

Отчёт по лабораторной работе №11

Текстовый редактор emacs

Зоригоо Номун

Содержание

1	Цель работы	1
2	Задание	1
3	Выполнение лабораторной работы.....	1
4	Выводы.....	13
5	Ответы на контрольные вопросы.....	13

1 Цель работы

Получить практические навыки работы с редактором Emacs.

2 Задание

1. Выполнить основные команды emacs

3 Выполнение лабораторной работы

Для данной работы, мне надо была установить Emacs:

```
(nomun@Zorigoo)-[~]  
$ sudo apt install emacs  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done
```

Рис. 1: Установка Emacs

Выполнив Emacs в командной строке, я открыла текстовый редактор:

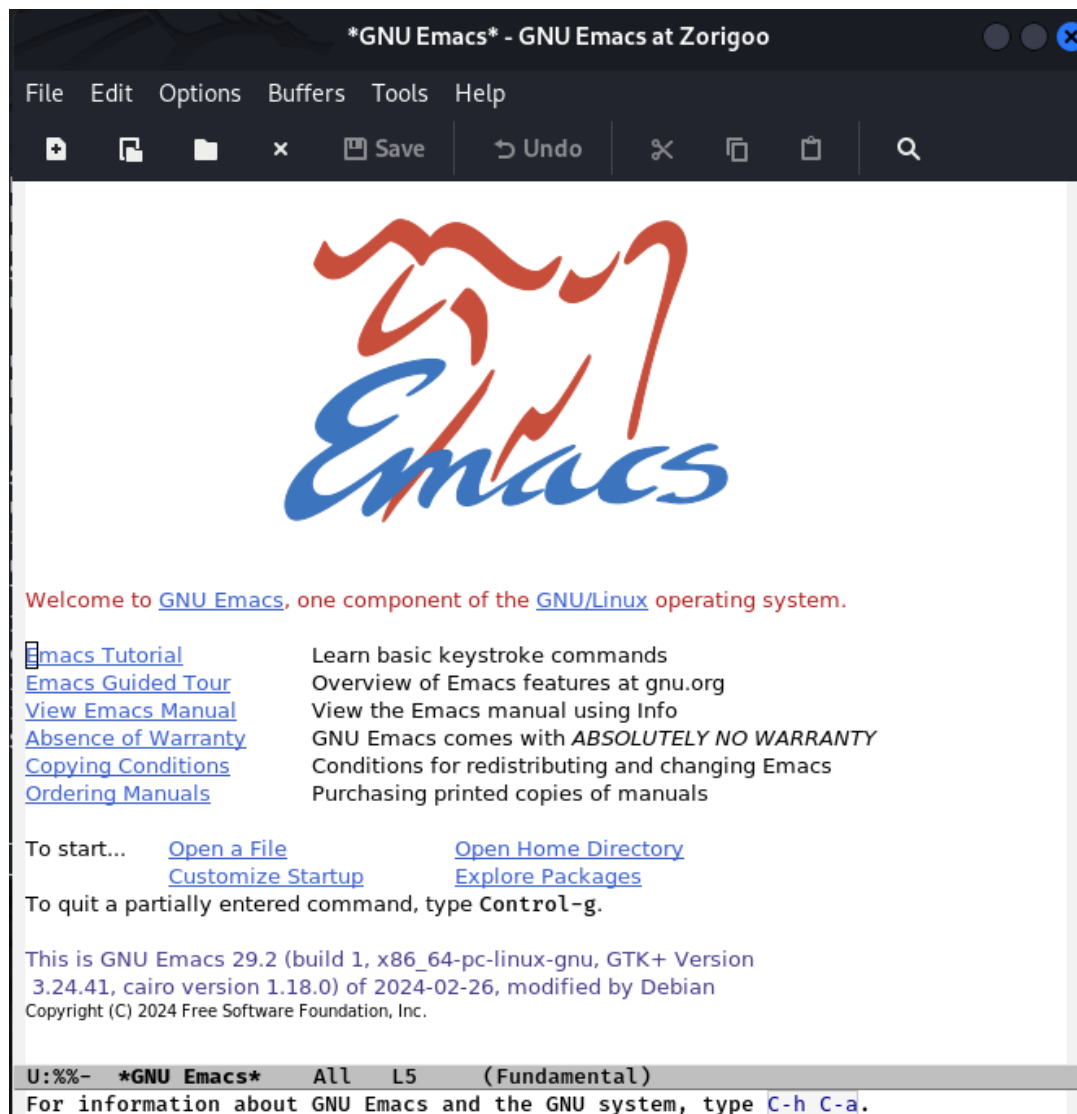


Рис. 2: Emacs

С помощью комбинации Ctrl-x Ctrl-f, создала файл lab07.sh:

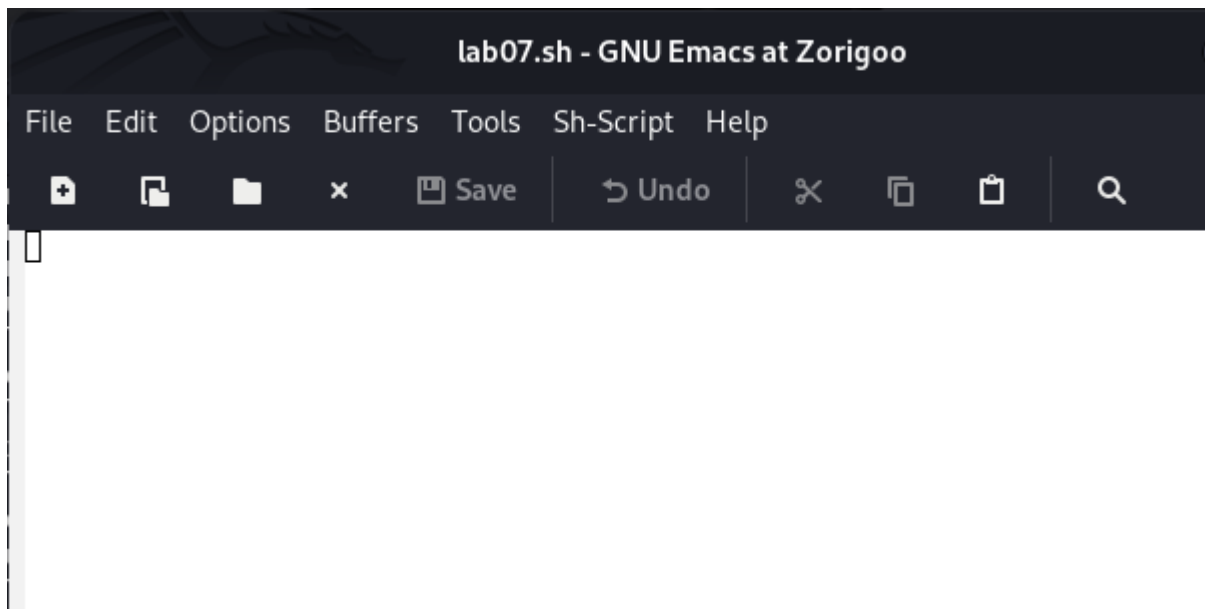


Рис. 3: Созданный файл

Я написала некоторый текст в этом же файле (lab07.sh). После этого сохранила файл с помощью комбинации Ctrl-x Ctrl-s:

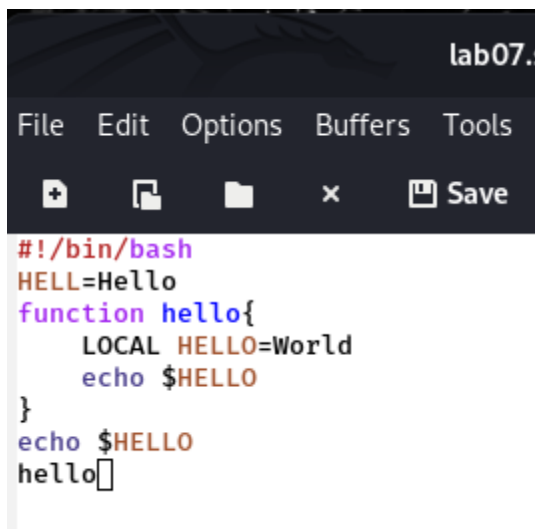
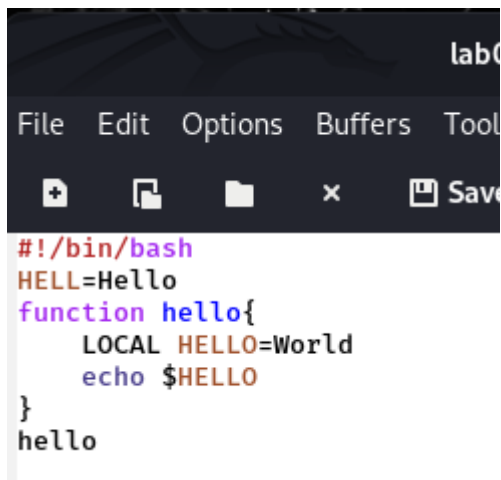


Рис. 4: текст в lab07.sh

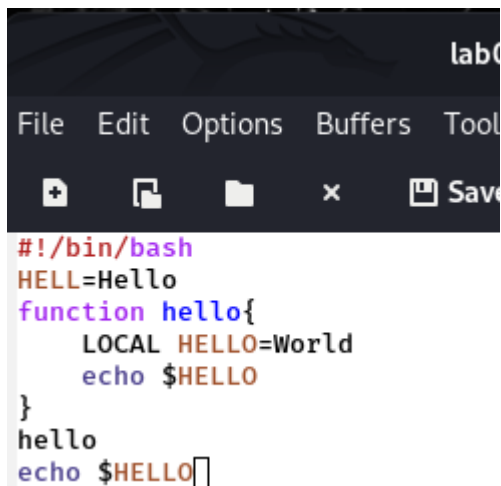
Одной командой вырезала целую строку (C-k):



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

Рис. 5: Вырезание строки

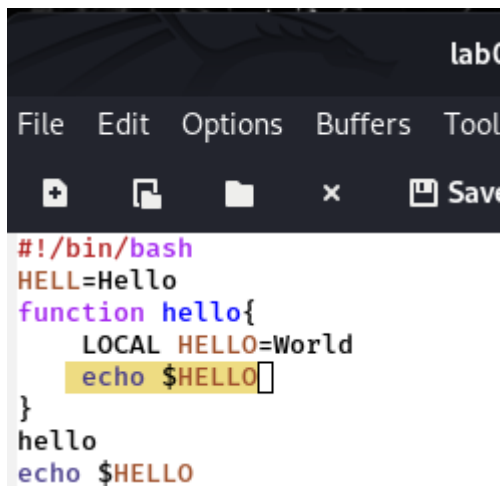
С помощью C-у вставила эту строку в конец файла:



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

Рис. 6: Перемещение строку в конец файла

Выделила область текста (C-space):

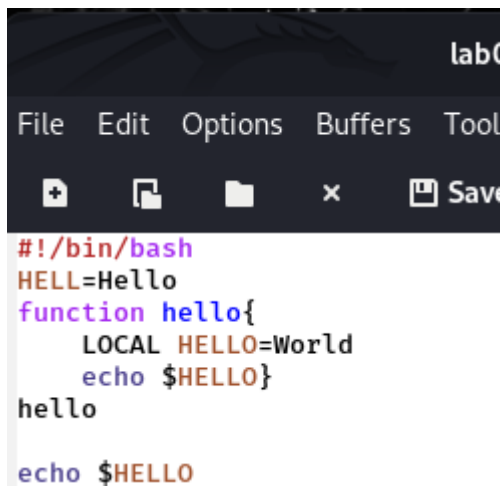
A screenshot of a terminal editor window titled 'lab0'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', and 'Tools'. The toolbar shows icons for adding a file, opening a file, saving a file, closing a file, and saving all files. The script content is as follows:

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

The line 'echo \$HELLO' inside the function 'hello' is highlighted in yellow.

Рис. 7: Выделенный текст

Скопировала область в буфер обмена (M-w) и вставила ее в конец файла:

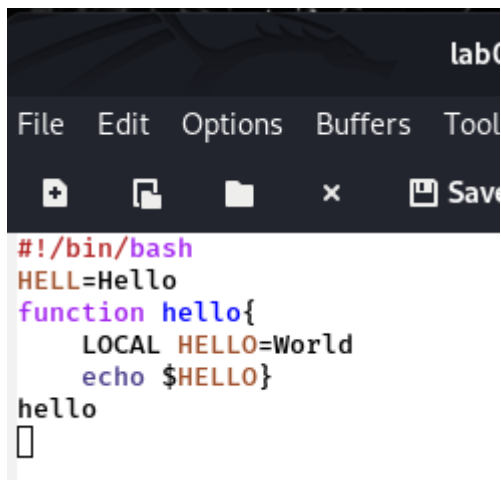
A screenshot of the same terminal editor window. The script content is now:

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

The new line 'echo \$HELLO' at the bottom of the script is highlighted in yellow.

Рис. 8: копирование и вставка

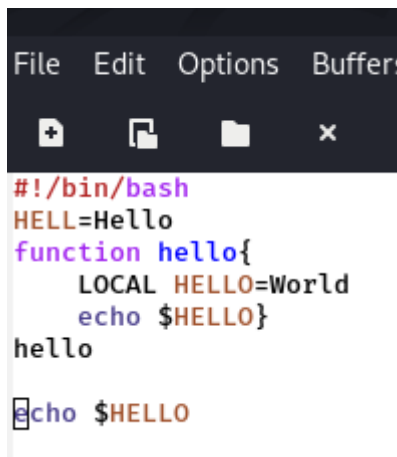
Выделила эту же область и на этот раз вырезала её (C-w):



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

Рис. 9: Вырезанная область

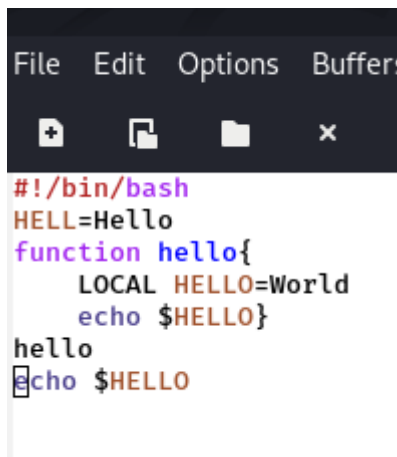
С помощью C-/ отменила последнее действие:



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

Рис. 10: отмена действие

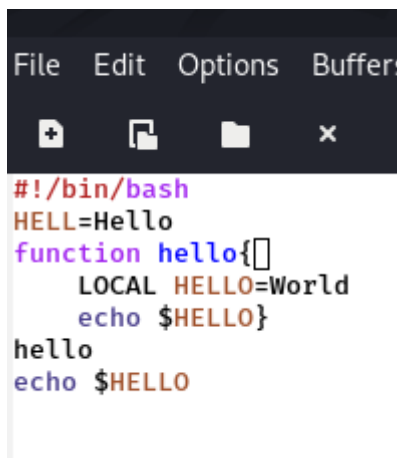
С помощью C-a переместила курсор в начало строки:



```
File Edit Options Buffers
+ [ ] x
#!/bin/bash
HELL=Hello
function hello{
    LOCAL HELLO=World
    echo $HELLO}
hello
echo $HELLO
```

Рис. 11: Перемещение курсор в начало строки

С помощью C-е переместила курсор в конец строки:



```
File Edit Options Buffers
+ [ ] x
#!/bin/bash
HELL=Hello
function hello{
    LOCAL HELLO=World
    echo $HELLO}
hello
echo $HELLO
```

Рис. 12: Перемещение курсор в конец строки

Переместила курсор в начало и конец буфера с помощью M-< и M-> соответственно:

```
U:***- lab07.sh ALL L/ (Shell-script[sh])
Mark set
Buffer is read-only: #<buffer *Messages*>
C-, is undefined
m is undefined
C-, is undefined
scroll-down-command: Beginning of buffer [4 times]
M is undefined
Mark set
M is undefined
Mark set
M is undefined
Mark set
M is undefined
Mark set
M is undefined
Mark set
█
```

Рис. 13: Перемещение курсор в буфере

Выводила список активных буферов на экран с помощью C-x C-b:

```
U:***- lab07.sh ALL L8 (Shell-script[sh])
CRM Buffer Size Mode File
█ * lab07.sh 99 Shell-script[sh] ~/lab07.sh
%* *Messages* 3838 Messages ~/*Messages*
* TUTORIAL 47846 Fundamental
% nomun 2362 Dired by name ~/
% *GNU Emacs* 734 Fundamental
* *scratch* 145 Lisp Interaction
%* *Async-native-compile... 4087 Fundamental
* *interpretation* 221 Comint:exit [12...
```

Рис. 14: Активные буферы

С помощью C-x o переместилась во вновь открытое окно со списком открытых буферов и переключилась на другой буфер:

```
U:***- lab07.sh ALL L8 (Shell-script[sh])
CRM Buffer Size Mode File
. * lab07.sh 99 Shell-script[sh] ~/lab07.sh
%* *Messages* 3906 Messages ~/*Messages*
* TUTORIAL 47846 Fundamental
% nomun 2362 Dired by name ~/
% *GNU Emacs* 734 Fundamental
* *scratch* 145 Lisp Interaction
%* *Async-native-compile... 4268 Fundamental
* *interpretation* 221 Comint:exit [12...
```

Mode: Fundamental

Рис. 15: список открытых буферов

С помощью C-x 0 закрыла окно со списком открытых буферов:

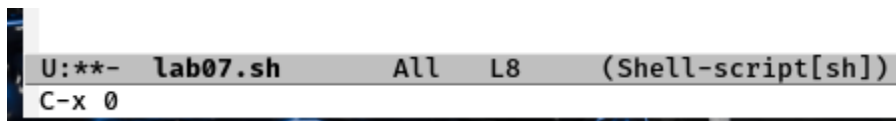


Рис. 16: Заккрытие окно

Без вывода списка буферов, я переключилась между буферами:

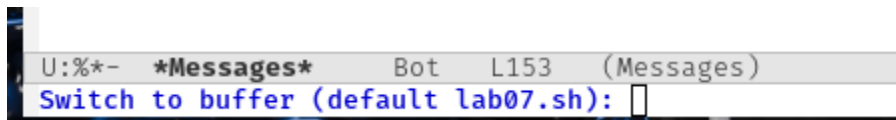


Рис. 17: Переключение между буферами

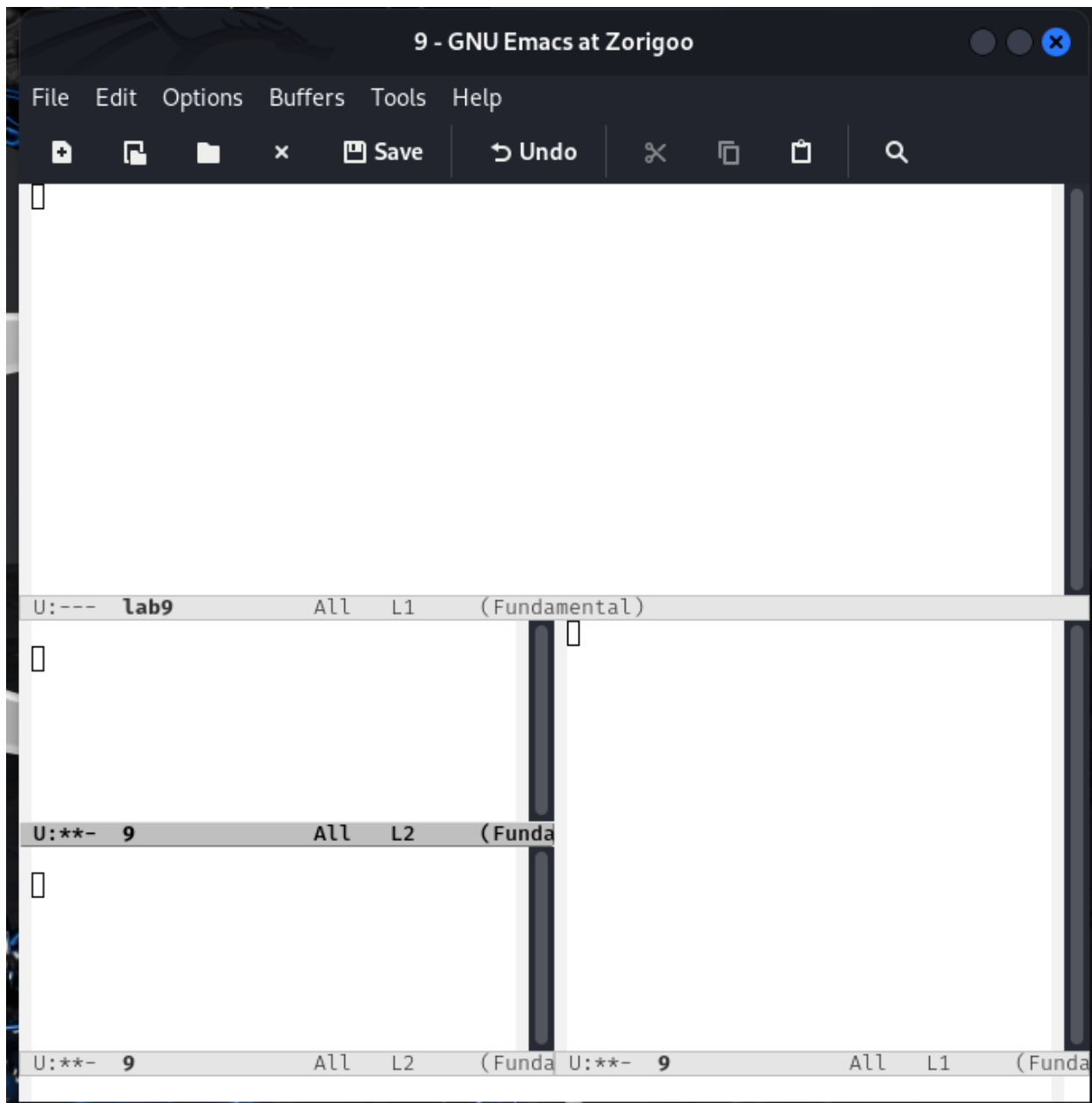


Рис. 18: Новый буфер

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

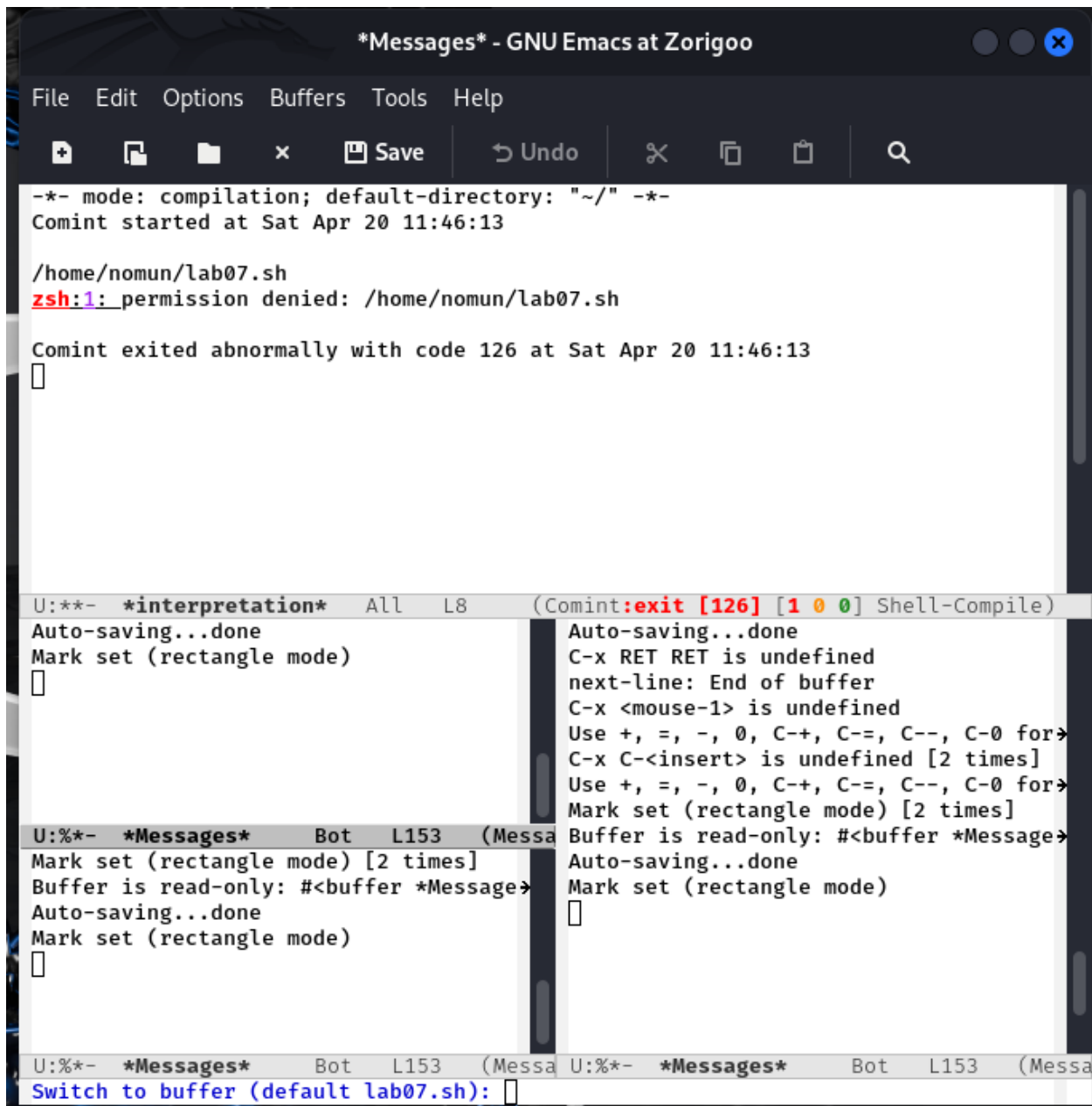


Рис. 19: Фрейм разделённый на 4 окна

В каждом из четырёх созданных окон открыла новый буфер:

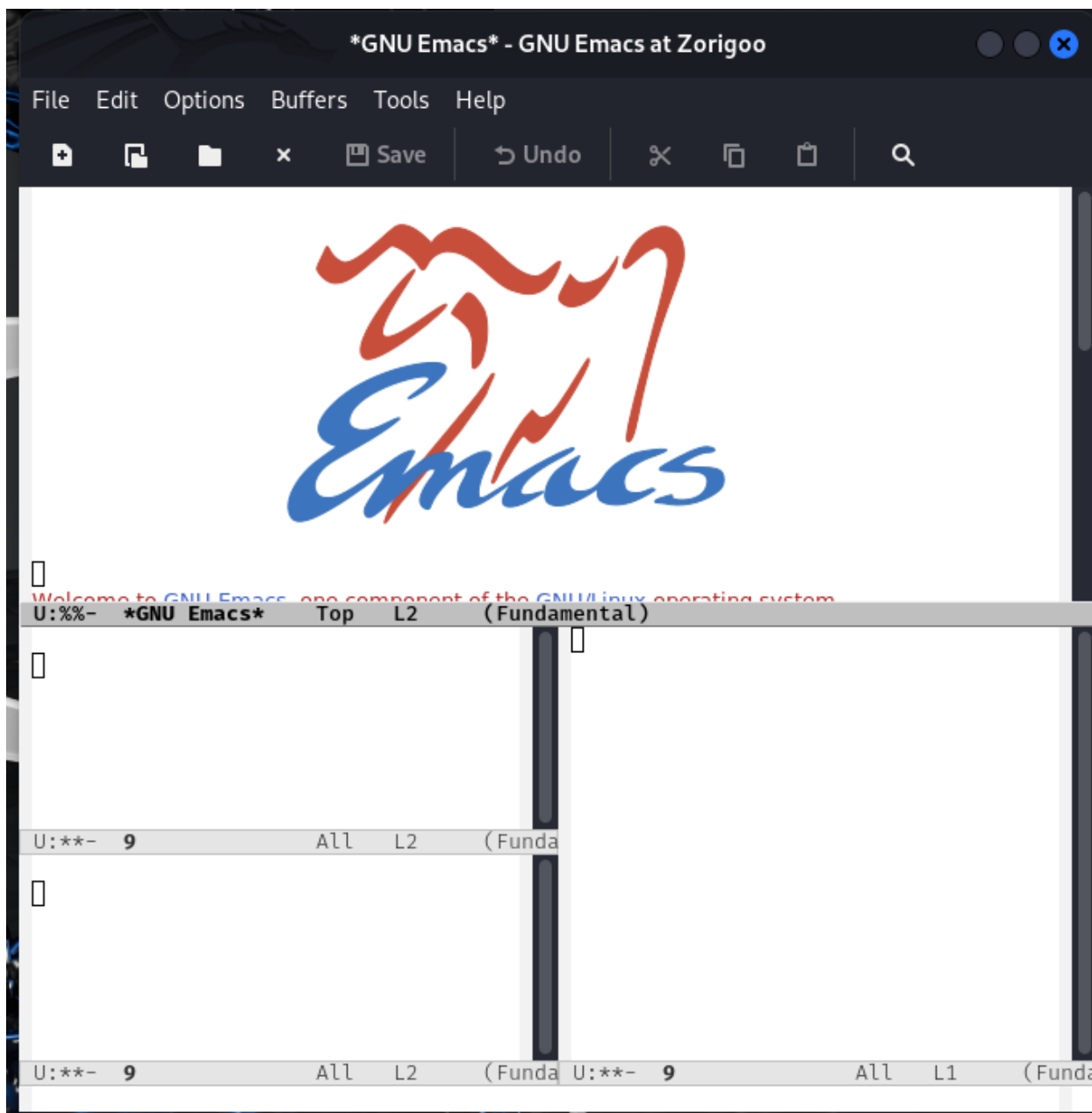


Рис. 20: Новые буферы

Переключилась в режим поиска (C-s) и искала Indent:

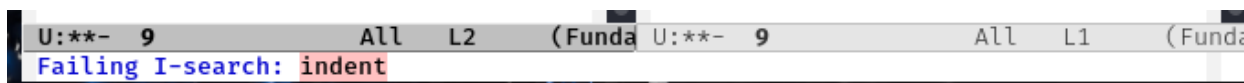


Рис. 21: Режим поиска

Переключалась между результатами поиска, нажимая C-s и вышла из режима поиска, нажав C-g:



Рис. 22: Переключение между результатами

4 Выводы

При выполнении данной работы я получила практические навыки работы с Emacs.

5 Ответы на контрольные вопросы

1. Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. Написан на языке высокого уровня Lisp.
2. Большое разнообразие сложных комбинаций клавиш, которые необходимы для редактирования файла и в принципе для работы с Emacs.
3. Буфер - это объект в виде текста. Окно - это область, в которой отображен буфер.
4. Да, можно.
5. 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. Ctrl + c, а потом | и Ctrl + c Ctrl + |
7. С помощью команды Ctrl + x 3 (по вертикали) и Ctrl + x 2 (по горизонтали).
8. Настройки emacs хранятся в файле .emacs, который хранится в домашней директории пользователя. Кроме этого файла есть ещё папка .emacs.
9. Выполняет функцию стереть, думаю можно переназначить.
10. Для меня удобнее был редактор Emacs, так как у него есть командная оболочка. А vi открывается в терминале, и выглядит своеобразно.