

Операционные системы

Лабораторная работа №10

Крупенникова Виктория Александровна

Содержание

| | | |
|---|--------------------------------|----|
| 1 | Цель работы | 3 |
| 2 | Задание | 4 |
| 3 | Выполнение лабораторной работы | 5 |
| 4 | Контрольные вопросы: | 26 |
| 5 | Выводы | 28 |

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 етас. 3.Выполнить упражнения. 4.Ответить на контрольные вопросы.

3 Выполнение лабораторной работы

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

```
vakrupennikova@dk3n52 ~ $ emacs &  
[1] 19433  
vakrupennikova@dk3n52 ~ $
```

Figure 3.1: Открытие редактора

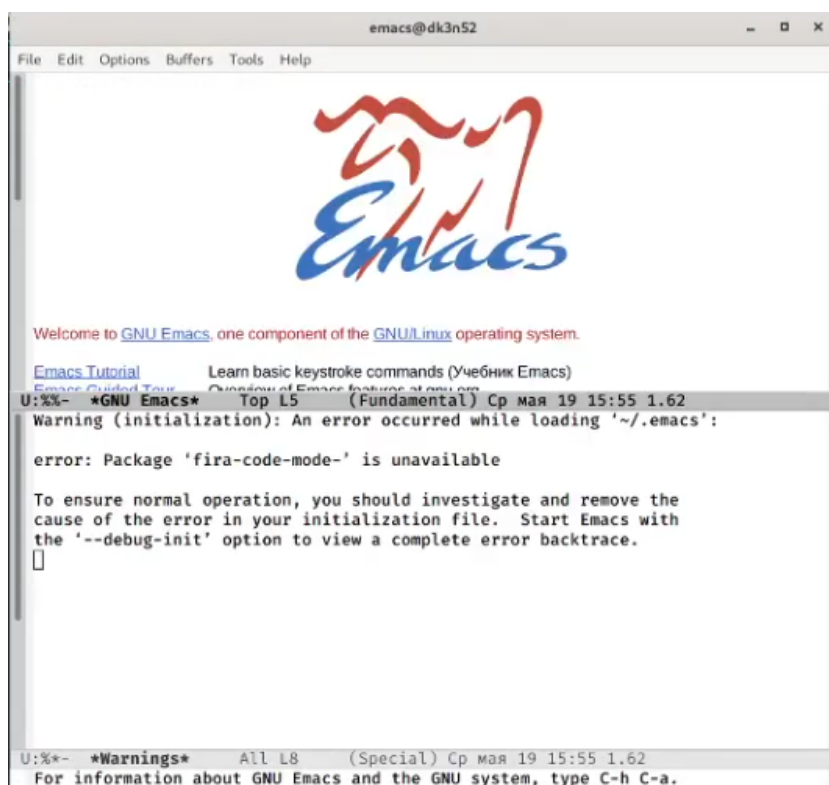


Figure 3.2: Редактор

2) Создадим файл lab10.sh с помощью комбинации «Ctrl-x»«Ctrl-f».

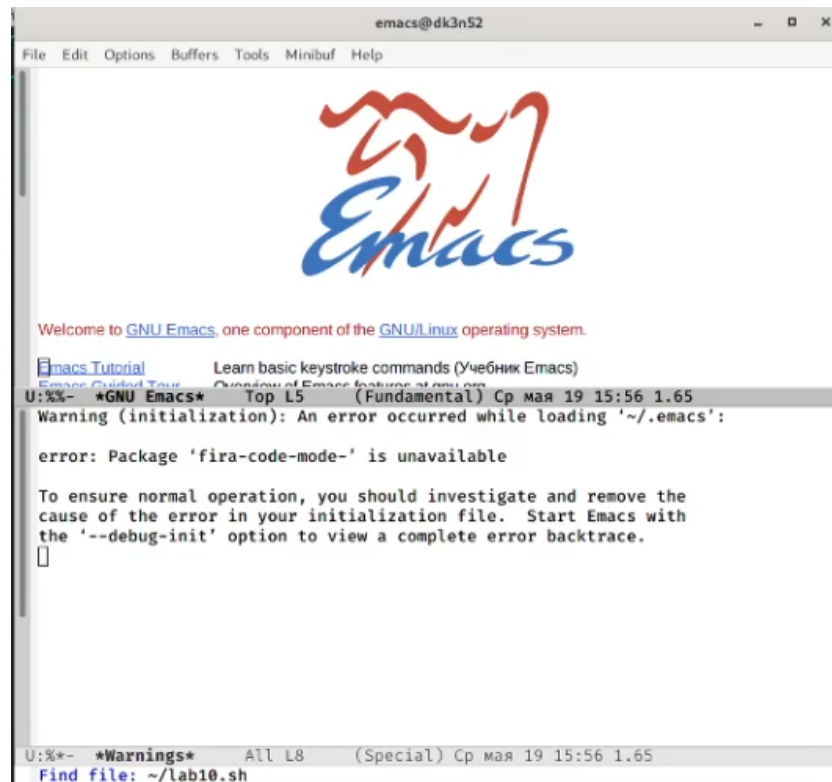
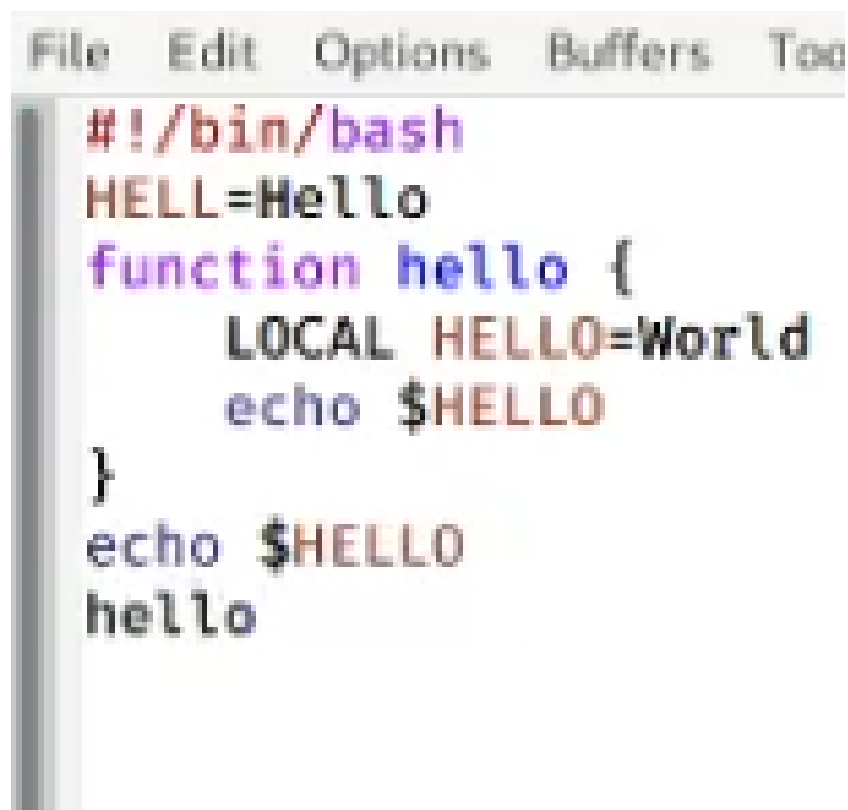


Figure 3.3: Создание файла

3) В открывшемся буфере наберем необходимый текст

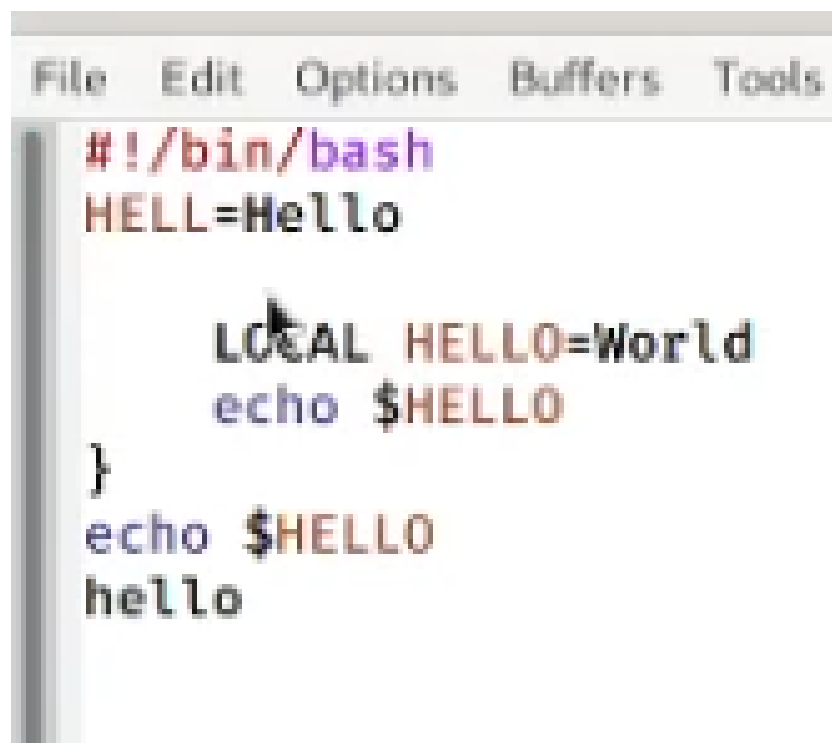
A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', and 'Too'. The terminal displays a shell script with the following content:

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

Figure 3.4: Набор текста

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

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

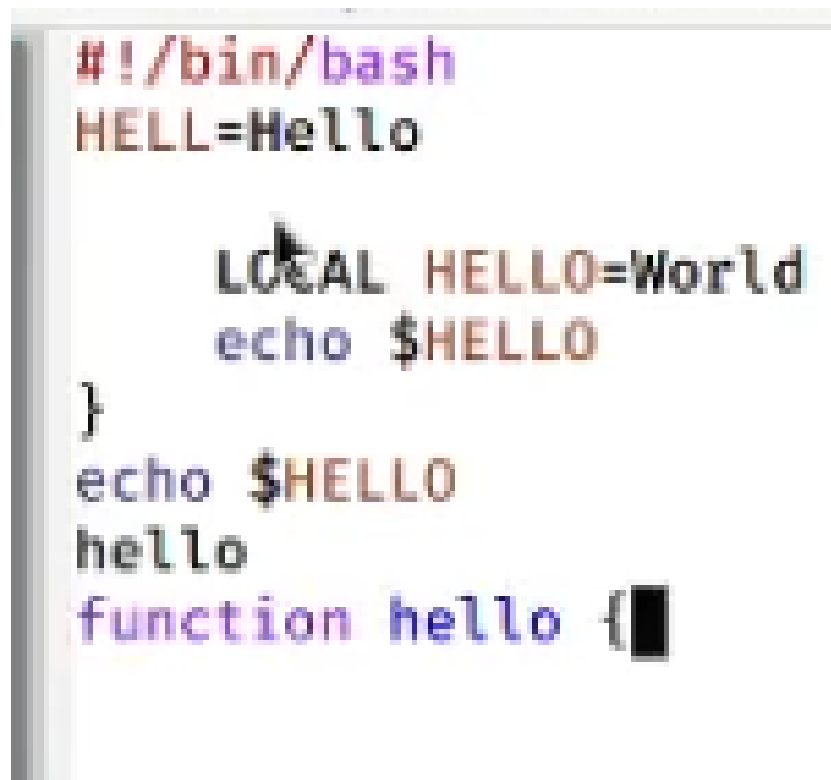


```
File Edit Options Buffers Tools
#!/bin/bash
HELL=Hello

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

Figure 3.5: Вырезка строки

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



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

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

Figure 3.6: Вставка строки

5.3) Выделим область текста («Ctrl-space»)

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

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

Figure 3.7: Выделение области текста

5.4) Скопируем область в буфер обмена («Alt-w»). 5.5) Вставим область в конец файла («Ctrl-y»)

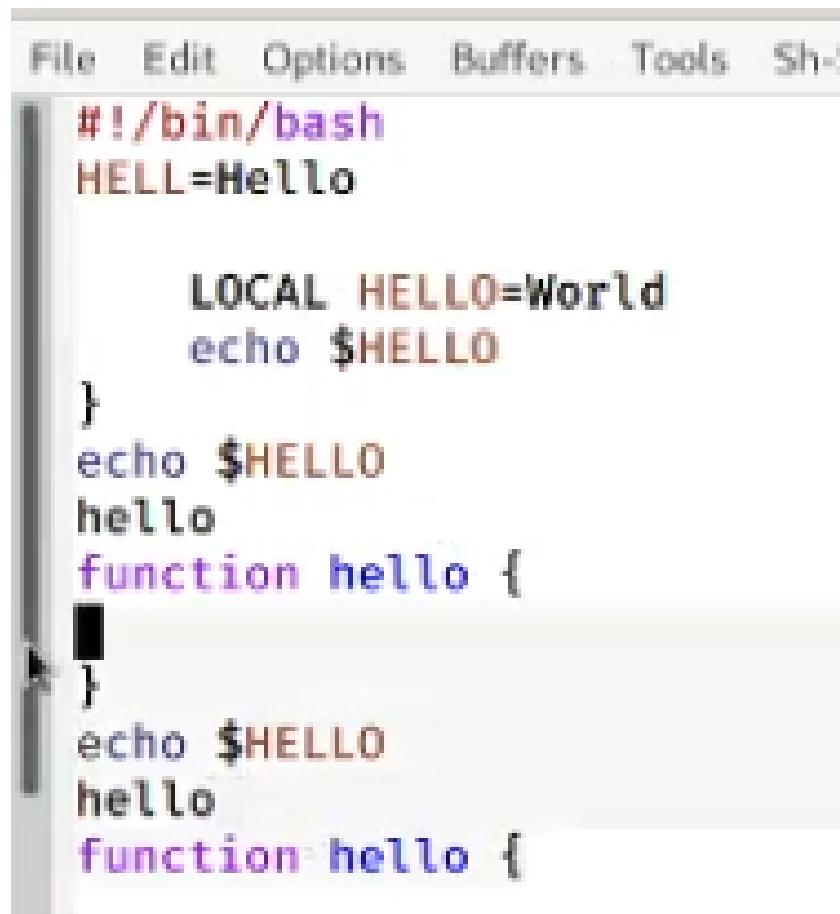
```
#!/bin/bash
HELL=Hello

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

}
echo $HELLO
hello
function hello {
```

Figure 3.8: Вставка области текста

5.6) Вновь выделим эту область(«Ctrl-space»)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Sh-'. The terminal displays a bash script. The first line is '#!/bin/bash' in red. The second line is 'HELL=Hello' in red. The third line is 'LOCAL HELLO=World' in red. The fourth line is 'echo \$HELLO' in red. The fifth line is '}' in black. The sixth line is 'echo \$HELLO' in red. The seventh line is 'hello' in black. The eighth line is 'function hello {' in red. The ninth line is a black cursor. The tenth line is '}' in black. The eleventh line is 'echo \$HELLO' in red. The twelfth line is 'hello' in black. The thirteenth line is 'function hello {' in red. A light gray rectangular highlight covers the lines from 'function hello {' to the end of the script. A mouse cursor is visible on the left side of the terminal window, pointing at the highlight.

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

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

Figure 3.9: Выделение области

и на этот раз вырежем её («Ctrl-w»)

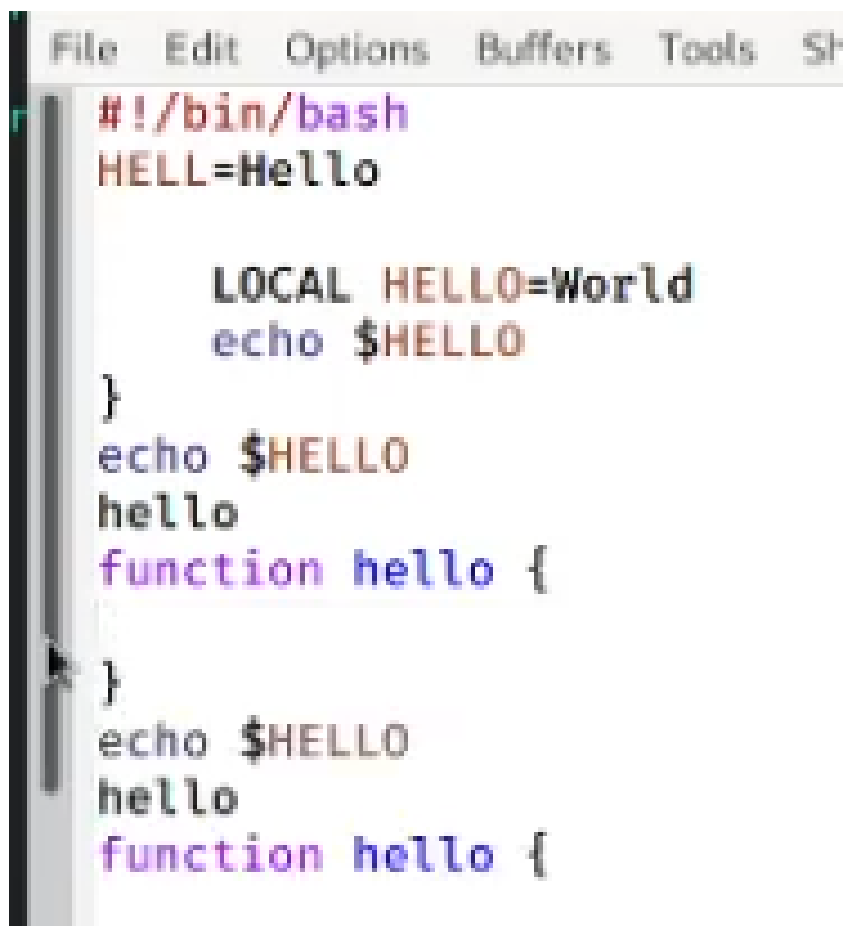
A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'St'. The terminal displays a Bash script snippet with the following lines:

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

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

Figure 3.10: Вырезка области

5.7) Отменим последнее действие («Ctrl-/-»)

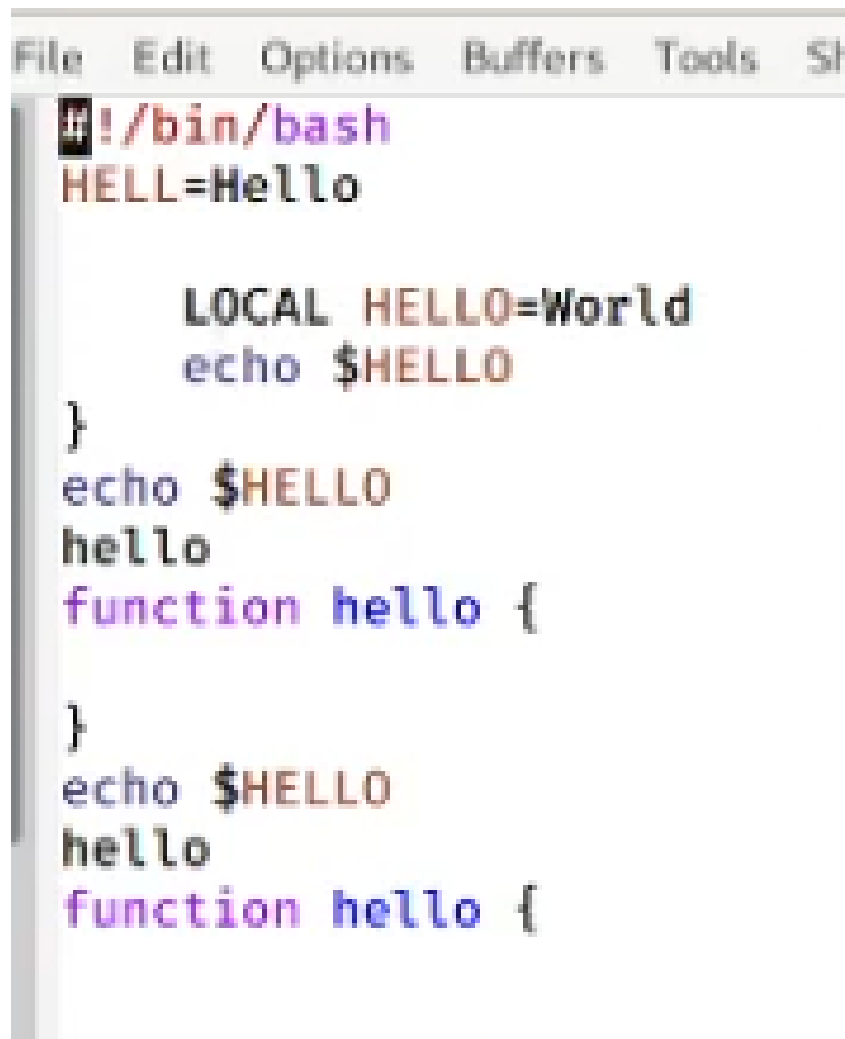


```
File Edit Options Buffers Tools Sh
#!/bin/bash
HELL=Hello

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

Figure 3.11: Отмена последнего действия

- 6.1) Переместим курсор в начало строки («Ctrl-a»)
- 6.2) Переместим курсор в конец строки («Ctrl-e»)
- 6.3) Переместим курсор в начало буфера («Alt-<»)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Shell'. The terminal displays a bash script with the following content:

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


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

}
echo $HELLO
hello
function hello {
```

 A black cursor is positioned at the beginning of the first line, before the shebang character.

Figure 3.12: Курсор в начале буфера

6.4) Переместим курсор в конец буфера («Alt->»)



```
File Edit Options Buffers Tools $
#!/bin/bash
HELL=Hello

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

}
echo $HELLO
hello
function hello {
```

Figure 3.13: Курсор в конце буфера

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

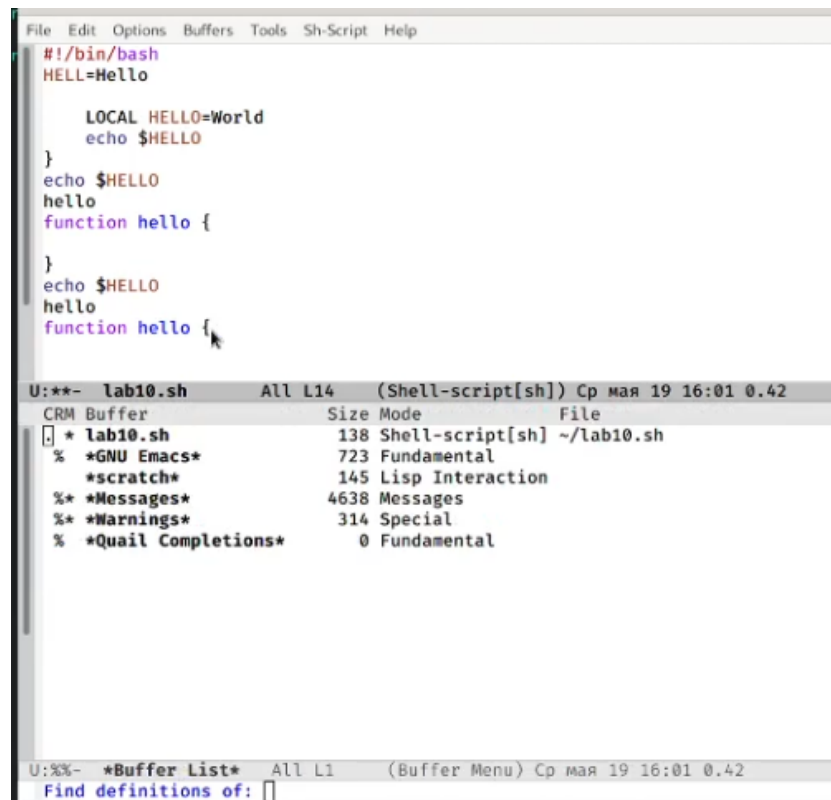


Figure 3.14: Список активных буферов

7.2) Переместимся во вновь открытое окно («Ctrl-хо») со списком открытых буферов

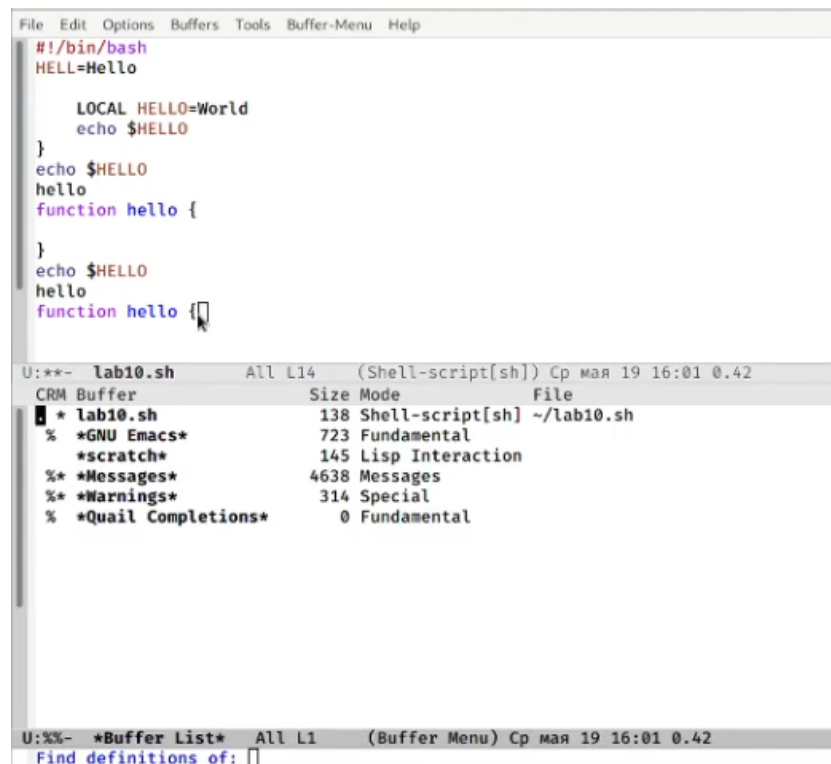


Figure 3.15: Перемещение в окно со списком буферов

и переключимся на другой буфер(для этого необходимо нажать на «enter» после выбора необходимого буфера)

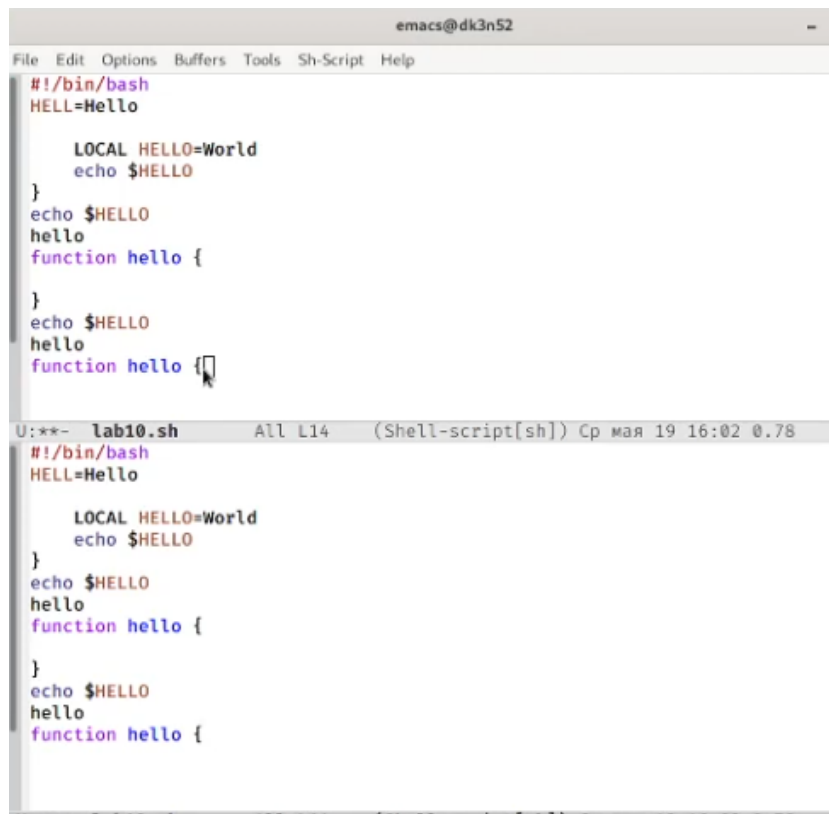
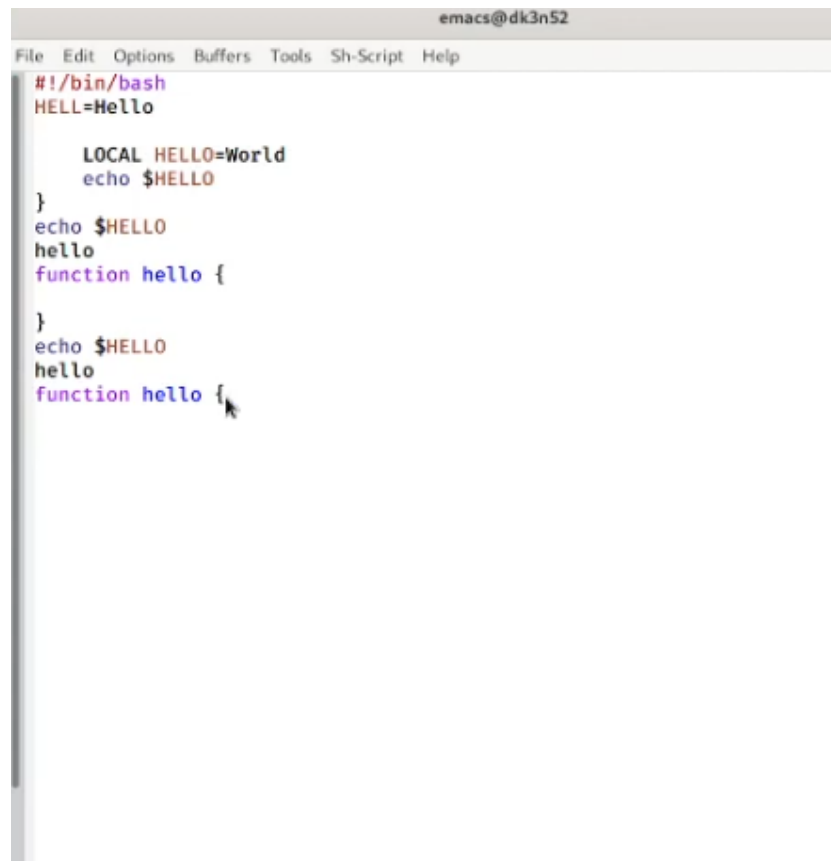


Figure 3.16: Переключение на другой буфер

7.3) Закроем это окно («Ctrl-x0»)



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

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

Figure 3.17: Заккрытие окна

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

The screenshot shows the Emacs editor window titled 'emacs@dk3n52'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The main buffer contains a shell script with the following content:

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

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

Below the script, a status line indicates 'U:*~ tab10.sh All L14 (Shell-script[sh]) Cp мая 19 16:02 0.78'. A warning message is displayed:

```
Warning (initialization): An error occurred while loading '~/.emacs':
error: Package 'fira-code-mode-' is unavailable

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

At the bottom, another status line shows 'U:%~ *Warnings* All L8 (Special) Cp мая 19 16:02 0.78' and the message 'Command attempted to use minibuffer while in minibuffer'.

Figure 3.18: Переключение 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»)

The screenshot shows the Emacs editor window titled 'emacs@dk3n52' with a split window layout. The left pane contains the same shell script as in Figure 3.18:

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

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

The right pane contains an identical copy of the same shell script:

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

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

Figure 3.19: Деление фрейма

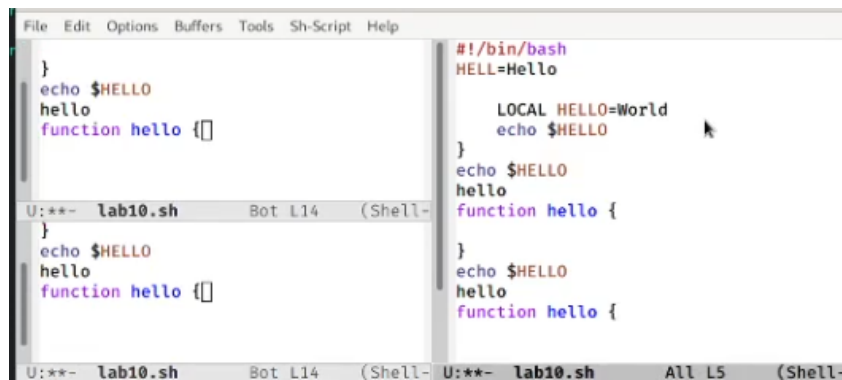


Figure 3.20: Деление фрейма

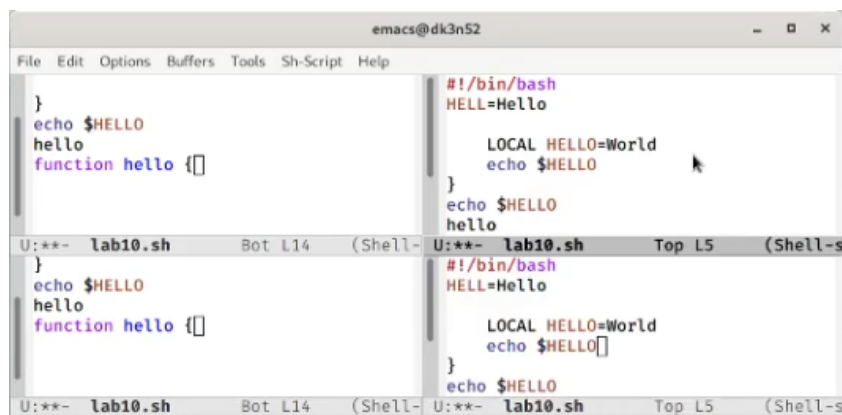


Figure 3.21: Деление фрейма

8.2) В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста. Для этого предварительно создадим эти файлы с помощью команд «touchexample1.txt», «touchexample2.txt», «touchexample3.txt», «touchexample4.txt»

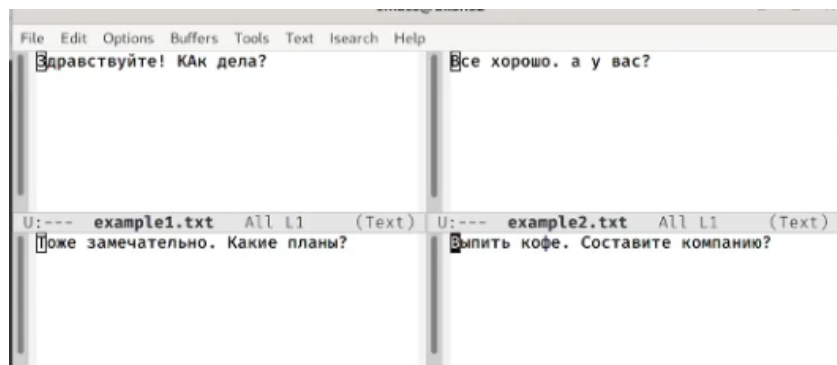
```
vakrpenikova@dk3n52 ~$ touch example1.txt
vakrpenikova@dk3n52 ~$ touch example2.txt
vakrpenikova@dk3n52 ~$ touch example3.txt
vakrpenikova@dk3n52 ~$ touch example4.txt
vakrpenikova@dk3n52 ~$
```

Figure 3.22: Создание файлов



Find file: ~/example4.txt

Figure 3.23: Открытие нового буфера



File Edit Options Buffers Tools Text Isearch Help

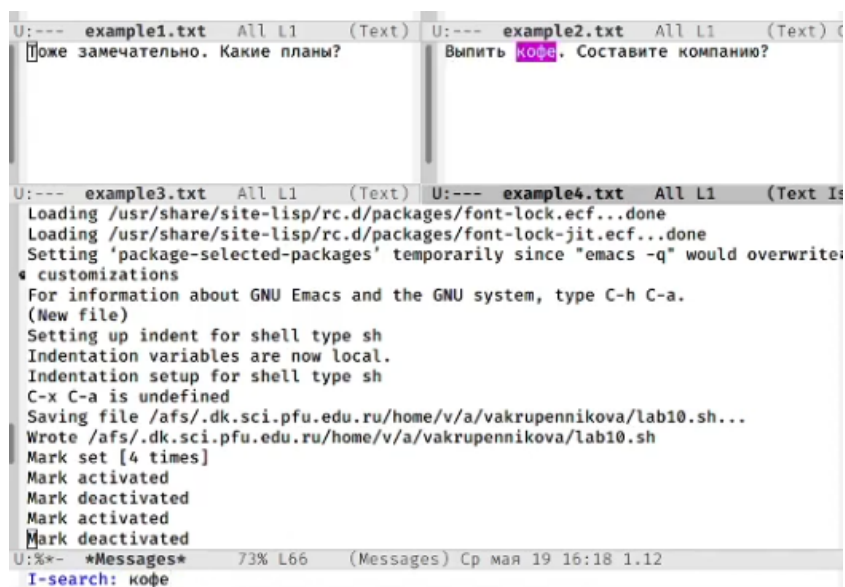
Здравствуйте! КАк дела? Все хорошо. а у вас?

U:--- example1.txt All L1 (Text) U:--- example2.txt All L1 (Text)

Поже замечательно. Какие планы? Выпить кофе. Составите компанию?

Figure 3.24: Вывод на экран

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



U:--- example1.txt All L1 (Text) U:--- example2.txt All L1 (Text)

Поже замечательно. Какие планы? Выпить кофе. Составите компанию?

U:--- example3.txt All L1 (Text) U:--- example4.txt All L1 (Text)

Loading /usr/share/site-lisp/rc.d/packages/font-lock.ecf...done
Loading /usr/share/site-lisp/rc.d/packages/font-lock-jit.ecf...done
Setting 'package-selected-packages' temporarily since "emacs -q" would overwrite
customizations
For information about GNU Emacs and the GNU system, type C-h C-a.
(New file)
Setting up indent for shell type sh
Indentation variables are now local.
Indentation setup for shell type sh
C-x C-a is undefined
Saving file /afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/lab10.sh...
Wrote /afs/.dk.sci.pfu.edu.ru/home/v/a/vakrupennikova/lab10.sh
Mark set [4 times]
Mark activated
Mark deactivated
Mark activated
Mark deactivated

U:~*- *Messages* 73% L66 (Messages) Ср мая 19 16:18 1.12

I-search: кофе

Figure 3.25: Поиск слова

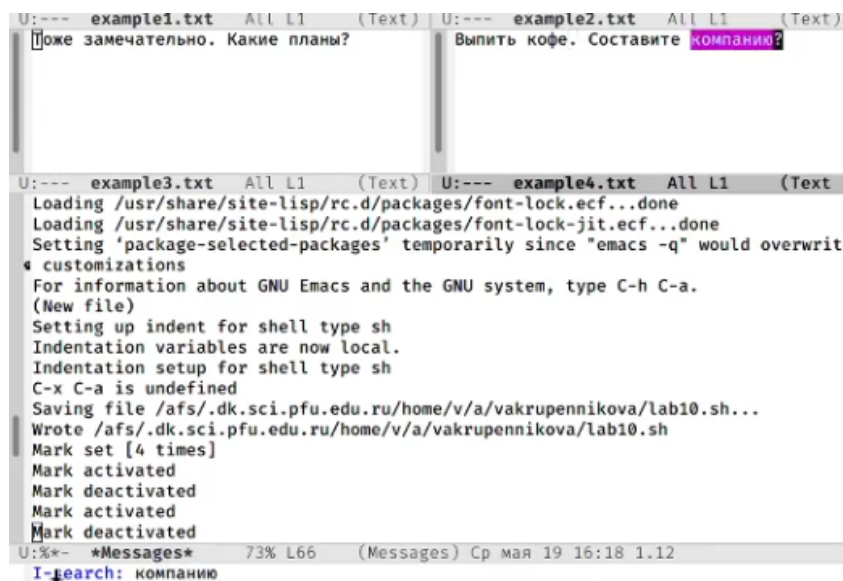


Figure 3.26: Поиск слова

- 9.2) Переключимся между результатами поиска, нажимая «Ctrl-s»
- 9.3) Выйдем из режима поиска, нажав «Ctrl-g»
- 9.4) Перейдем в режим поиска и замены («Alt-%»), 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. Важно, чтобы курсор находился в начале текста.

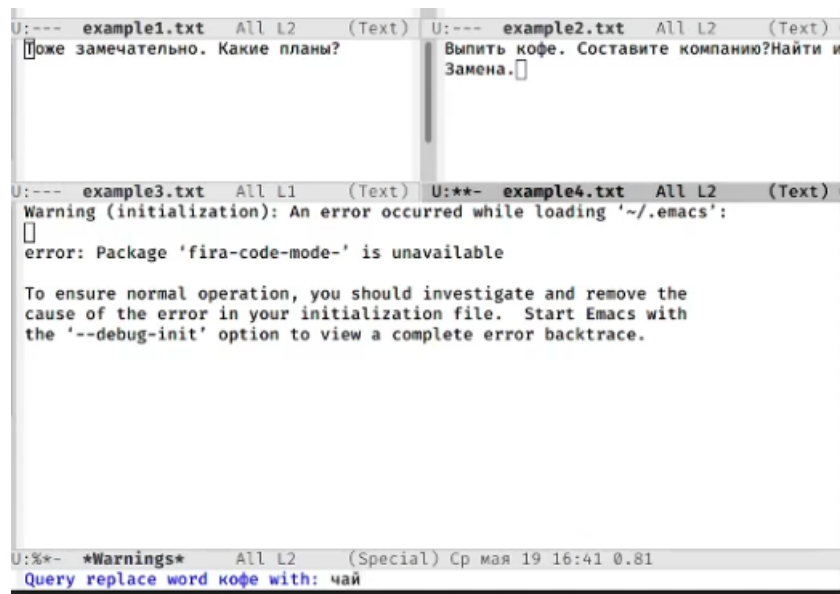


Figure 3.27: Замена

9.5) Пробуем другой режим поиска, нажав «Alt-so» Данный вид поиска отличается от обычного тем, что тут считывается строка поиска, которая трактуется как регулярное выражение, и не осуществляется поиск точного совпадения в тексте буфера. Регулярное выражение – это образец, который обозначает набор строк, возможно, и неограниченный набор.

4 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 того 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 открыться 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) «GNU Emacs» (справочный буфер о редакторе)

6) C-c | сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и нажимаю «|» C-c C-| 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 хранятся в файле .emacs. 9) По умолчанию клавиша «» удаляет символ перед курсором, но в 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 неудобными.

5 Выводы

В ходе выполнения данной лабораторной работы я познакомилась с операционной системой Linux и получила практические навыки работы с редактором Emacs