

Отчёт по лабораторной работе №11

Операционные системы

Овезов Мерген

Содержание

1 Цель работы	1
2 Задание	1
3 Теоретическое введение	1
4 Выполнение лабораторной работы	2
5 Выводы	10
6 Ответы на контрольные вопросы.....	10

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 Теоретическое введение

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:

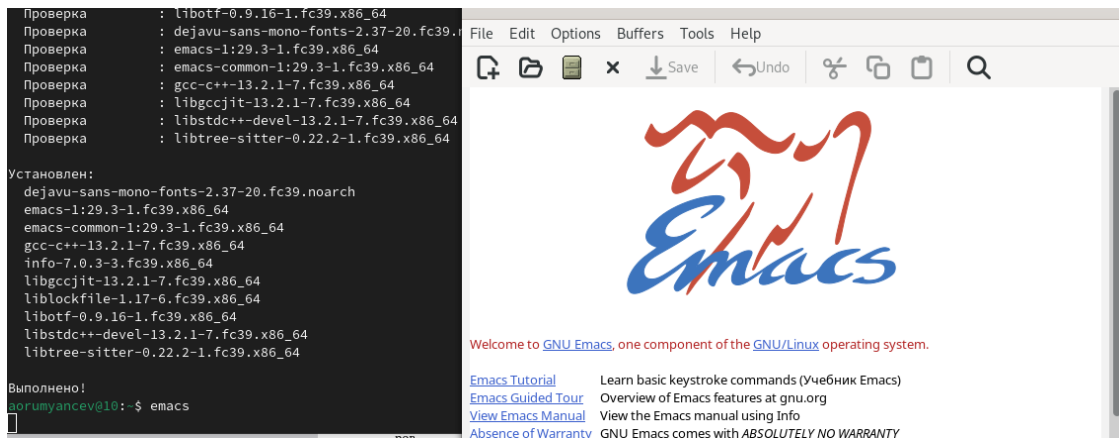
- текстовым редактором;
- программой для чтения почты и новостей Usenet;
- интегрированной средой разработки (IDE);
- операционной системой;

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.

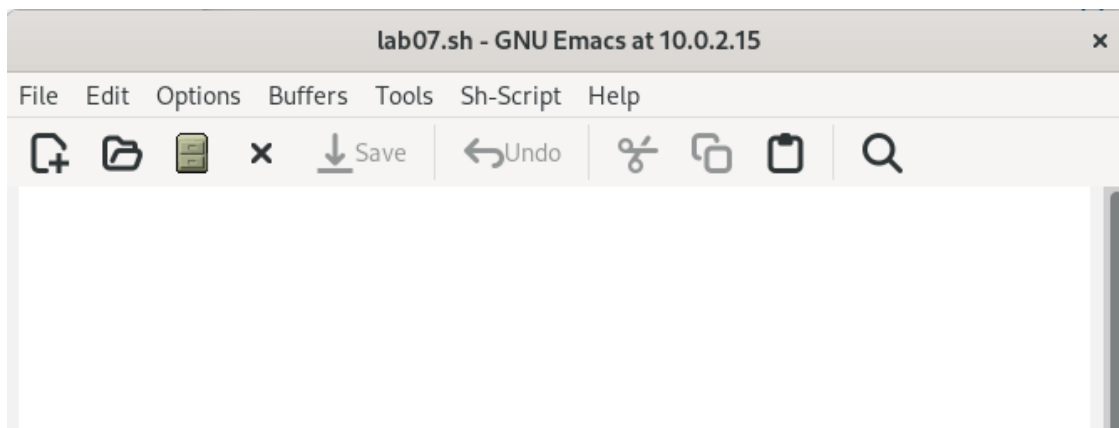
Первая версия редактора Emacs была написана в 70-х годах 20-го столетия Richard Stallman (Ричардом Столманом) как набор макросов для редактора TECO . В дальнейшем, уже будучи основателем Фонда Свободного программного обеспечения Free Software Foundation и проекта GNU, Stallman разработал GNU Emacs в развитие оригинального Emacs и до сих пор сопровождает эту программу. Emacs является одним из старейших редакторов. Он использовался тысячами программистов на протяжении последних 20 с лишним лет, для него создано много дополнительных пакетов расширений. Эти дополнения позволяют делать с помощью Emacs такие вещи, которые Stallman , вероятно, даже не считал возможными в начале своей работы над редактором.

4 Выполнение лабораторной работы

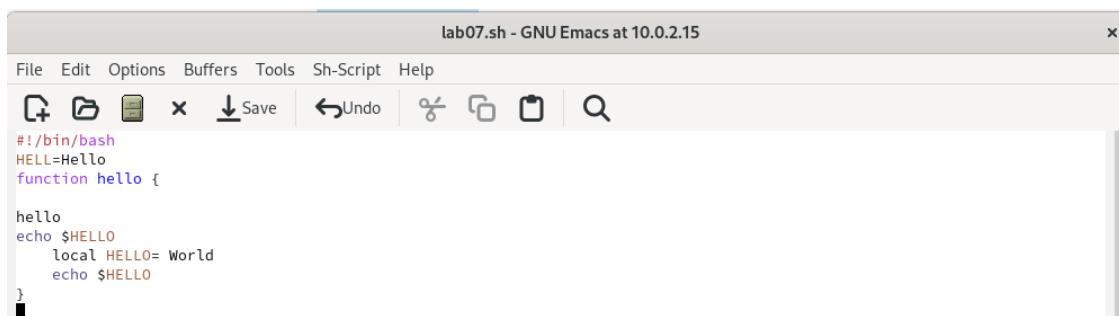
Открываю Emacs через терминал.



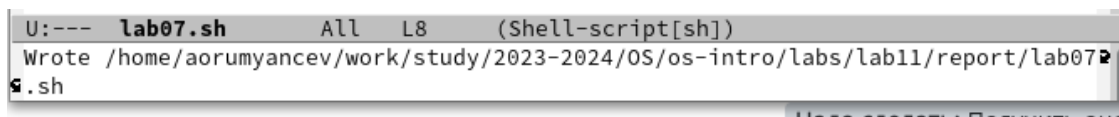
Создаю файл lab07.sh



Прописываю в файле текст программы.



Сохраняю файл с помощью комбинации C-x C-s.



Вырезаю одной командой целую строку (C-k)

```
    echo $HELLO
}

hello
```

U:*** lab07.sh All L7 (Shell-script[sh])

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

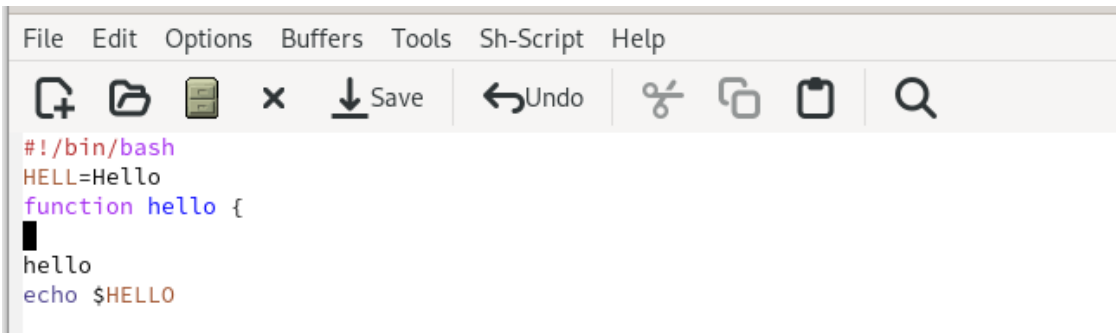
```
    echo $HELLO
}

hello
echo $HELLO
```

U:*** lab07.sh All L9 (Shell-script[sh])

Mark set

Выделяю область текста(C-space), копирую область в буфер обмена (M-w), вырезаю эту область с помощью C-w.



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

Вставляю область в конец файла (C-y).

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

U:*** lab07.sh All L10 (Shell-script[sh])

Mark set

Отменяю последнее действие C-/

```
#!/bin/bash
HELL=Hello
function hello {

hello
echo $HELLO
█
```

```
U:***- lab07.sh All L7 (Shell-script[sh])
Undo
```

Перевожу курсор в начало строки C-a

```
#!/bin/bash
HELL=Hello
function hello {

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


```

```
U:***- lab07.sh All L7 (Shell-script[sh])
```

Перемещаю курсор в конец строки C-e

```
#!/bin/bash
HELL=Hello
function hello {

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


```

```
U:***- lab07.sh All L7 (Shell-script[sh])
```

Перемещаю курсор в начало файла M-<

```
#!/bin/bash
HELL=Hello
function hello {

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

U:***- lab07.sh All L1 (Shell-script[sh])
```

Перемещаю курсор в конец файла M->

```
#!/bin/bash
HELL=Hello
function hello {

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

U:***- lab07.sh All L9 (Shell-script[sh])
```

Открываю список активных буферов в другой окне C-x C-b

```
U:***- lab07.sh Bot L9 (Shell-script[sh])
CRM Buffer Size Mode File
[ ] * lab07.sh 101 Shell-script[sh] ~/work/study/2023-2024/OS/os-in
% *GNU Emacs* 734 Fundamental
*scratch* 145 Lisp Interaction
%* *Messages* 1176 Messages
%* *Async-native-compile... 165 Fundamental
U:%%%- *Buffer List* All L1 (Buffer Menu)
C-x C-b
```

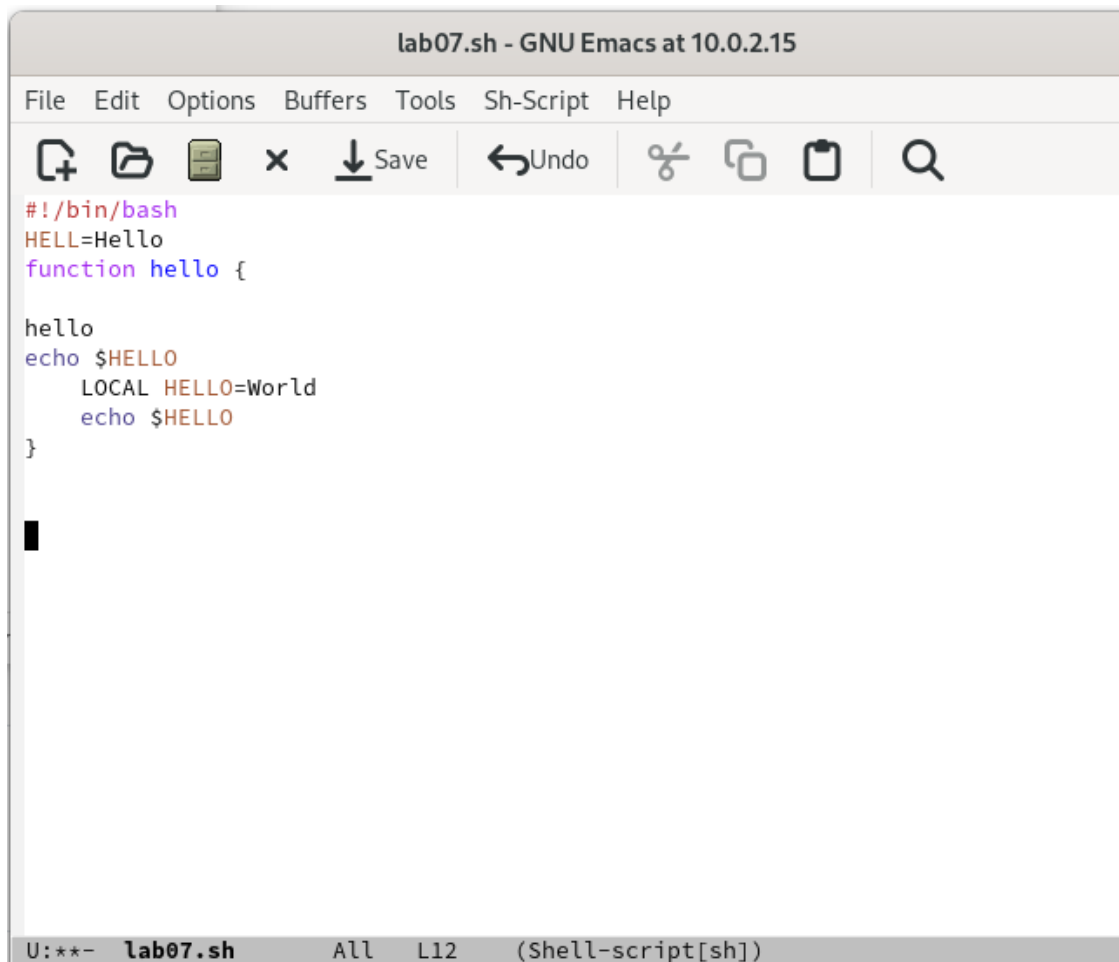
Переключаюсь на другой буфер C-x o

```
U:***- lab07.sh All L13 (Shell-script[sh])
#!/bin/bash
HELL=Hello
function hello {

hello
echo $HELLO
    LOCAL HELLO=World

U:***- lab07.sh Top L7 (Shell-script[sh])
```

Закрываю окно другого буфера C-x 0

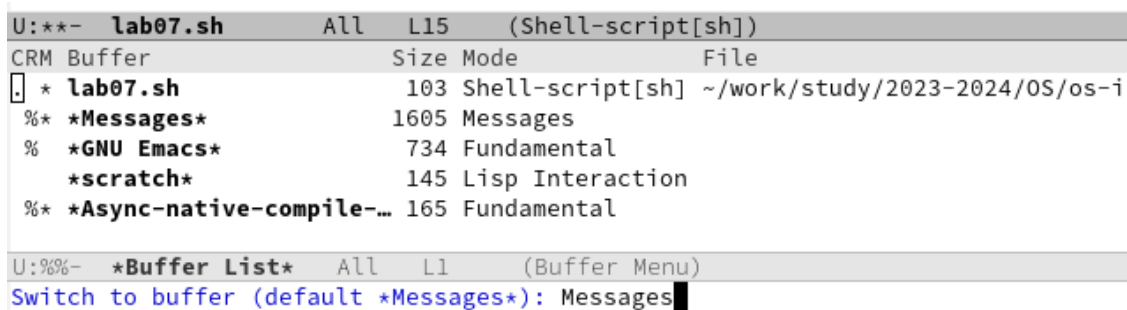


The screenshot shows the GNU Emacs editor interface. The title bar reads "lab07.sh - GNU Emacs at 10.0.2.15". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for file operations, saving, undo, and search. The main text area displays a shell script with the following content:

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

The status bar at the bottom shows "U:***- lab07.sh All L12 (Shell-script[sh])".

Открываю другой буфер без вывода их списка на экран с помощью C-x b

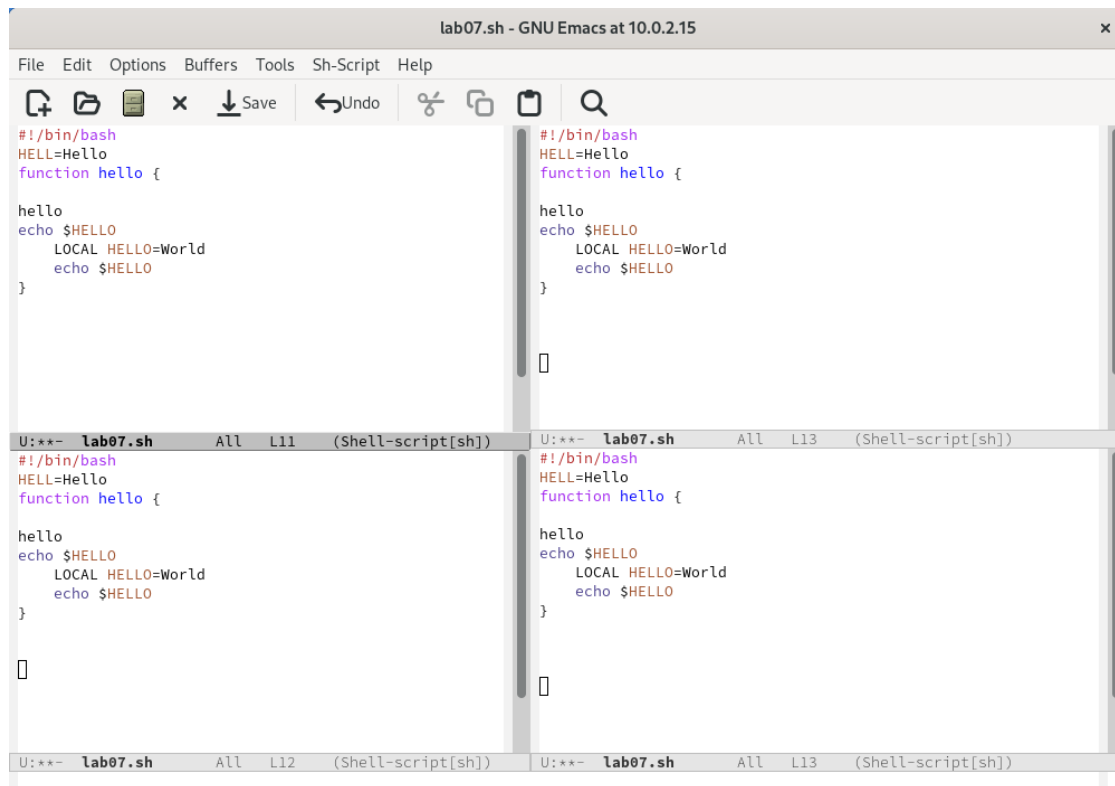


The screenshot shows the GNU Emacs editor interface with the Buffer List menu open. The menu lists several buffers:

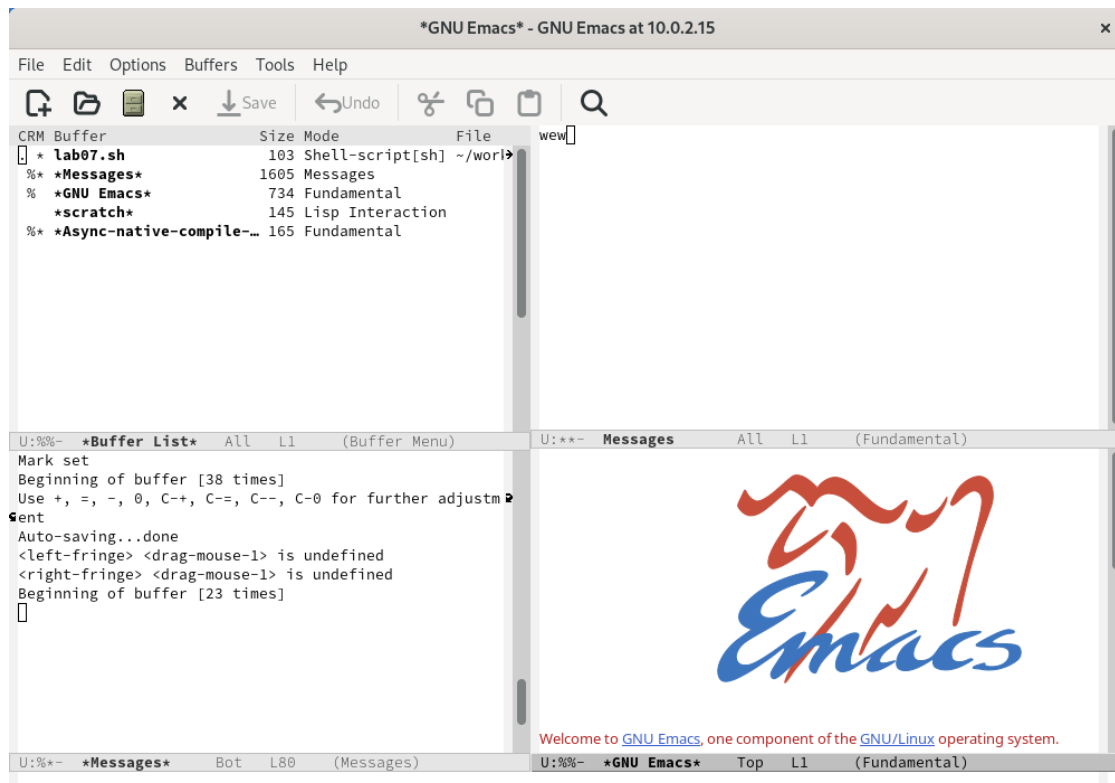
CRM Buffer	Size	Mode	File
* lab07.sh	103	Shell-script[sh]	~/work/study/2023-2024/OS/os-i
%* *Messages*	1605	Messages	
% *GNU Emacs*	734	Fundamental	
scratch	145	Lisp Interaction	
%* *Async-native-compile-...	165	Fundamental	

The status bar at the bottom shows "U:%%- *Buffer List* All L1 (Buffer Menu)". The prompt "Switch to buffer (default *Messages*):" is visible, followed by the text "Messages".

Делю фрейм на 4 части: сначала на два окна по вертикали(C-x 3), а затем каждое из этих окон на две части по горизонтали(C-x 2).



В каждой из четырех созданных окон с помощью C-x b открыл разные буферы.



Перехожу в режим поиска с помощью C-s, ищу слова в тексте, они подсвечиваются.


```
U:%%- *Buffer List* All L1 (Buffer Menu)
Mark set
Beginning of buffer [38 times]
Use +, =, -, 0, C+, C=, C--, C-0 for further adjustment
Sent
Auto-saving...done
<left-fringe> <drag-mouse-1> is undefined
<right-fringe> <drag-mouse-1> is undefined
Beginning of buffer [23 times]

U:%%- *Messages* Bot L80 (Messages Isearch)
Failing I-search: left
```

С помощью той же комбинации C-s я могу перемещаться по результатам поиска.

```
U:%%- *Buffer List* All L1 (Buffer Menu)
Mark set
Beginning of buffer [38 times]
Use +, =, -, 0, C+, C=, C--, C-0 for further adjustment
Sent
Auto-saving...done
<left-fringe> <drag-mouse-1> is undefined
<right-fringe> <drag-mouse-1> is undefined
Beginning of buffer [23 times]
C-c C-s is undefined
ESC <mouse-1> is undefined
[]

U:%%- *Messages* Bot L82 (Messages)
I-search: ABSOLUTELY

U:%%- *GNU Emacs* 1% L8 (Fundamental Isearch)
Welcome to GNU Emacs, one component of the GNU/Linux operating system.

Emacs Tutorial      Learn basic keystroke commands (Учебник Emacs)
Emacs Guided Tour   Overview of Emacs features at gnu.org
View Emacs Manual    View the Emacs manual using Info
Absence of Warranty  GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions   Conditions for redistributing and changing Emacs
Ordering Manuals     Purchasing printed copies of manuals

To start...  Open a File      Open Home Directory
             Customize Startup Explore Packages
To quit a partially entered command, type Control-g.

This is GNU Emacs 29.3 (build 1, x86_64-redhat-linux-gnu, GTK+ Version
3.24.41, cairo version 1.18.0) of 2024-03-25
```

С помощью C-g выхожу из режима поиска, снимается выделение.

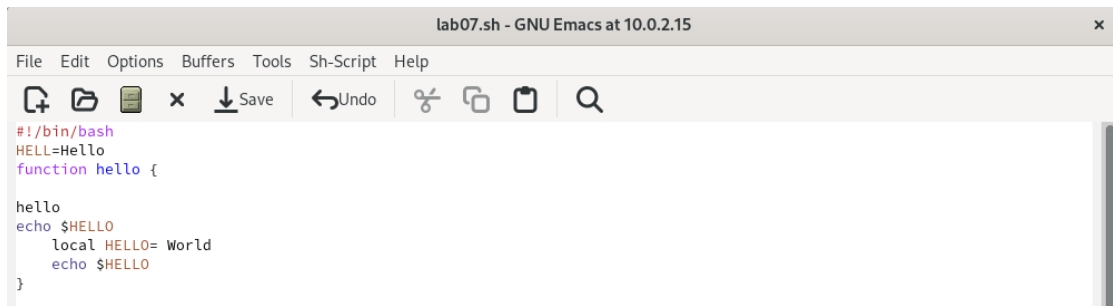
```
U:%%- *Messages* Bot L104 (Messages)
Quit

U:%%- *GNU Emacs* Bot L4 (Fundamental)
This is GNU Emacs 29.3 (build 1, x86_64-redhat-linux-gnu, GTK+ version
3.24.41, cairo version 1.18.0) of 2024-03-25
```

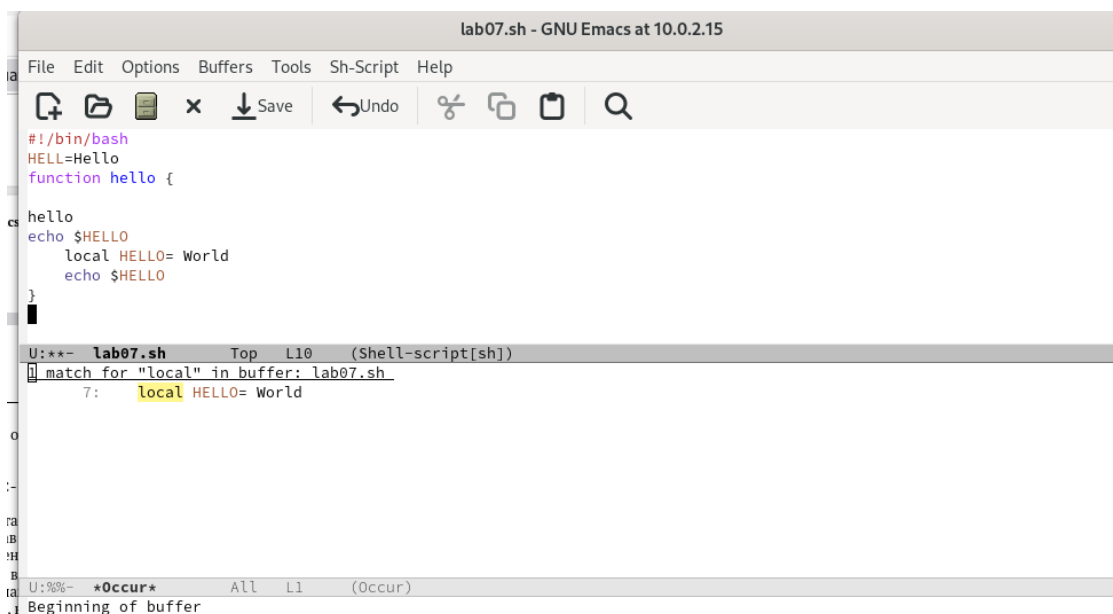
Перехожу в режим поиска и замены с помощью M-%, ввожу какое слово хочу заменить, затем ввожу на какое хочу заменить

```
U:%%- lab07.sh All L14 (Shell-script[sh])
Query replace LOCAL with: local
```

Видим, что слова были заменены успешно



С помощью M-s о перехожу в другой режим поиска. Он отличается от предыдущего тем, что выводит результат в отдельном окне от окна буфера.



5 Выводы

В ходе данной лабораторной работы я познакомился с ос Linux, получил практические навыки работы с редактором Emacs.

6 Ответы на контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. Написан на языке высокого уровня Lisp.

2. Какие особенности данного редактора могут сделать его сложным для освоения новичком?

Большое разнообразие сложных комбинаций клавиш, которые необходимы для редактирования файла и в принципе для работа с Emacs.

3. Своими словами опишите, что такое буфер и окно в терминологии emacs'а.

Буфер - это объект в виде текста. Окно - это прямоугольная область, в которой отображен буфер.

4. Можно ли открыть больше 10 буферов в одном окне?

Да, можно.

5. Какие буферы создаются по умолчанию при запуске emacs?

Emacs использует буферы с именами, начинающимися с пробела, для внутренних целей. Отчасти он обращается с буферами с такими именами особым образом — например, по умолчанию в них не записывается информация для отмены изменений.

6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?

Ctrl + c, а потом | и Ctrl + c Ctrl + |

7. Как поделить текущее окно на две части?

С помощью команды Ctrl + x 3 (по вертикали) и Ctrl + x 2 (по горизонтали).

8. В каком файле хранятся настройки редактора emacs?

Настройки emacs хранятся в файле .emacs, который хранится в домашней директории пользователя. Кроме этого файла есть ещё папка .emacs.

9. Какую функцию выполняет клавиша и можно ли её переназначить?

Выполняет функцию стереть, думаю можно переназначить.

10. Какой редактор вам показался удобнее в работе vi или emacs? Поясните почему.

Для меня удобнее был редактор Emacs, так как у него есть командная оболочка. А vi открывается в терминале, и выглядит своеобразно.