

Операционные системы

Лабораторная работа №10

Безрук Мария Андреевна

Содержание

1 Цель работы	3
2 Задание	4
3 Выполнение лабораторной работы	5
4 Контрольные вопросы:	22
5 Выводы	24

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.

3 Выполнение лабораторной работы

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

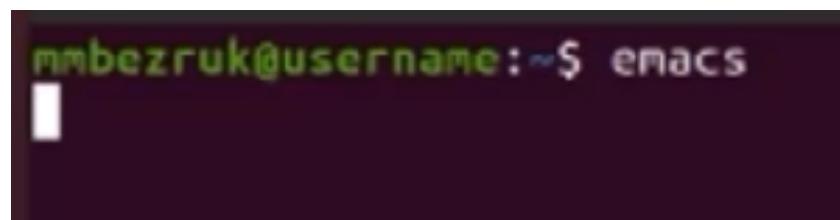


Figure 3.1: Открытие редактора

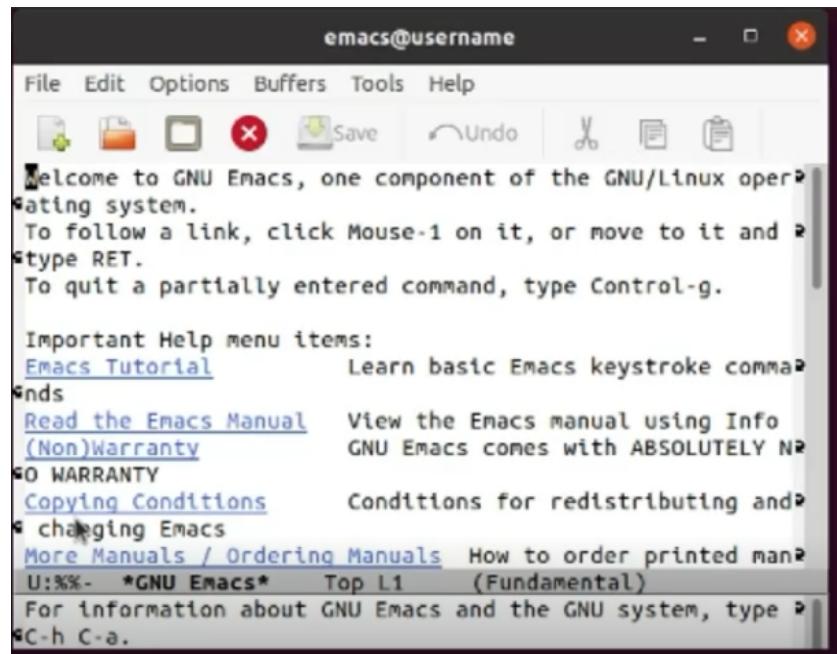


Figure 3.2: Редактор

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

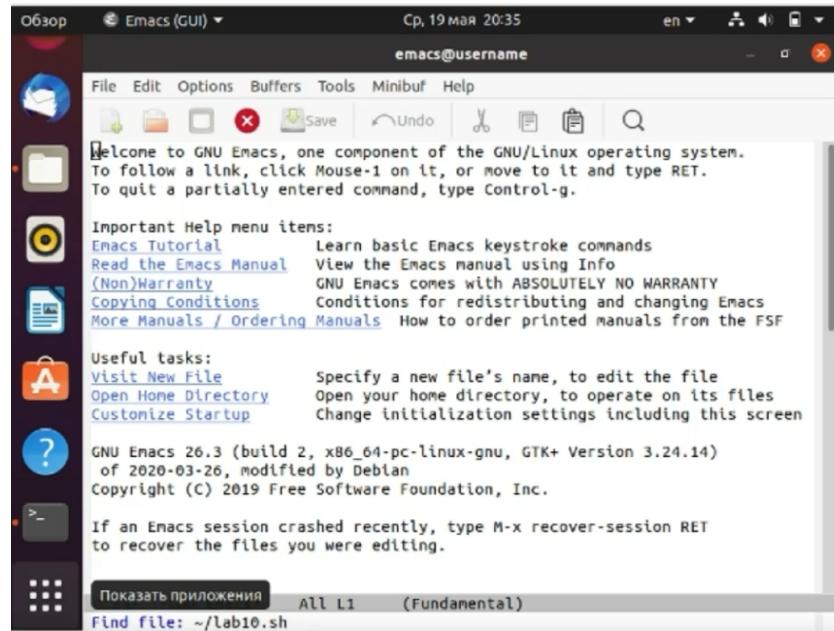


Figure 3.3: Создание файла

3) В открывшемся буфере наберем необходимый текст

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

The image shows a screenshot of an Emacs buffer containing a shell script. The script defines a variable \$HELL, a function hello that sets a local variable \$LOCAL to "World" and prints \$HELLO, and then prints \$HELLO directly. The cursor is positioned at the end of the word "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»)

```
#!/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 $HELLO
}
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 {
```

Figure 3.8: Вставка области текста

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



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

LOCAL HELLO=World
echo $HELLO
}

echo $HELLO
hello
function hello {
    echo $HELLO
```

Figure 3.9: Выделение области

и на этот раз вырежем её («Ctrl-w»)



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

LOCAL HELLO=World
echo $HELLO
}

hello
function hello {
    echo $HELLO
```

Figure 3.10: Вырезка области

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

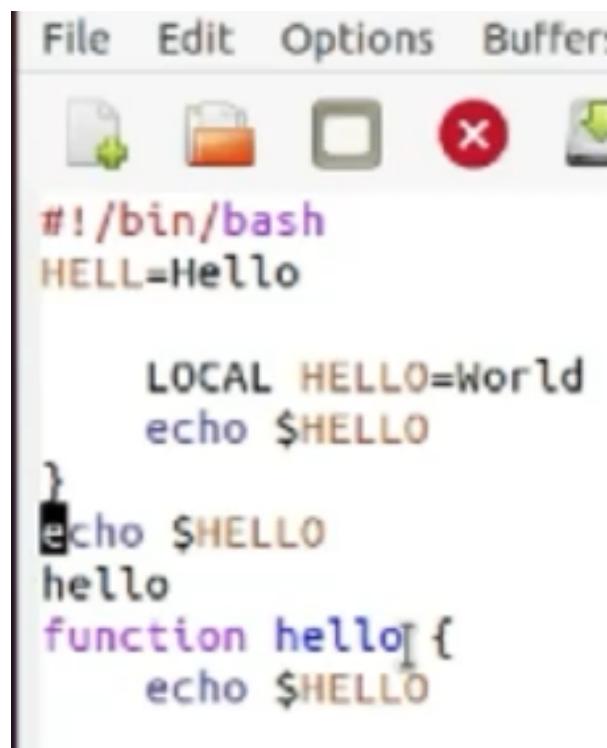


```
#!/bin/bash
HELL=Hello

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

Figure 3.11: Отмена последнего действия

6.1) Переместим курсор в начало строки («Ctrl-a»)

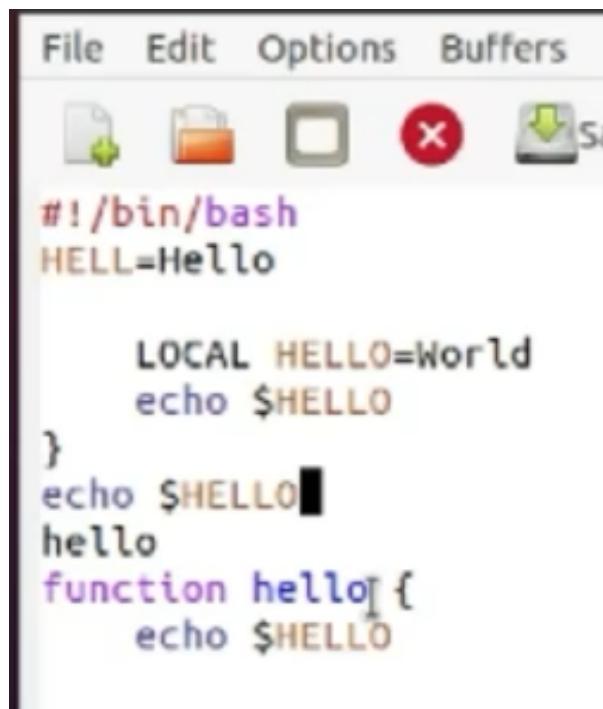


```
#!/bin/bash
HELL=Hello

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

Figure 3.12: Перемещение курсора

6.2) Переместим курсор в конец строки («Ctrl-e»)



The screenshot shows a terminal window with a menu bar containing "File", "Edit", "Options", and "Buffers". Below the menu are several icons: a file with a green plus sign, a folder with an orange plus sign, a square, a red circle with a white 'X', and a download icon. The main area of the terminal contains the following bash script code:

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

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

The cursor is positioned at the end of the line "echo \$HELLO".

Figure 3.13: Перемещение курсора

6.3) Переместим курсор в начало буфера («Alt-<»)

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

LOCAL HELLO=World
echo $HELLO
}

echo $HELLO
hello
function hello {
    echo $HELLO
```

Figure 3.14: Курсор в начале буфера

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

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

LOCAL HELLO=World
echo $HELLO
}

echo $HELLO
hello
function hello {
    echo $HELLO
```

Figure 3.15: Курсор в конце буфера

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

The screenshot shows the Emacs interface with a buffer list. The top menu bar includes File, Edit, Options, Buffers, Tools, Minibuf, and Help. Below the menu is a toolbar with icons for file operations like Save, Undo, and Cut/Paste. The main window displays a shell script named lab10.sh:

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

Below the script is a buffer list titled "Buffer List". It shows the following information:

CRM Buffer	Size	Mode	File
lab10.sh	114	Shell-script	[... ~/lab10.sh]
* *GNU Emacs*	1066	Fundamental	
scratch	145	Lisp Interaction	
Messages	687	Messages	

The bottom status bar indicates "U:%%- *Buffer List* All L1 (Buffer Menu)".

Figure 3.16: Список активных буферов

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

The screenshot shows the Emacs interface with a messages buffer. The top status bar says "U:%%- *Messages* Bot L16 (Messages)". The main window displays the following text:

```
For information about GNU Emacs and the GNU system, type C-h C-a.
lab07.sh has auto save data; consider M-x recover-this-file
Setting up indent for shell type bash
Indentation variables are now local.
Indentation setup for shell type bash
Mark set [3 times]
Auto-saving...done
```

Figure 3.17: Список активных буферов

7.3) Закроем это окно («Ctrl-x0»)

```

#!/bin/bash
HELL=Hello

LOCAL HELLO=World
echo $HELLO

echo $HELLO
hello
function hello {
    echo $HELLO
}

```

Figure 3.18: Закрытие окна

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

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»)



Figure 3.19: Переключение между буферами

8.2) В каждом из четырёх созданных окон откроем новый буфер (файл) и введем не сколько строк текста. Для этого предварительно создадим эти файлы с

помощью команд «touchexample1.txt», «touchexample2.txt», «touchexample3.txt», «touchexample4.txt»

```
nmbezruk@username:~$ touch example1.txt
nmbezruk@username:~$ touch example2.txt
nmbezruk@username:~$ touch example3.txt
nmbezruk@username:~$ touch example4.txt
nmbezruk@username:~$
```

Figure 3.20: Создание файлов

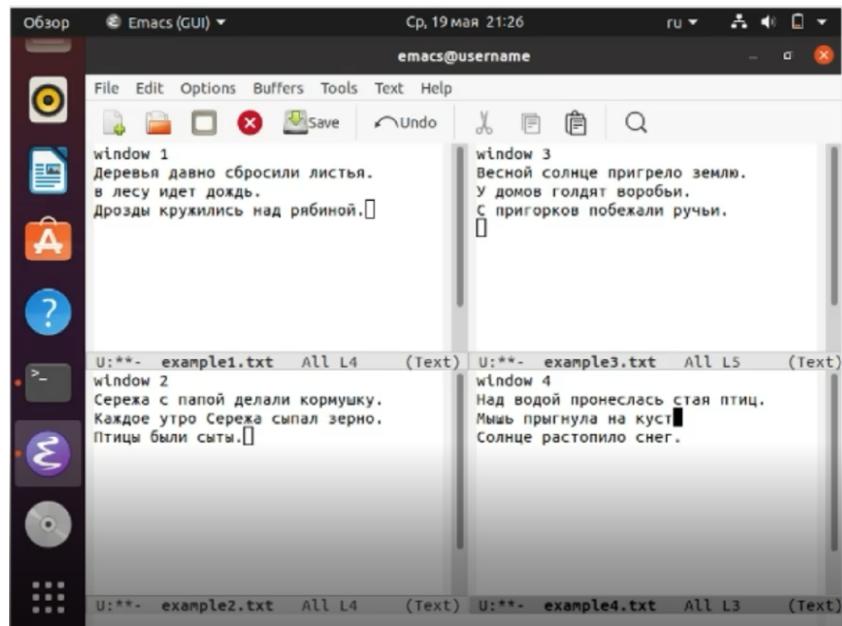


Figure 3.21: Вывод на экран

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

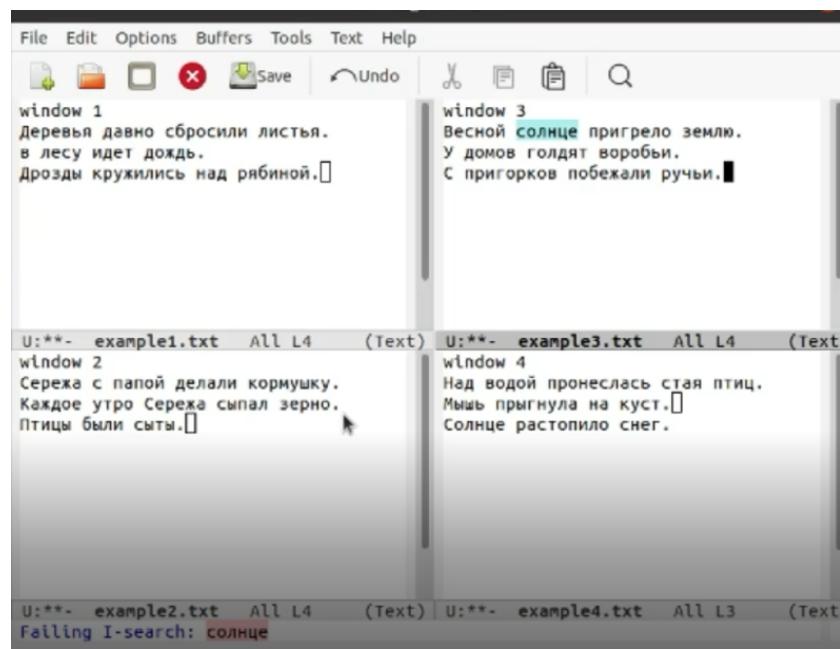


Figure 3.22: Поиск слова

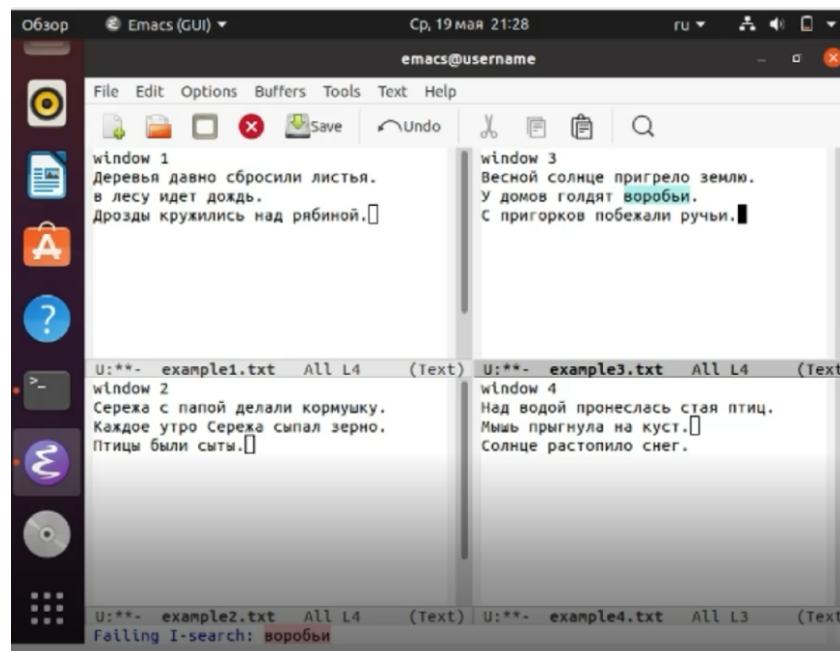


Figure 3.23: Поиск слова

9.2) Переключимся между результатами поиска, нажимая «Ctrl-s»

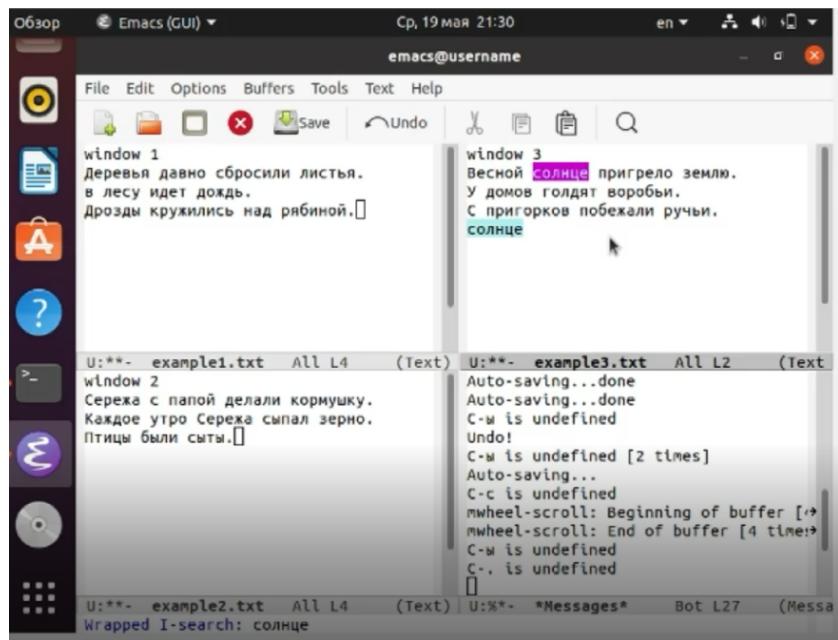


Figure 3.24: Переключение

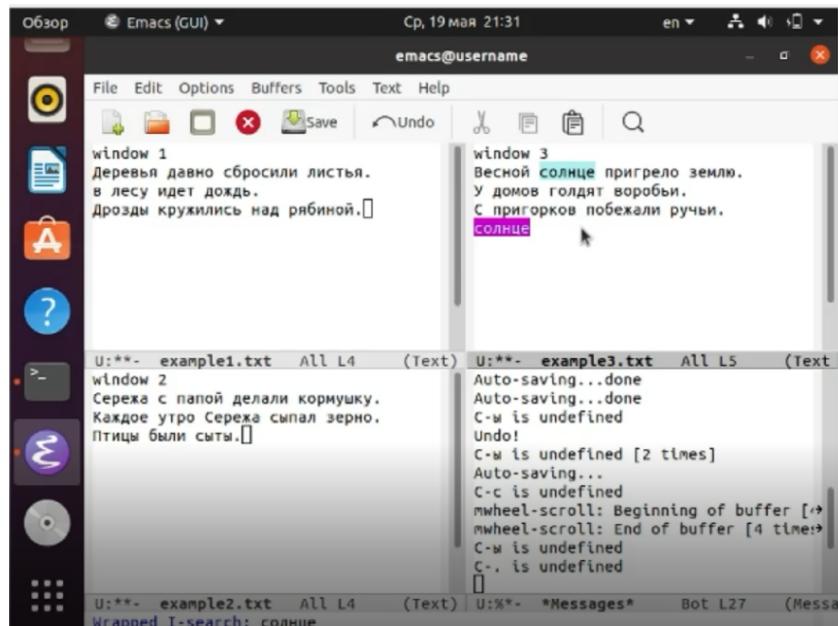


Figure 3.25: Переключение

9.3) Выйдем из режима поиска, нажав «Ctrl-g»

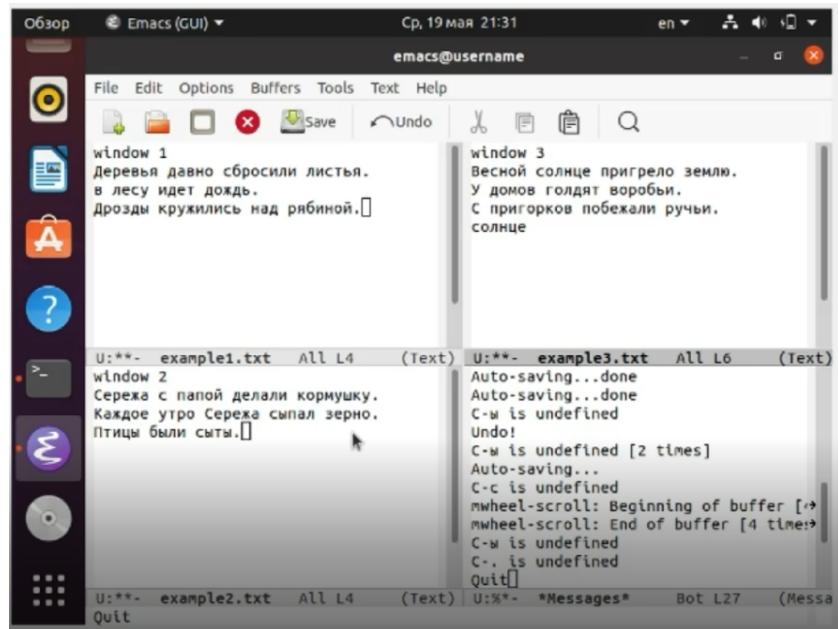


Figure 3.26: Переключение

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.

File Edit Options Buffers Tools Text Help

window 1
деревья давно сбросили листья.
в лесу идет дождь.
дрозды кружились над **рябиной**.

window 2
Сережа с папой делали кормушку.
Каждое утро Сережа сыпал зерно.
Птицы были сыты.□

U:**- example1.txt All L4 (Text)
U:**- example2.txt All L4 (Text)
Query replacing рябиной with ивой: (? f)

Figure 3.27: Замена

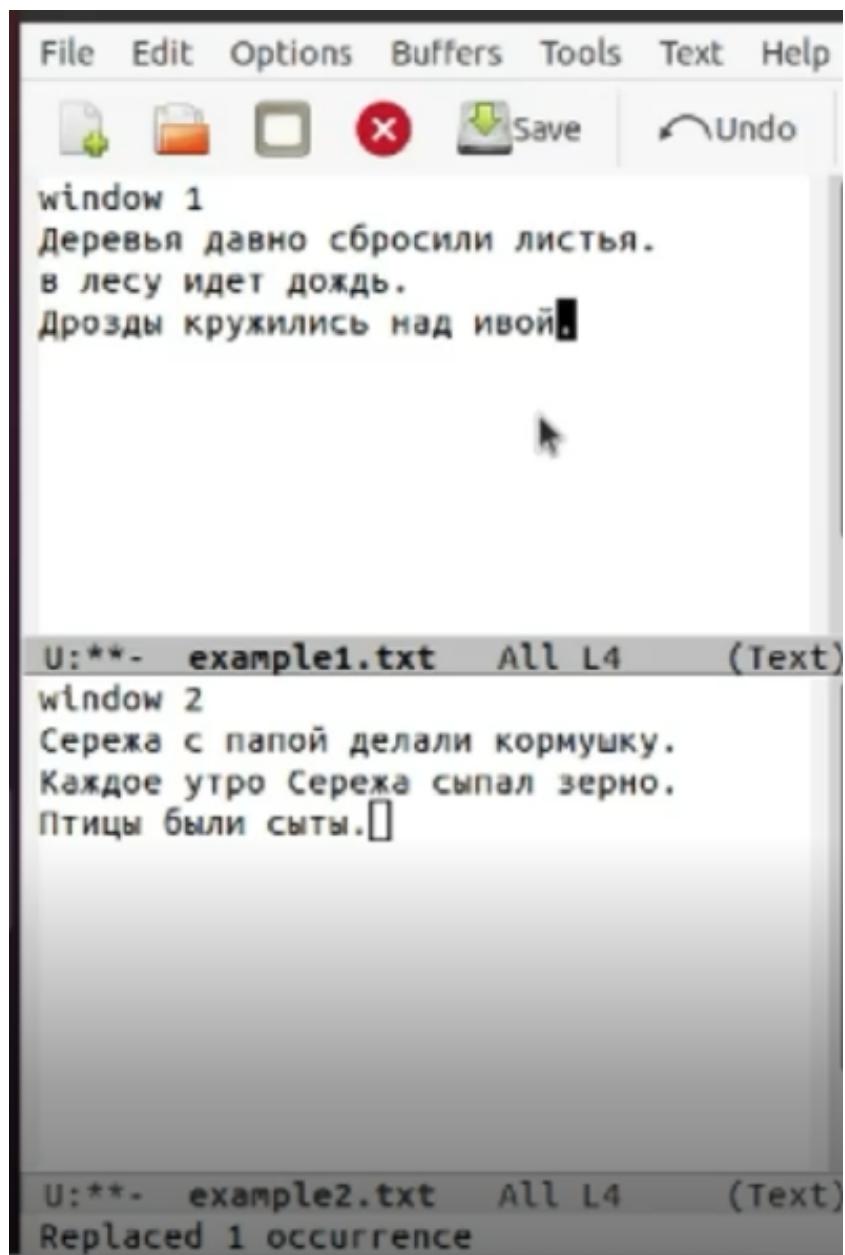


Figure 3.28: Замена

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.

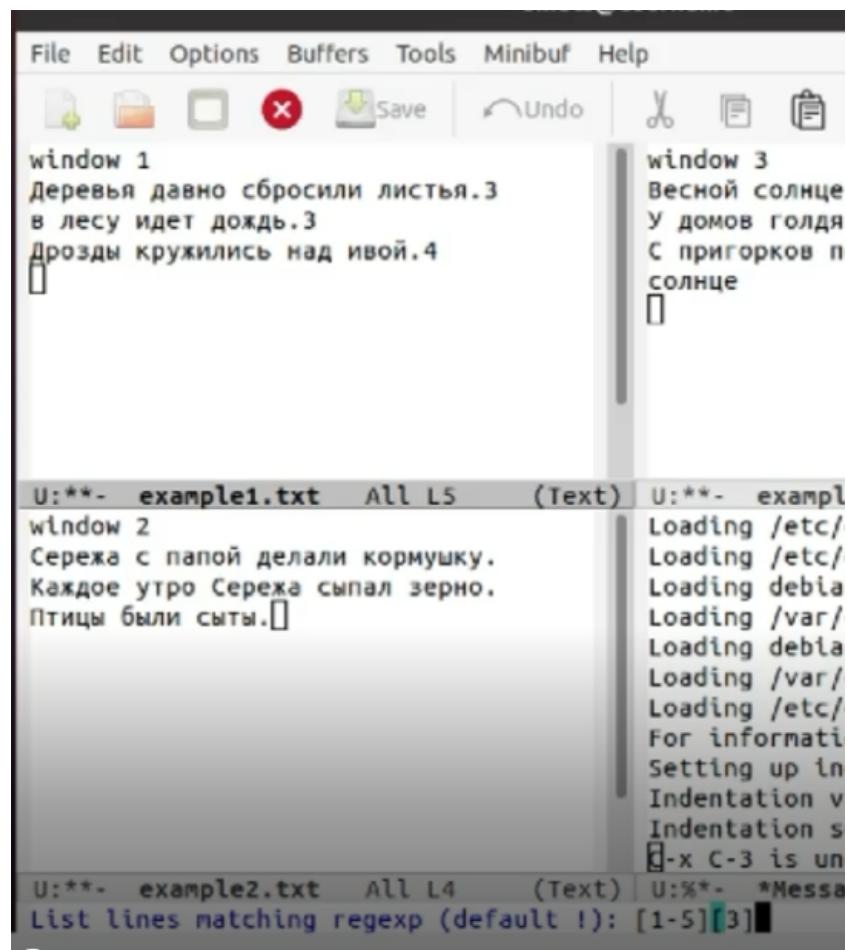


Figure 3.29: Замена

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 С 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 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 буфера: «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 «c», после – отпускаю обе клавиши нажимаю «|» С-сС-| сначала, удерживая «ctrl», нажимаю «c», 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, 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 неудобными.

5 Выводы

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