

Текстовой редактор emacs

Отчёт по лабораторной работе №10

Мурашко В.В.

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Вывод	21
5	Библиография	22
6	Контрольные вопросы	23

Список таблиц

Список иллюстраций

3.1	Emacs	8
3.2	Создание файла	8
3.3	Текст	9
3.4	Сохранение файла	10
3.5	Вырезка целой строки	11
3.6	Вставка строки	11
3.7	Выделение области текста	12
3.8	Вставка области	12
3.9	Выделение и вырезка области	13
3.10	Отмена последнего действия	13
3.11	Перемещение курсора в начало строки	14
3.12	Перемещение курсора в конец строки	14
3.13	Перемещение курсора в начало буфера	15
3.14	Перемещение курсора в конец буфера	15
3.15	Список активных буферов	16
3.16	Другой буфер	16
3.17	Закрытие окна	16
3.18	Переключение между буферами	17
3.19	4 части фрейма	17
3.20	Новый буфер с текстом	18
3.21	Режим поиска	18
3.22	Переключение между результатами	19
3.23	Выход из режима поиска	19
3.24	Режим поиска и замены	19
3.25	Другой режим поиска	20
3.26	Другой режим поиска	20

1 Цель работы

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

2 Задание

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

Основные команды emacs. 1. Открыть emacs. 2. Создать файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f). 3. Наберите текст:

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

4. Сохранить файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
5. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш. 5.1.Вырезать одной командой целую строку (C-k). 5.2.Вставить эту строку в конец файла (C-y). 5.3.Выделить область текста (C-space). 5.4.Скопировать область в буфер обмена (M-w). 5.5.Вставить область в конец файла. 5.6.Вновь выделить эту область и на этот раз вырезать её (C-w). 5.7.Отмените последнее действие (C-/).

6. Научитесь использовать команды по перемещению курсора. 6.1.Переместите курсор в начало строки (C-a). 6.2.Переместите курсор в конец строки (C-e). 6.3.Переместите курсор в начало буфера (M-<). 6.4.Переместите курсор в конец буфера (M->).
7. Управление буферами. 7.1.Вывести список активных буферов на экран (C-x C-b). 7.2.Переместитесь во вновь открытое окно (C-x) со списком открытых буферов и переключитесь на другой буфер. 7.3.Закройте это окно (C-x 0). 7.4.Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран (C-x b).
8. Управление окнами. 8.1.Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2). 8.2.В каждом из четырёх созданных окон откройте новый буфер (файл) и введите несколько строк текста.
9. Режим поиска. 9.1.Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте. 9.2.Переключайтесь между результатами поиска, нажимая C-s. 9.3.Выйдите из режима поиска, нажав C-g. 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. 9.5.Испробуйте другой режим поиска, нажав M-s o.

3 Выполнение лабораторной работы

Основные команды emacs

1. Я открыла emacs.

```
vvmurashko1@dk8n61 ~ $ emacs
```

Рис. 3.1: Emacs

2. Я создала файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).

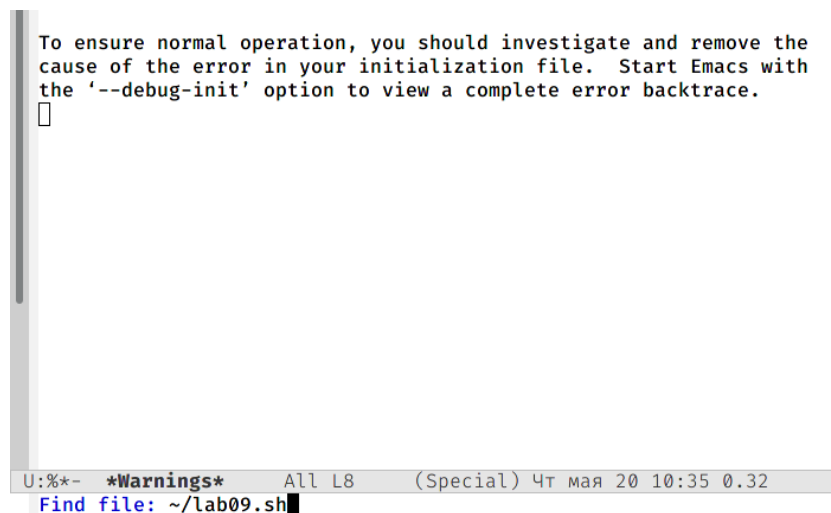


Рис. 3.2: Создание файла

3. Я набрала текст:


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



Рис. 3.3: Текст

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

The screenshot shows an Emacs editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a buffer titled 'lab09.sh'. The script content is as follows:

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

Below the script, a warning message is displayed:

```
U:--- lab09.sh All L8 (Shell-script[sh]) Чт мая 20
Warning (initialization): An error occurred while loading '~.
error: Package 'fira-code-mode-' is unavailable

To ensure normal operation, you should investigate and remov
cause of the error in your initialization file. Start Emacs
the '--debug-init' option to view a complete error backtrace

```

The bottom status bar shows the buffer 'U:%*- *Warnings*' and the file path: 'Wrote /afs/.dk.sci.pfu.edu.ru/home/v/v/vvmurashko1/lab09.sh'.

Рис. 3.4: Сохранение файла

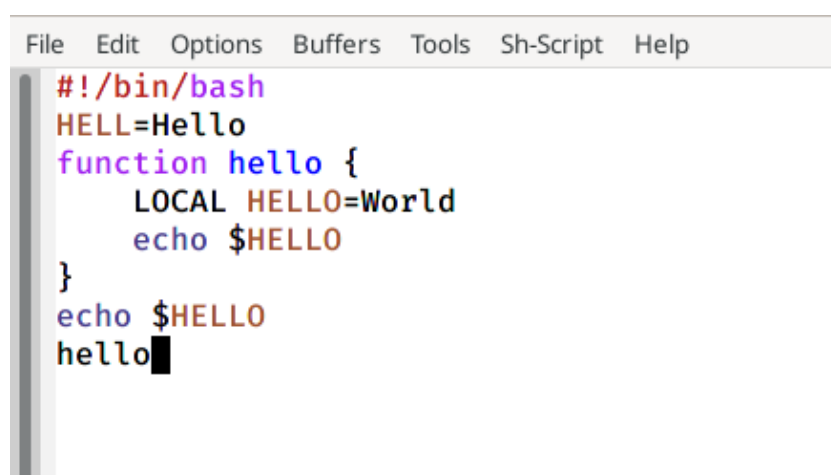
5. Я проделала с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш. 5.1. Я вырезала одной командой целую строку (C-k). 5.2. Я вставила эту строку в конец файла (C-y). 5.3. Я выделила область текста (C-space). 5.4. Я скопировала

область в буфер обмена (M-w). 5.5. Я вставила область в конец файла. 5.6. Я вновь выделила эту область и на этот раз вырезала её (C-w). 5.7. Я отменила последнее действие (C-/).



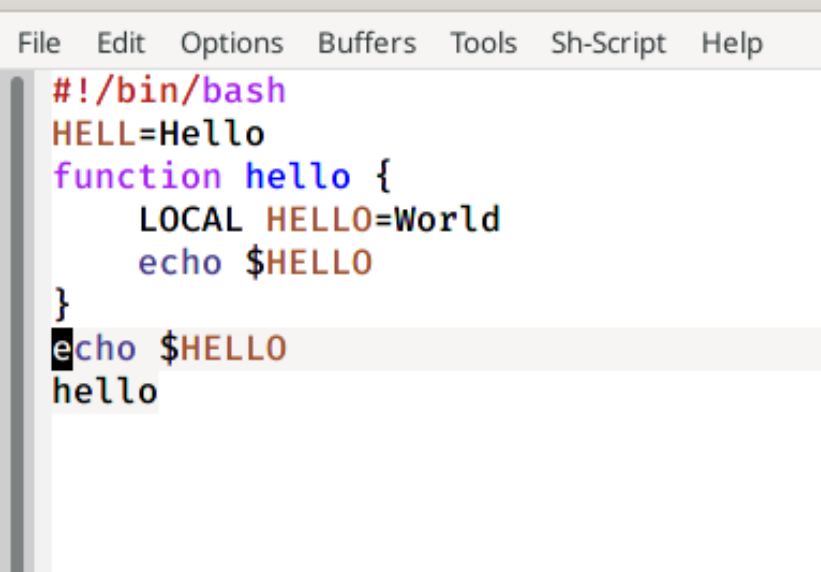
```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
█
```

Рис. 3.5: Вырезка целой строки



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello█
```

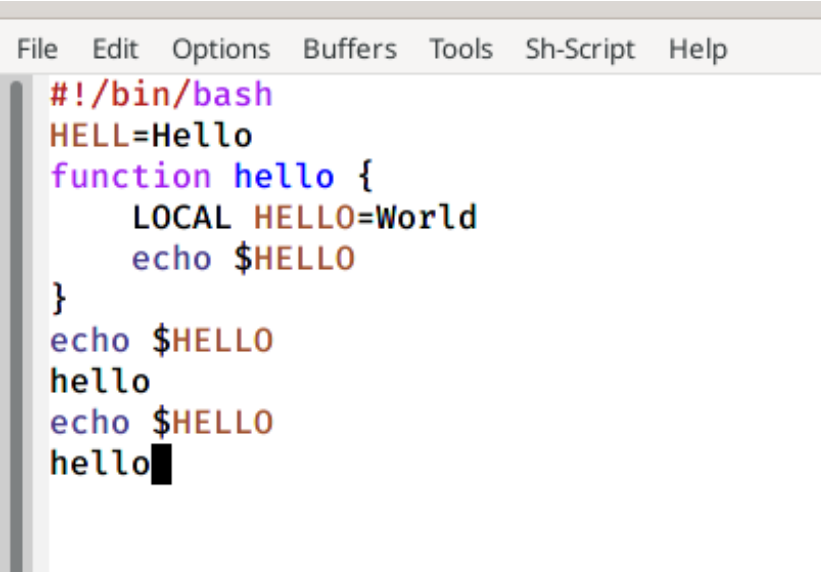
Рис. 3.6: Вставка строки

A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The editor contains a shell script with the following content:

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

 A light gray rectangular selection highlight is placed over the line `echo $HELLO` in the script.

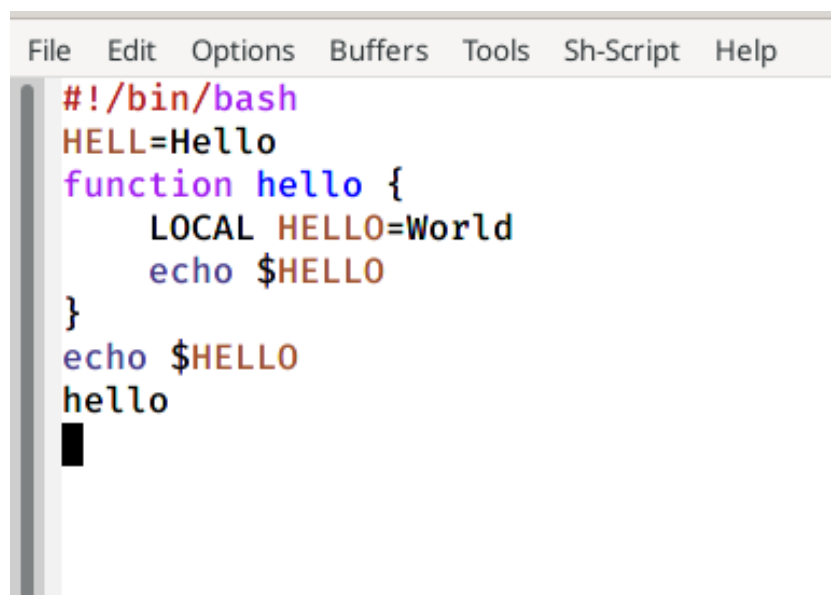
Рис. 3.7: Выделение области текста

A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The editor contains a shell script with the following content:

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

 The script is identical to the one in Figure 3.7, but it includes an additional `echo $HELLO` line at the bottom, resulting in two `hello` output lines.

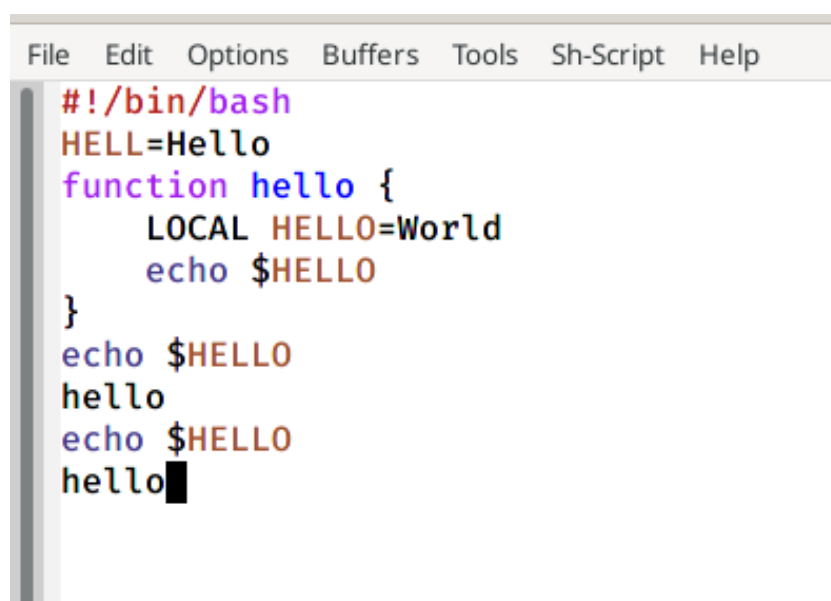
Рис. 3.8: Вставка области



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

The screenshot shows a Sh-Script editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a text area containing a shell script. A vertical selection bar is visible on the left side of the text area, indicating that a portion of the script has been selected.

Рис. 3.9: Выделение и вырезка области



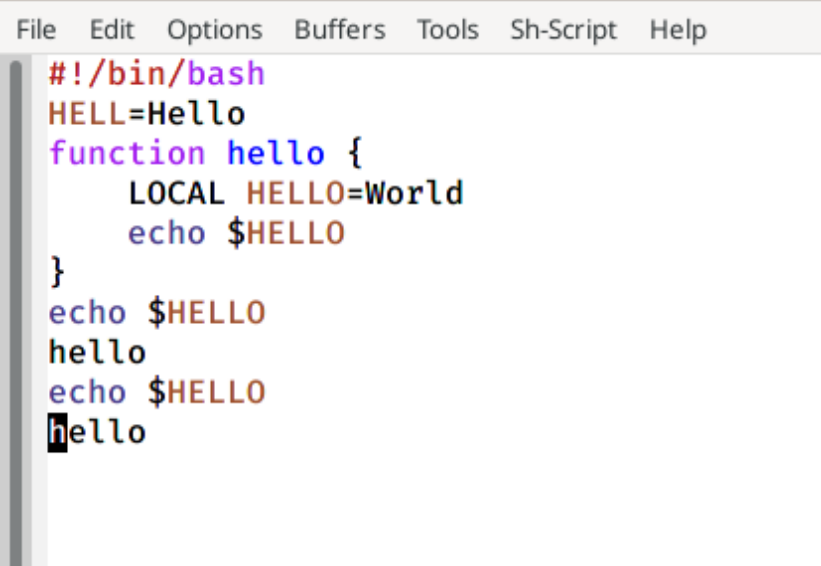
```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
hello
```

The screenshot shows the same Sh-Script editor window as in Figure 3.9, but with an additional 'echo \$HELLO' and 'hello' line added at the bottom of the script. The selection bar is still present on the left side of the text area.

Рис. 3.10: Отмена последнего действия

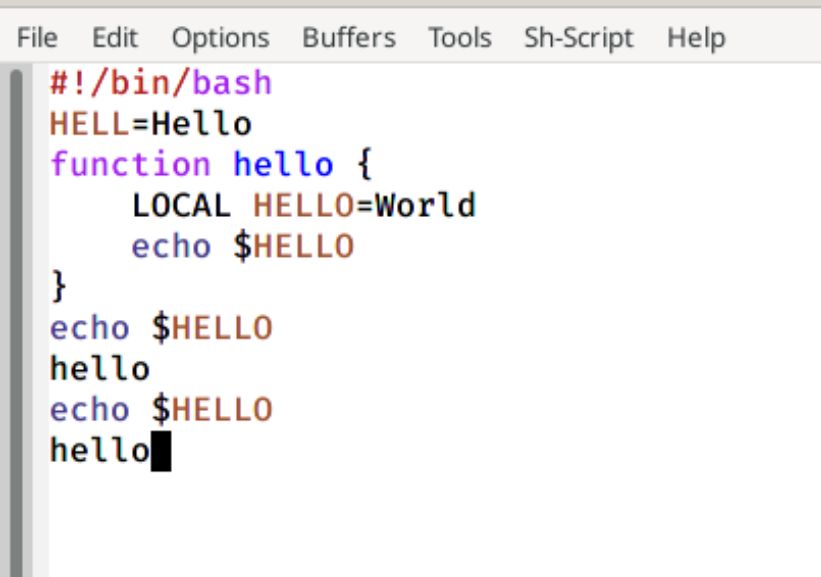
6. Я научилась использовать команды по перемещению курсора. 6.1. Я переместила курсор в начало строки (C-a). 6.2. Я переместила курсор в конец строки (C-e). 6.3. Я переместила курсор в начало буфера (M-<). 6.4. Я переместила

курсор в конец буфера (M->).



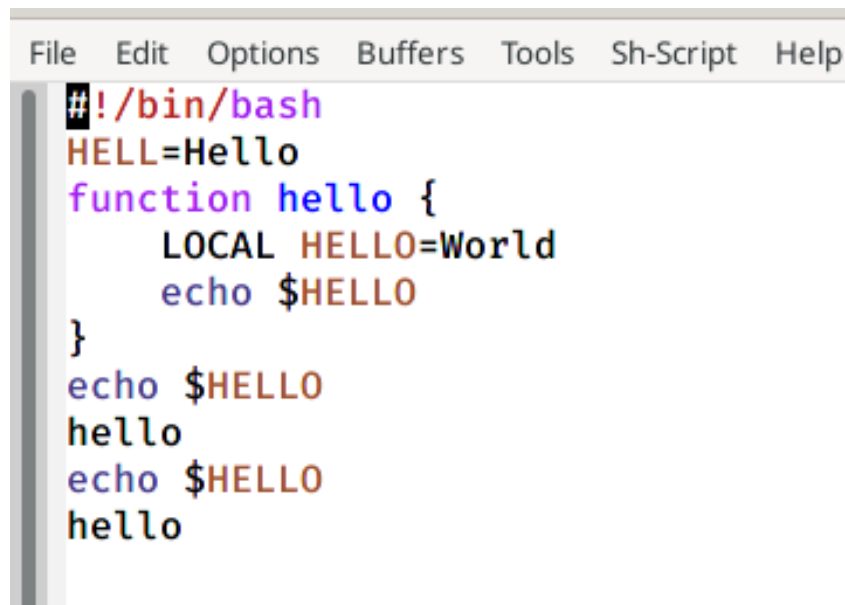
```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
hello
```

Рис. 3.11: Перемещение курсора в начало строки



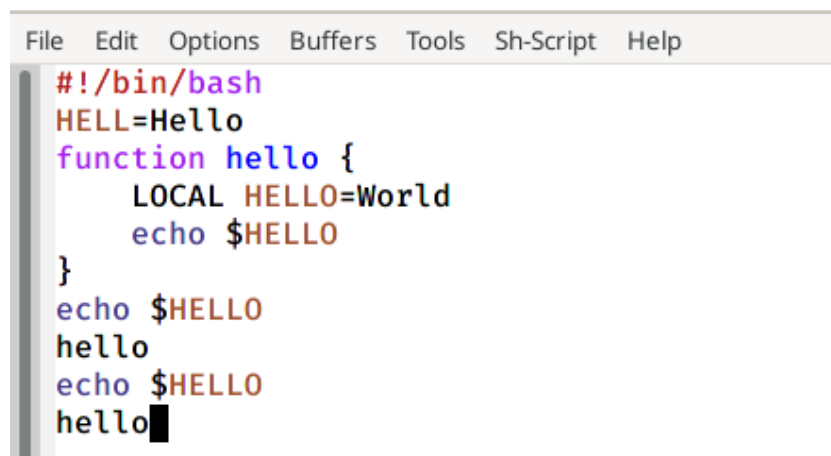
```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
echo $HELLO
hello
```

Рис. 3.12: Перемещение курсора в конец строки



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

Рис. 3.13: Перемещение курсора в начало буфера



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

Рис. 3.14: Перемещение курсора в конец буфера

7. Управление буферами. 7.1. Я вывела список активных буферов на экран (C-x C-b). 7.2. Я переместилась во вновь открытое окно (C-x) со списком открытых буферов и переключилась на другой буфер. 7.3. Я закрыла это окно (C-x 0). 7.4. Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран (C-x b).

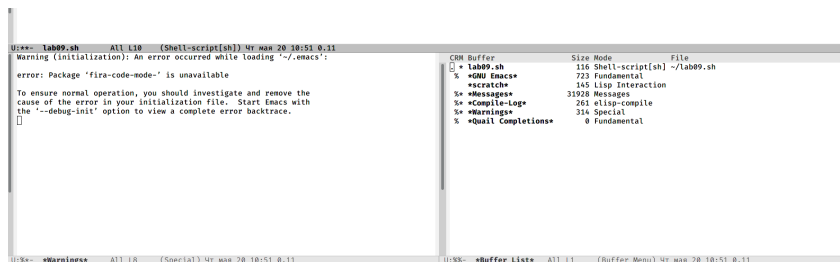


Рис. 3.15: Список активных буферов

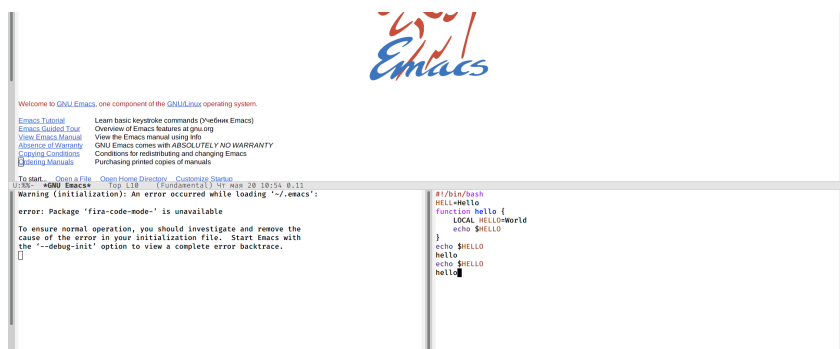


Рис. 3.16: Другой буфер

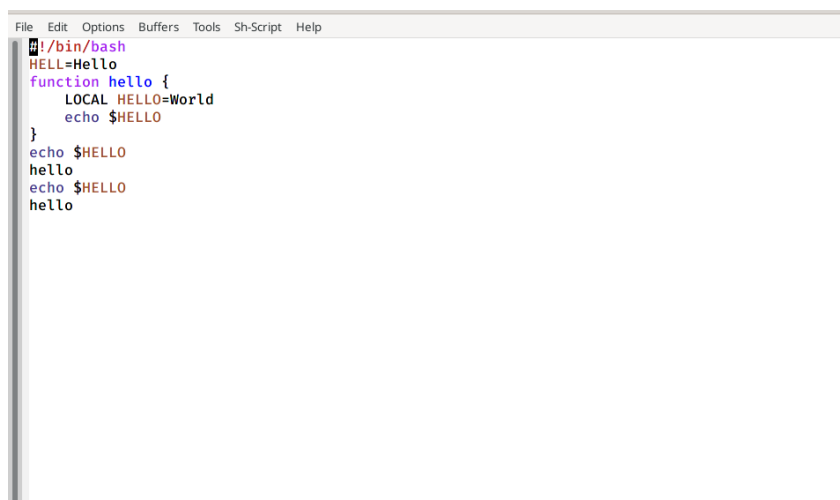


Рис. 3.17: Закрытие окна

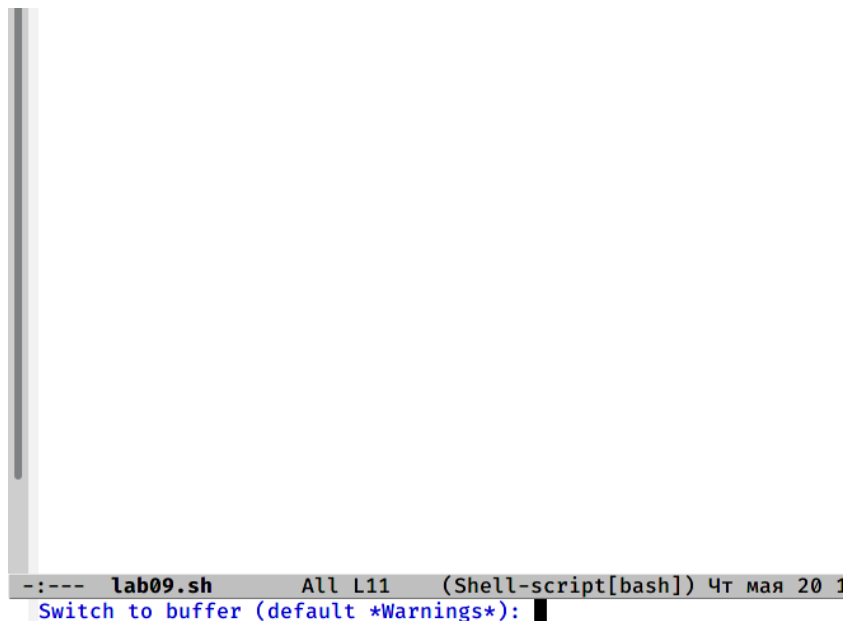


Рис. 3.18: Переключение между буферами

8. Управление окнами. 8.1. Я поделила фрейм на 4 части: разделила фрейм на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2). 8.2. В каждом из четырёх созданных окон я открыла новый буфер (файл) и ввела несколько строк текста.

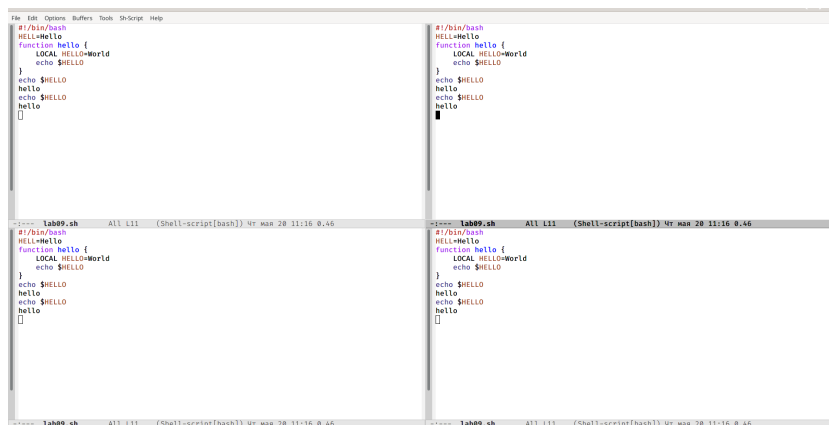


Рис. 3.19: 4 части фрейма

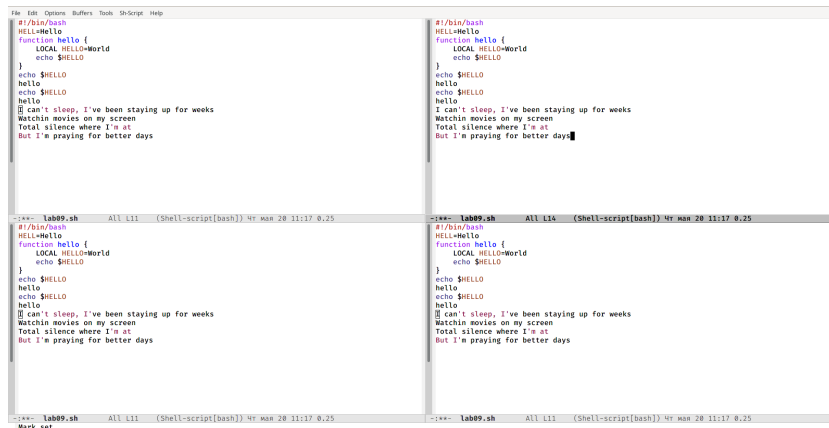


Рис. 3.20: Новый буфер с текстом

9. Режим поиска. 9.1. Я переключилась в режим поиска (C-s) и нашла несколько слов, присутствующих в тексте. 9.2. Я переключалась между результатами поиска, нажимая C-s. 9.3. Я вышла из режима поиска, нажав C-g. 9.4. Я перешла в режим поиска и замены (M-%), ввела текст, который следует найти и заменить, нажала Enter, затем ввела текст для замены. После того как подсветились результаты поиска, я нажала ! для подтверждения замены. 9.5. Я испробовала другой режим поиска, нажав M-so.

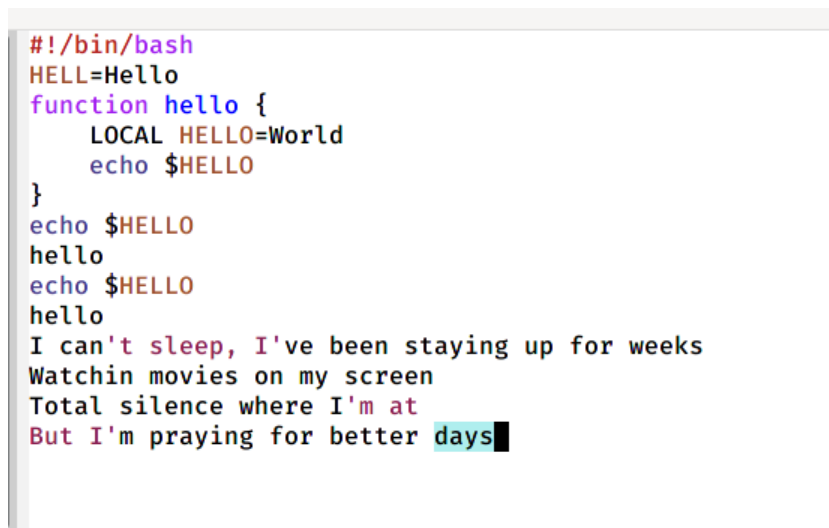


Рис. 3.21: Режим поиска



Рис. 3.22: Переключение между результатами

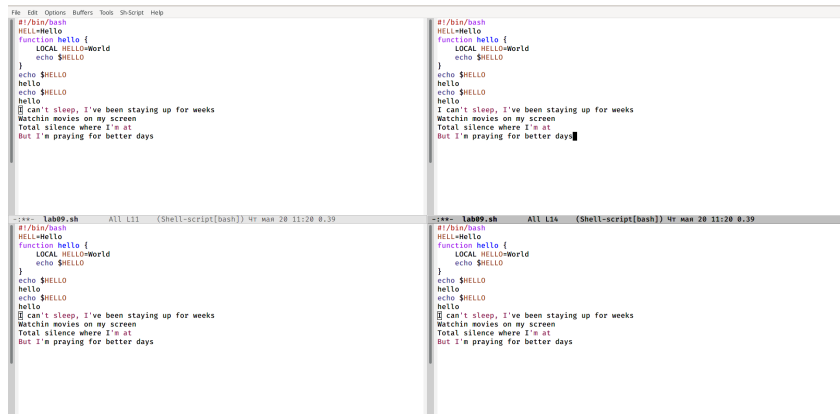


Рис. 3.23: Выход из режима поиска



Рис. 3.24: Режим поиска и замены



Рис. 3.25: Другой режим поиска

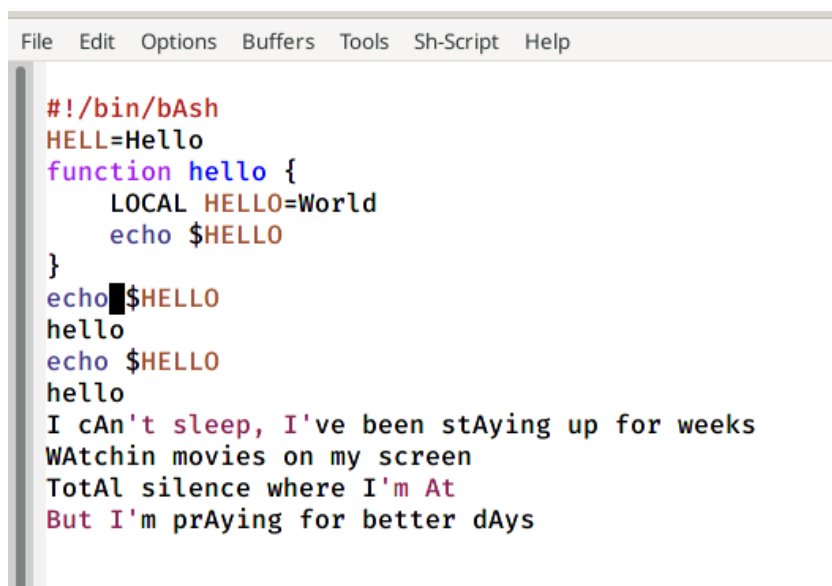


Рис. 3.26: Другой режим поиска

4 Вывод

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

5 Библиография

https://esystem.rudn.ru/pluginfile.php/1142514/mod_resource/content/3/007-lab_emacs.pdf

6 Контрольные вопросы

1. Emacs представляет собой мощный экраный 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-с | я нажму клавиши: Control+c и Shift+, и для C-с 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 f 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.