

Отчёт по лабораторной работе №11

Текстовый редактор emacs

Зоригоо Номун

Содержание

- **Цель работы**

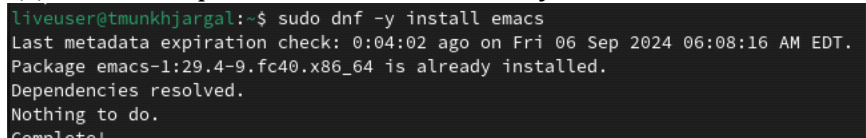
Получить практические навыки работы с редактором Emacs.

- **Задание**

- Выполнить основные команды emacs

- **Выполнение лабораторной работы**

Для данной работы, мне надо была установить Emacs:



```
liveuser@tmunkhjargal:~$ sudo dnf -y install emacs
Last metadata expiration check: 0:04:02 ago on Fri 06 Sep 2024 06:08:16 AM EDT.
Package emacs-1:29.4-9.fc40.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

Рис. 1: Установка Emacs

Выполнив Emacs в командной строке, я открыла текстовый редактор:

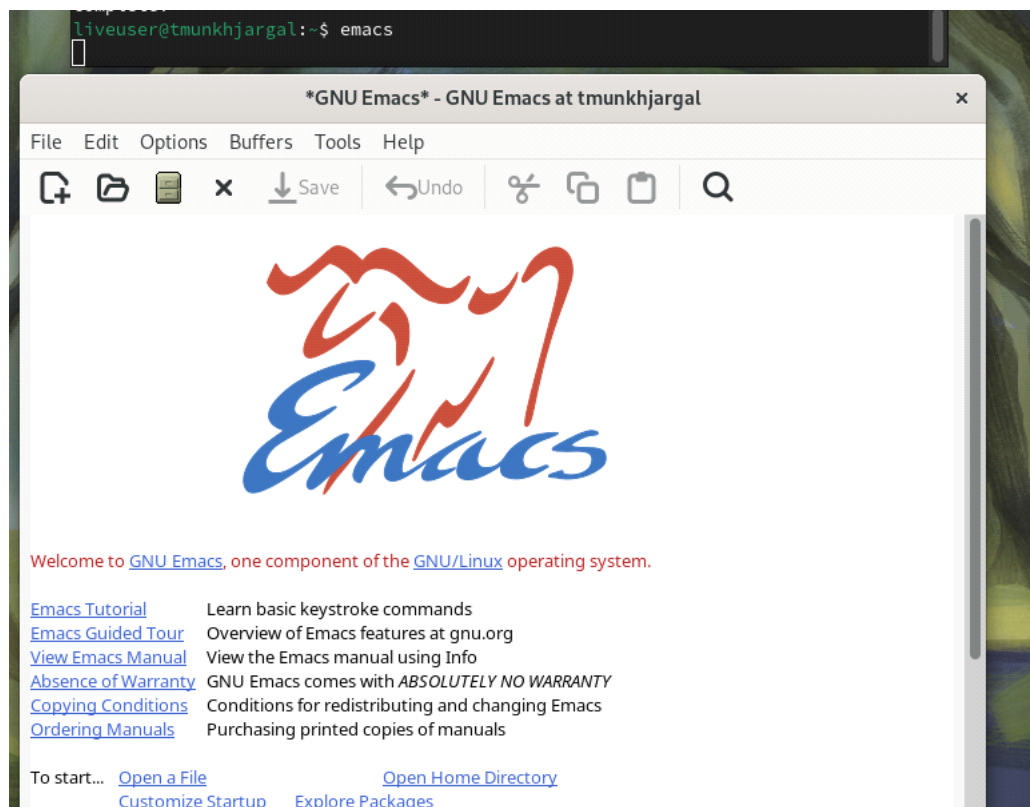


Рис. 2: Emacs

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

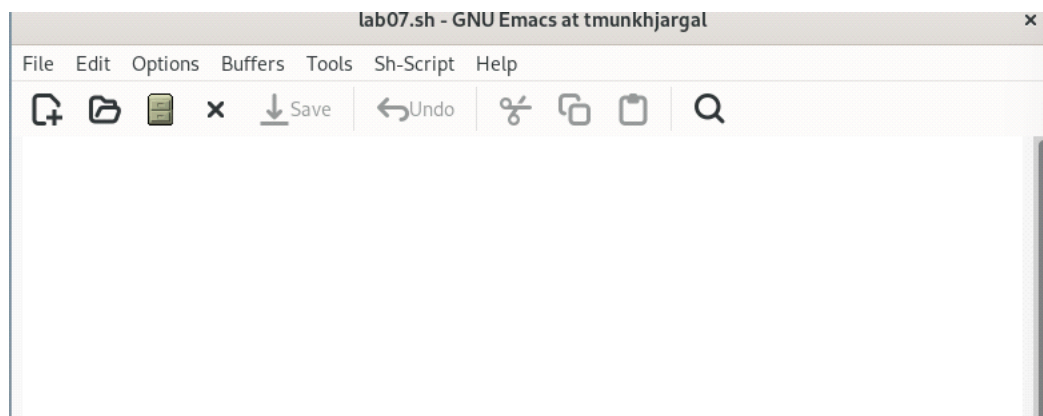
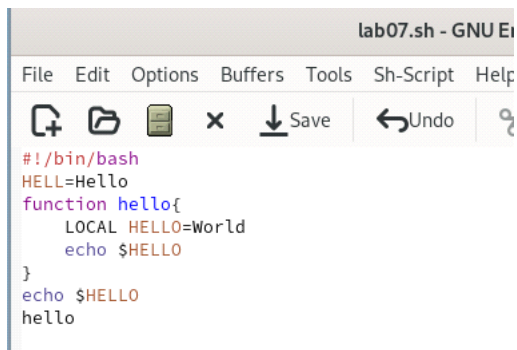


Рис. 3: Созданный файл

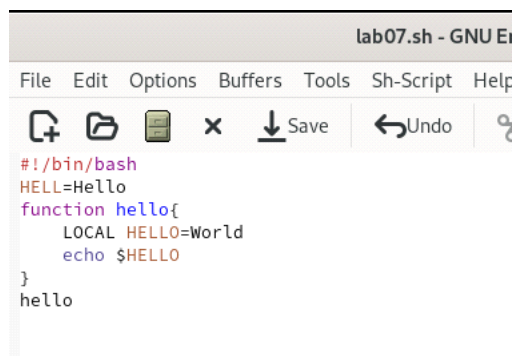
Я написала некоторый текст в этом же файле (lab07.sh). После этого сохранила файл с помощью комбинации Ctrl-x Ctrl-s:



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

Рис. 4: текст в lab07.sh

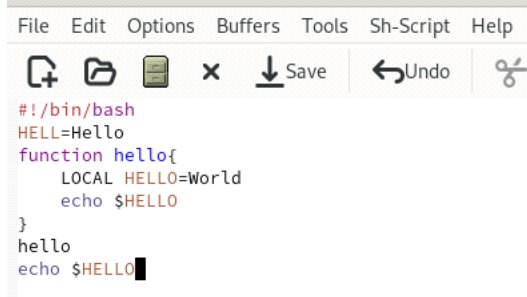
Одной командой вырезала целую строку (C-k):



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

Рис. 5: Вырезание строки

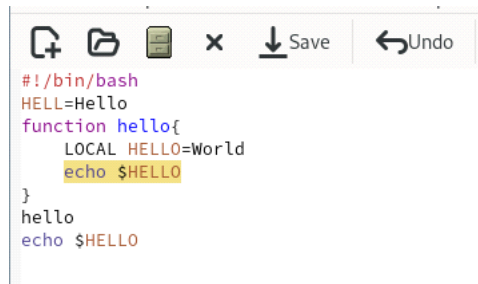
С помощью C-u вставила эту строку в конец файла:



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

Рис. 6: Перемещение строку в конец файла

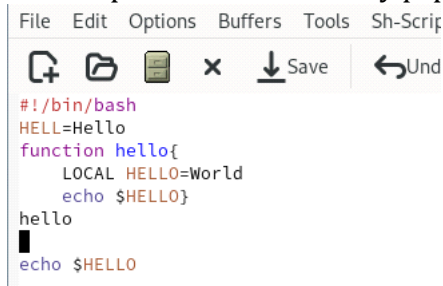
Выделила область текста (C-space):



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

Рис. 7: Выделенный текст

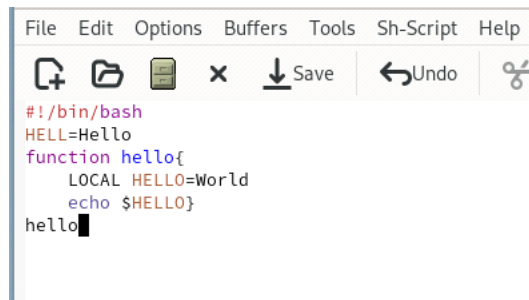
Скопировала область в буфер обмена (М-w) и вставила ее в конец файла:



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

Рис. 8: копирование и вставка

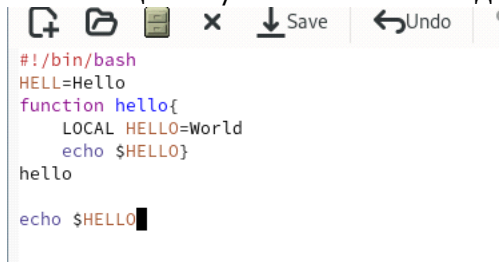
Выделила эту же область и на этот раз вырезала её (С-w):



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

Рис. 9: Вырезанная область

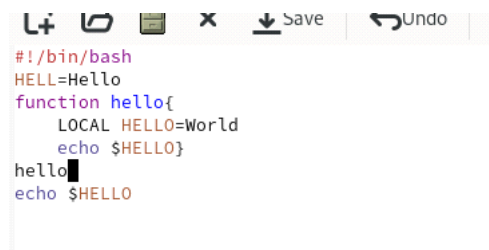
С помощью С-/ отменила последнее действие:



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

Рис. 10: отмена действие

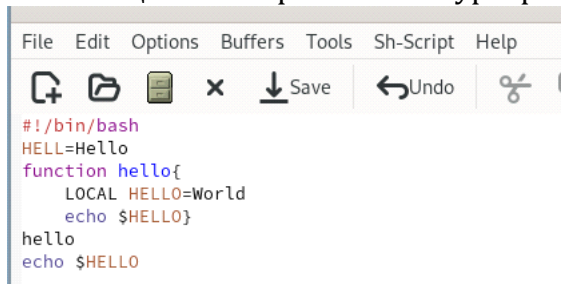
С помощью C-а переместила курсор в начало строки:



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

Рис. 11: Перемещение курсор в начало строки

С помощью C-e переместила курсор в конец строки:



```
#!/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	L9	(Shell-script[sh])	
CRM	Buffer		Size	Mode	File
	* lab07.sh		106	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*		718	Fundamental	
	scratch		145	Lisp Interaction	
***	*Messages*		473	Messages	
***	*Async-native-compile-log*		165	Fundamental	

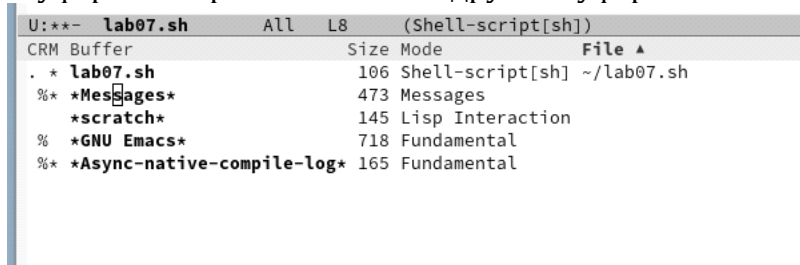
Рис. 13: Перемещение курсор в буфере

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		106	Shell-script[sh]	~/lab07.sh
%	*Messages*		473	Messages	
%	*scratch*		145	Lisp Interaction	
%	*GNU Emacs*		718	Fundamental	
%	*Async-native-compile-log*		165	Fundamental	

Рис. 14: Активные буферы

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



The screenshot shows the Emacs buffer list window. At the top, it says 'U:*** lab07.sh All L8 (Shell-script[sh])'. Below this is a table with columns: CRM, Buffer, Size, Mode, and File. The table lists several buffers: 'lab07.sh' (106 bytes, Shell-script[sh], ~/lab07.sh), '*Messages*' (473 bytes, Messages), '*scratch*' (145 bytes, Lisp Interaction), '*GNU Emacs*' (718 bytes, Fundamental), and '*Async-native-compile-log*' (165 bytes, Fundamental).

CRM	Buffer	Size	Mode	File
.	*lab07.sh	106	Shell-script[sh]	~/lab07.sh
%*	*Messages*	473	Messages	
%*	*scratch*	145	Lisp Interaction	
%	*GNU Emacs*	718	Fundamental	
%*	*Async-native-compile-log*	165	Fundamental	

Рис. 15: список открытых буферов

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

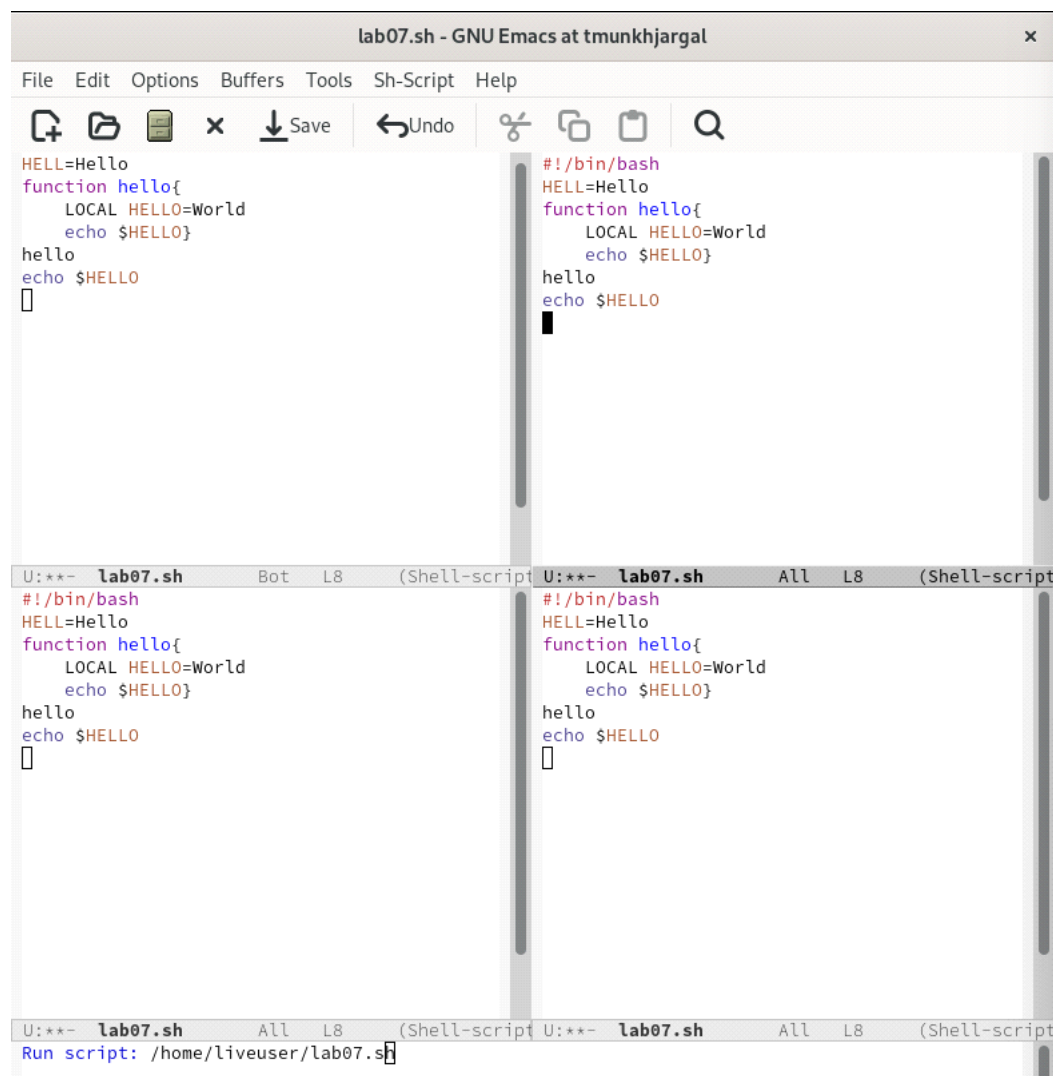


Рис. 16: Закрытие окно

Без вывода списка буферов, я переключилась между буферами:

```
U:***- lab07.sh All L8 (Shell-sc
Run script: /home/liveuser/lab07.sh
```

Рис. 17: Переключение между буферами

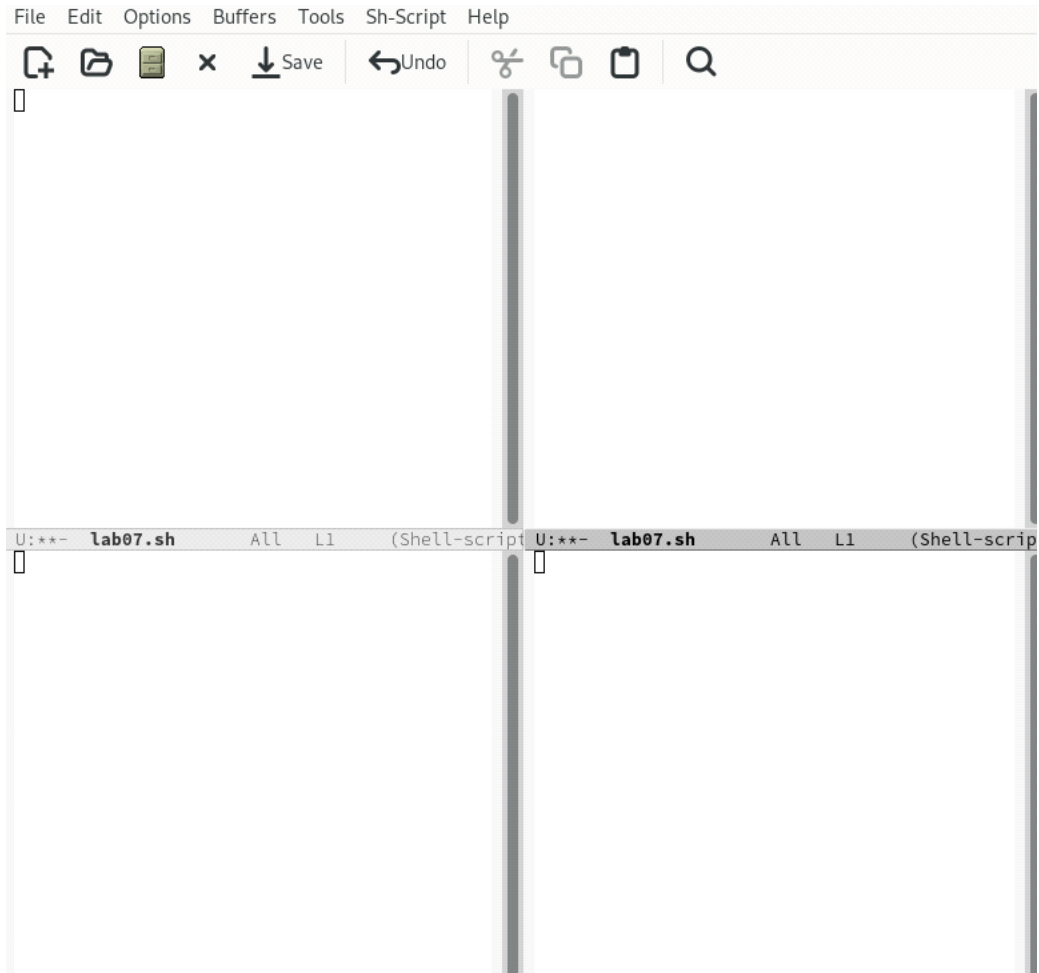


Рис. 18: Новый буфер

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

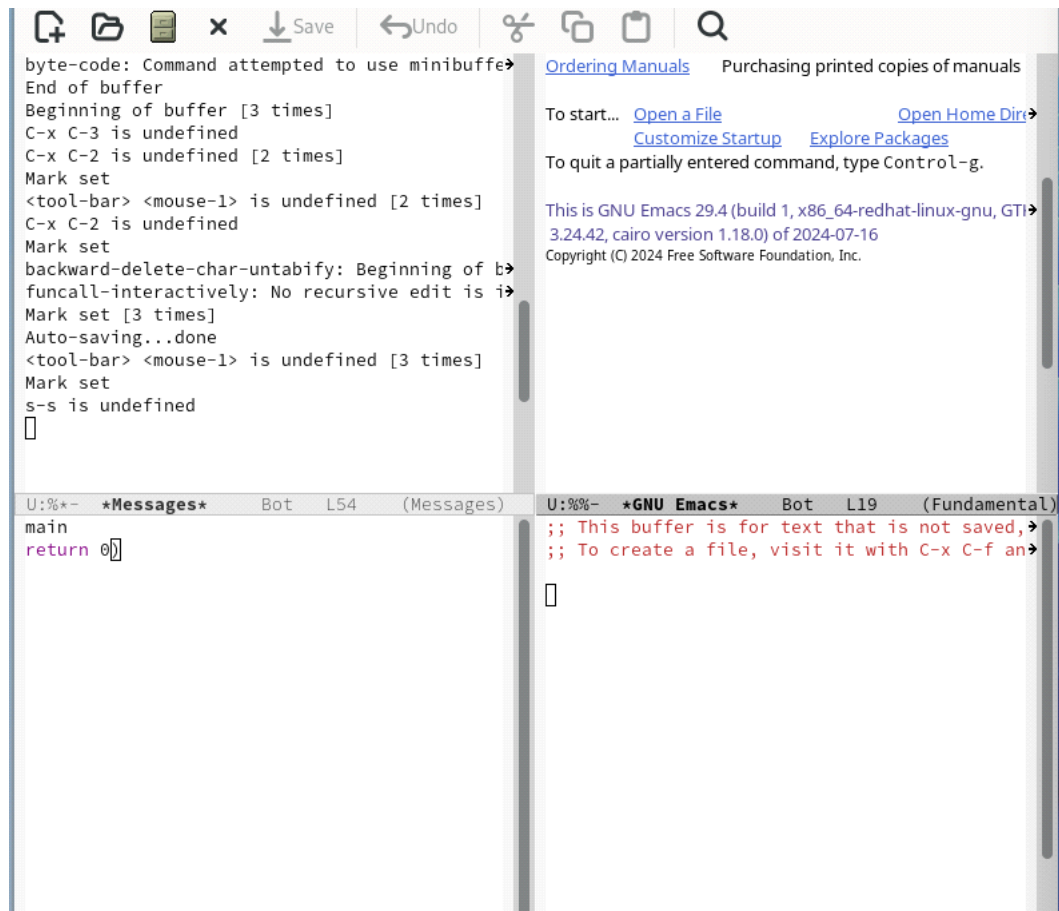


Рис. 19: Фрейм разделённый на 4 окна

В каждом из четырёх созданных окон открыла новый буфер:

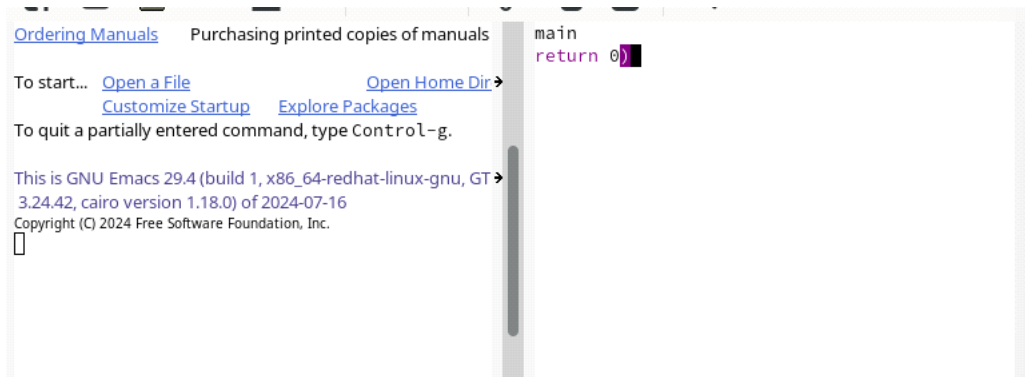
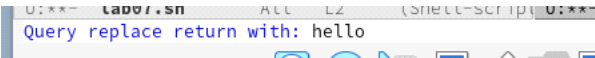


Рис. 20: Новые буферы

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

Рис. 21: Режи



м поиска

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



Рис. 22: Переключение между результатами

• Выводы

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

• Ответы на контрольные вопросы

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