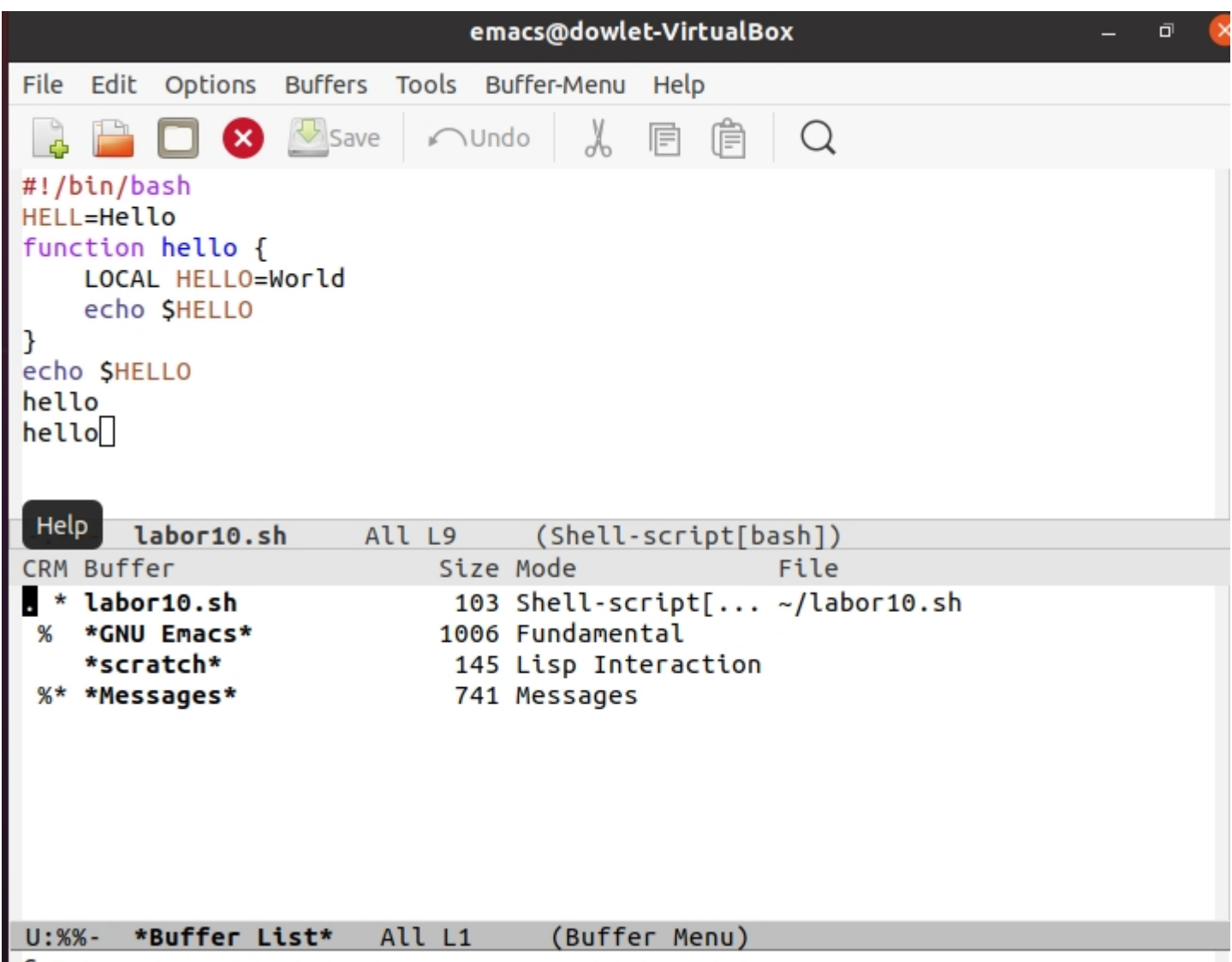
U:***- labor10.sh All L9 (Shell-script[sh])

CRM	Buffer	Size	Mode	File
1	* labor10.sh	104	Shell-script[sh]	~/labor10.sh
%	*GNU Emacs*	1006	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	851	Messages	

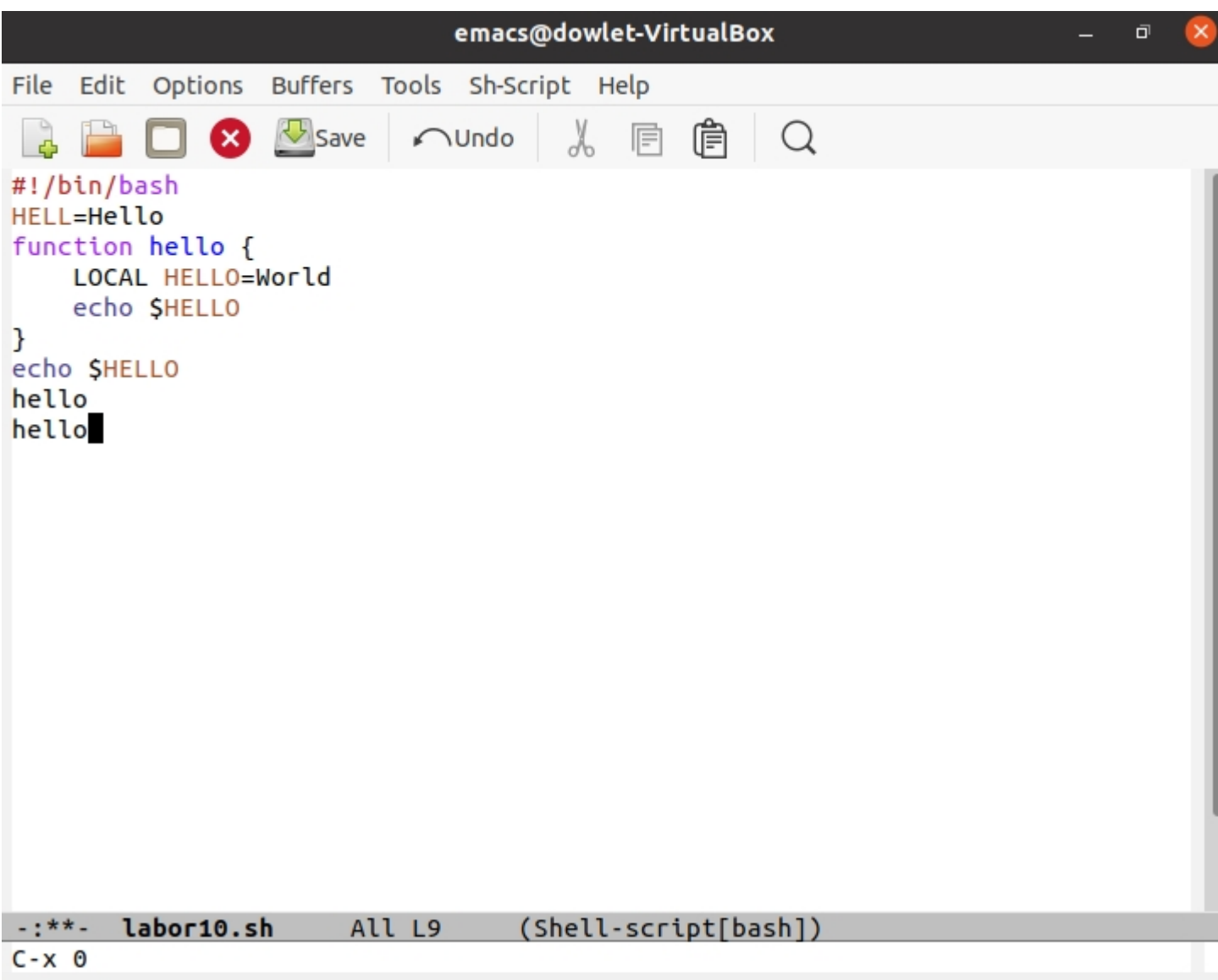
U:%%- *Buffer List* All L1 (Buffer Menu)

C-x C-b

- Переместился во вновь открытое окно (C-x) о со списком открытых буферов и переключился на другой буфер.



- Закрыл это окно (C-x 0).

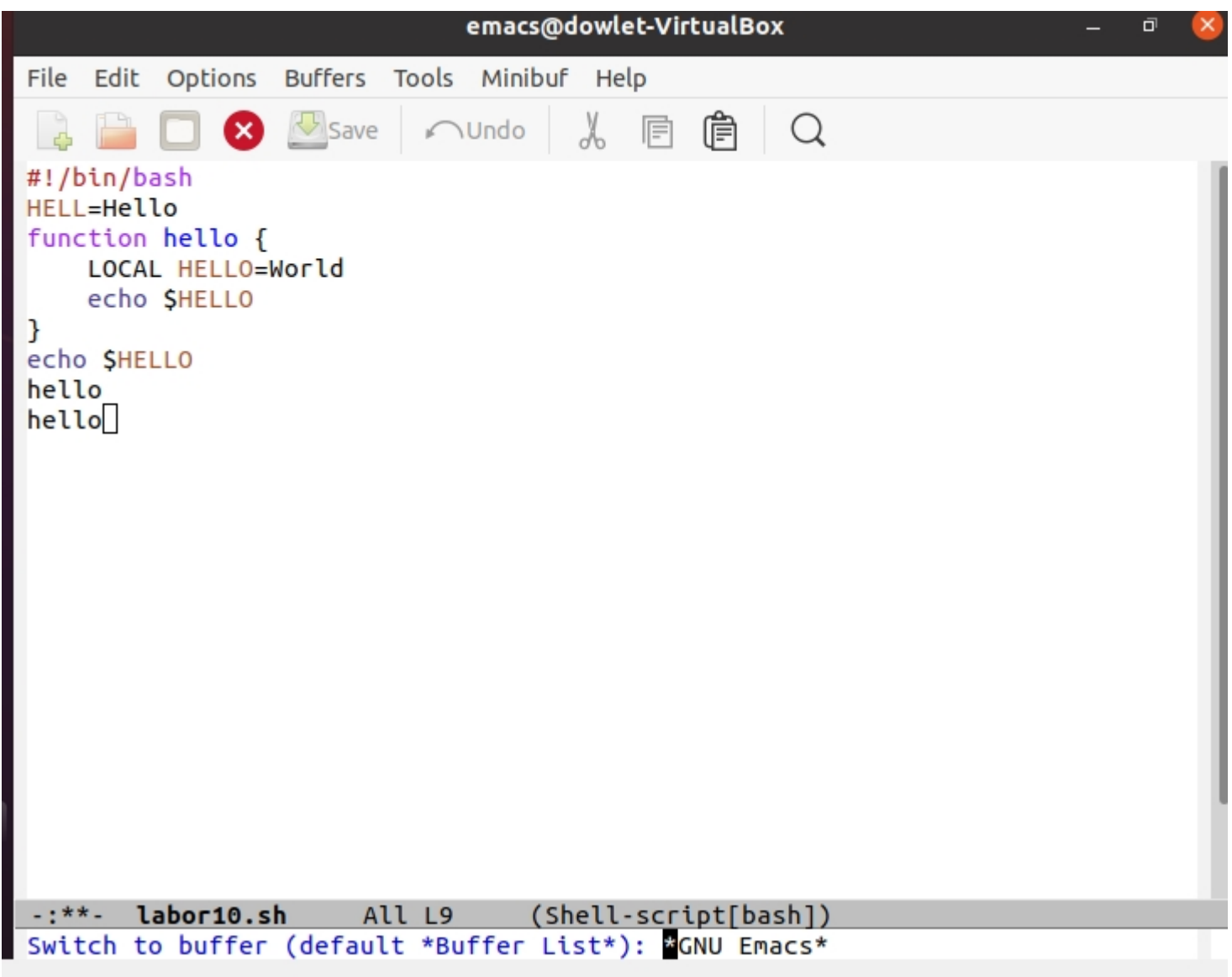


The screenshot shows the Emacs editor interface within a virtual box titled "emacs@dowlet-VirtualBox". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for file operations (add, open, close, save), editing (undo, redo, cut, copy, paste), and search. The main text area contains a shell script with the following content:

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

The status bar at the bottom displays the file name "labor10.sh", the buffer "All L9", and the shell type "(Shell-script[bash])". The command prompt "C-x 0" is visible.

- Вновь переключился между буферами, но уже без вывода их списка на экран (C-x b).



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

-:***- labor10.sh All L9 (Shell-script[bash])
Switch to buffer (default *Buffer List*): *GNU Emacs*

8. Управление окнами. • Поделил фрейм на 4 части: разделил фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2)

File Edit Options Buffers Tools Sh-Script Help



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

[Help](#)

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

-:***- labor10.sh All L9 (Shell- -:***- labor10.sh All L9 (Shell
C-x 3

File Edit Options Buffers Tools Sh-Script Help



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

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

-:***- labor10.sh All L9 (Shell-

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

-:***- labor10.sh All L9 (Shel

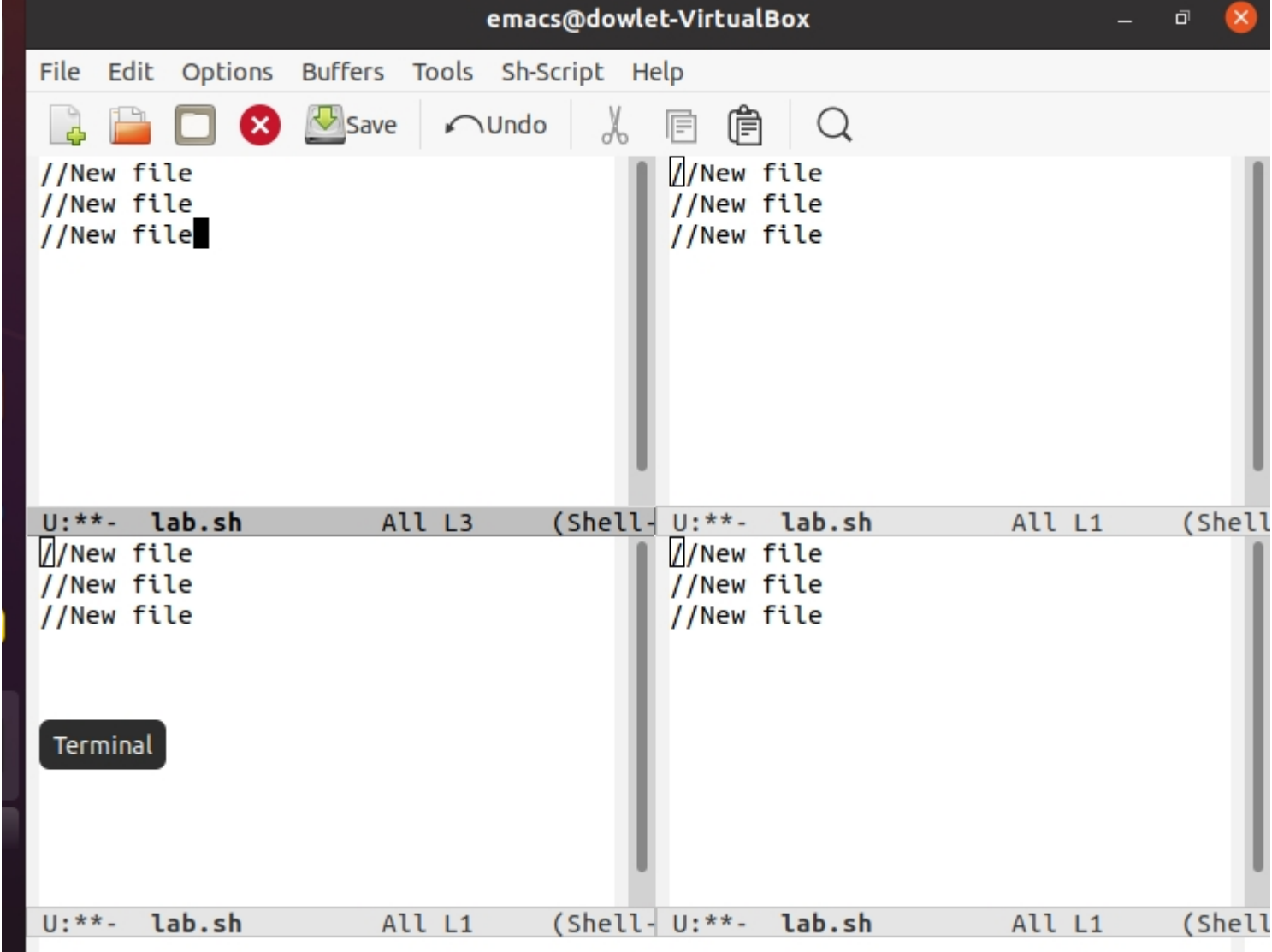
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
hello
```

-:***- labor10.sh All L9 (Shell-

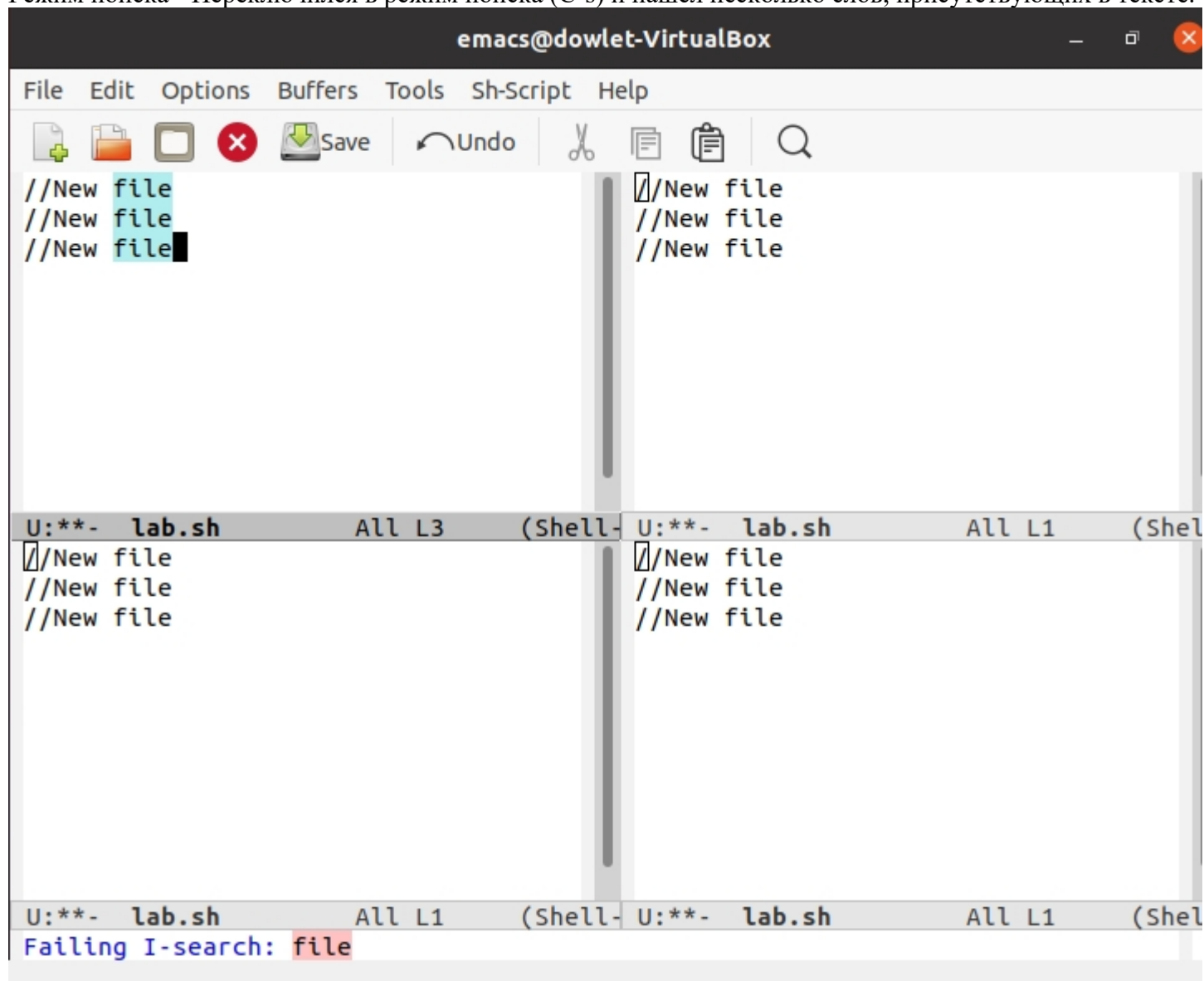
-:***- labor10.sh All L9 (Shel

C-x 2

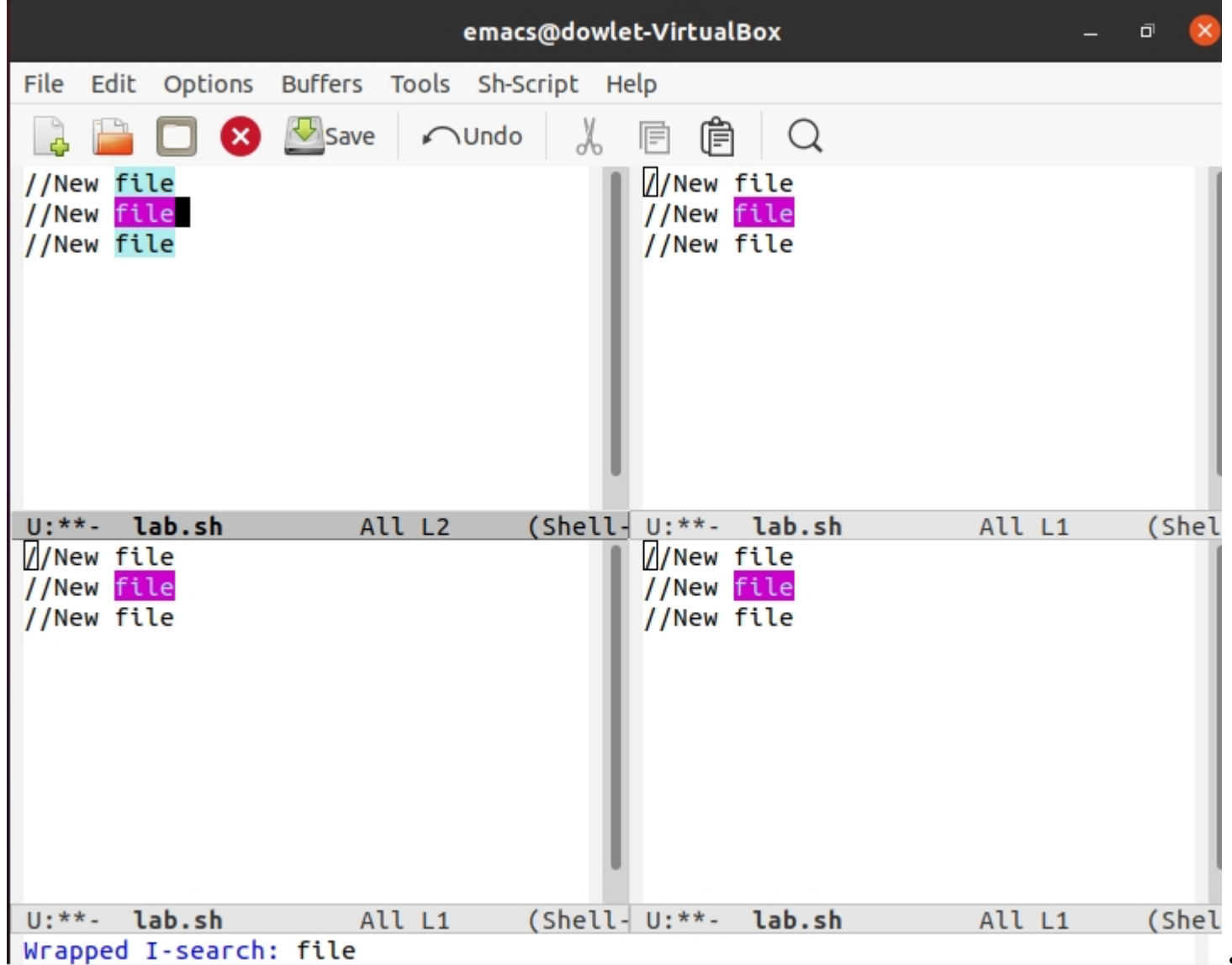
- В каждом из четырёх созданных окон открыл новый буфер (файл) и ввел несколько строк текста.



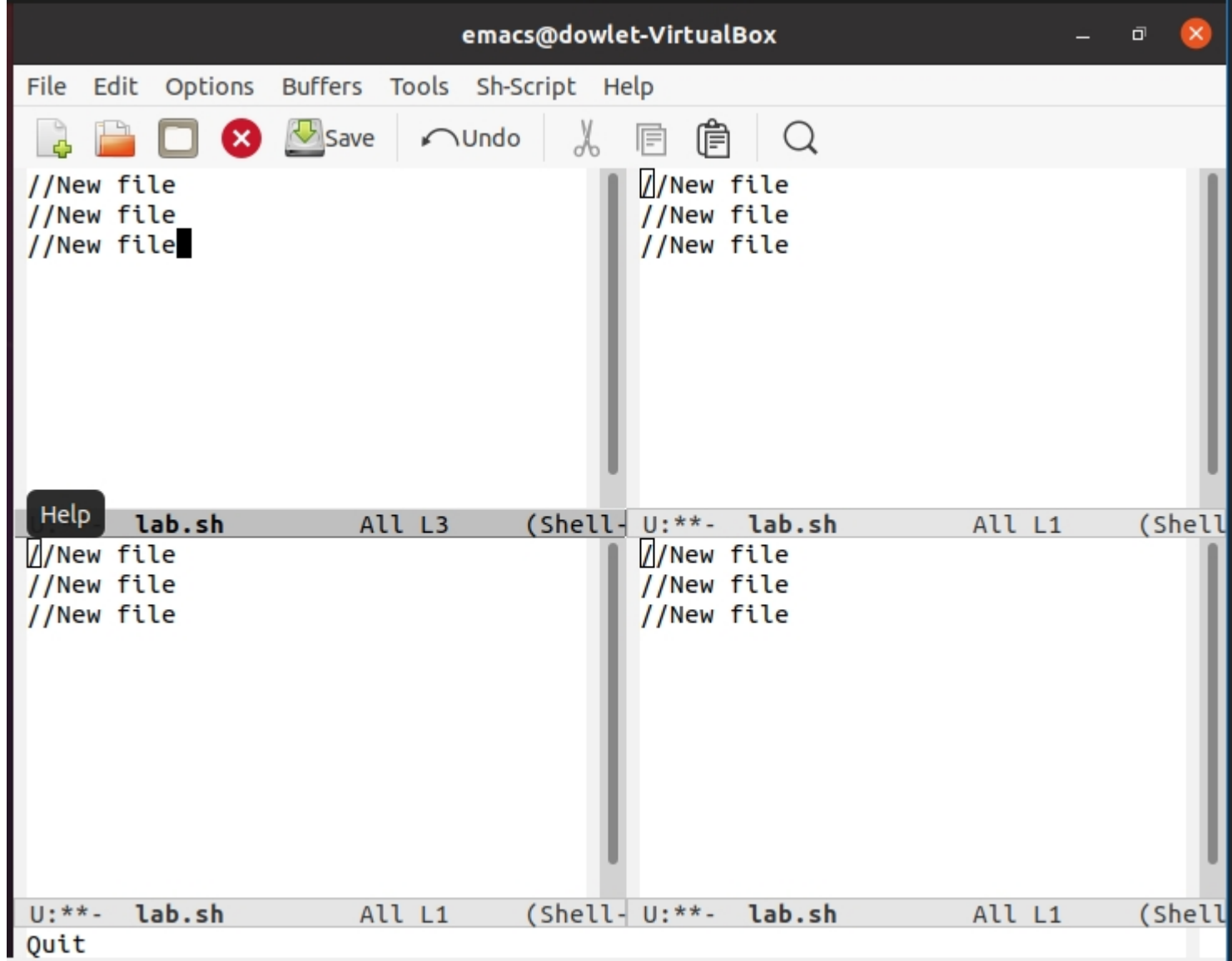
9. Режим поиска • Переключился в режим поиска (C-s) и нашел несколько слов, присутствующих в тексте.



Переключился между результатами поиска, нажимая C-s.



Вышел из режима поиска, нажав C-g.



- Перешел в режим поиска и замены (M-%), ввел текст, который следует найти и заменить, нажал Enter , затем ввел текст для замены. После подсвечивания результатов поиска, нажал ! для подтверждения замены.

File Edit Options Buffers Tools Minibuf Help



```
//New file
//New file
//New file
```

```
//New file
//New file
//New file
```

U:**- lab.sh All L3 (Shell-

```
//New file
//New file
//New file
```

U:**- lab.sh All L1 (Shel

```
//New file
//New file
//New file
```

U:**- lab.sh All L1 (Shell-

Query replace: file

U:**- lab.sh All L1 (Shel

File Edit Options Buffers Tools Minibuf Help



```
//New file  
//New file  
//New file
```

```
//New file  
//New file  
//New file
```

U:**- lab.sh All L3 (Shell-

```
//New file  
//New file  
//New file
```

U:**- lab.sh All L1 (Shell

```
//New file  
//New file  
//New file
```

U:**- lab.sh All L1 (Shell-

Query replace file with: catalog

U:**- lab.sh All L1 (Shell

File Edit Options Buffers Tools Sh-Script Help



```
//New catalog
//New catalog
//New catalog
```

```
//New catalog
//New catalog
//New catalog
```

```
U:**- lab.sh All L1 (Shell-
```

```
//New catalog
//New catalog
//New catalog
```

```
U:**- lab.sh All L1 (Shell-
```

```
//New catalog
//New catalog
//New catalog
```

```
U:**- lab.sh All L1 (Shell-
```

```
U:**- lab.sh All L1 (Shell-
```

- Попробовал другой режим поиска, нажав M-s o. Он отличается от обычного режима тем, что при поиске указывает номера строк в которых найдено введённое слово и выделяет их цветом. В обычном режиме выделение цветом появляется, только когда нужно подтвердить замену



```
//New catalog
//New catalog
//New catalog
```

```
3 matches for "catalog" in buffer: lab
1://New catalog
2://New catalog
3://New catalog
```

```
U:**- lab.sh All L1 (Shell-
//New catalog
//New catalog
//New catalog
```

```
U:%%- *Occur* All L1 (Occu
//New catalog
//New catalog
//New catalog
```

```
U:**- lab.sh All L1 (Shell-
Searched 1 buffer; 3 matches
```

```
U:**- lab.sh All L1 (Shell-
```

Вывод

Познакомился с операционной системой Linux, получил практические навыки работы с редактором Emacs.

Ответы на контрольные вопросы:

1. Emacs представляет собой мощный экраннй редактор текста, написанный на языке высокого уровня Lisp.
2. Развитие Emacs в сторону его многогранности послужило причиной того, что и без того интуитивно непонятная программа стала чрезвычайно сложной в применении. В частности, управление осуществляется при помощи различных клавиатурных комбинаций, запомнить которые будет непросто.
3. Буфер – что-то, состоящее из текста. Окно – область с одним из буферов.
4. В одном окне можно открыть больше 10 буферов.
5. После запуска emacs без каких-либо параметров в основном окне отображается буфер scratch, который используется для оценки выражений Emacs Lisp, а также для заметок, которые вы не хотите сохранять. Этот буфер не сохраняется автоматически.
6. Чтобы ввести следующую комбинацию C-c | я нажму клавиши: Control+c и Shift+, и для C-c C-|: Control+c и Control+Shift+.
7. Поделить текущее окно на две части можно двумя комбинациями клавиш: C-x 3 или C-x 2.
8. Настроить или расширить Emacs можно написав или изменив файл ~/.emacs.
9. Клавиша ↓ выполняет функцию перемещения курсора в открытом окне также, как и многие другие клавиши её можно переназначить.
10. Редактор emacs показался мне удобнее из-за возможности открытия нескольких окон с буферами и работать комбинациями клавиш в этот редакторе мне было проще.