

Отчёт по лабораторной работе №11

Текстовый редактор emacs

Вакутайпа Милдред

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	18
5	Ответы на контрольные вопросы	19

Список иллюстраций

3.1	Установка Emacs	7
3.2	Emacs	7
3.3	Созданный файл	8
3.4	текст в lab07.sh	8
3.5	Вырезание строки	9
3.6	Перемещение строку в конец файла	9
3.7	Выделенный текст	10
3.8	копирование и вставка	10
3.9	ВЫрезанная область	11
3.10	отмена действие	11
3.11	Перемещение курсор в начало строки	12
3.12	Перемещение курсор в конец строки	12
3.13	Перемещение курсор в буфере	12
3.14	Активные буферы	13
3.15	список открытых буферов	13
3.16	Закрытие окно	14
3.17	Переключение между буферами	14
3.18	Новый буфер	14
3.19	Фрейм разделённый на 4 окна	15
3.20	Новые буферы	15
3.21	Режим поиска	15
3.22	Переключение между результатами	16
3.23	Режим поиска	16
3.24	другой режим поиска	16
3.25	Результаты поиска	17

Список таблиц

1 Цель работы

Получить практические навыки работы с редактором Emacs.

2 Задание

1. Выполнить основные команды etacs

3 Выполнение лабораторной работы

Для данной работы, мне надо была установить Emacs:

```
mwakutaipa@mwakutaipa:~$ sudo dnf install emacs
[sudo] пароль для mwakutaipa:
Copr repo for gitflow owned by elegos      [
```

Рис. 3.1: Установка Emacs

Выполнив Emacs в командной строке, я открыла текстовый редактор:

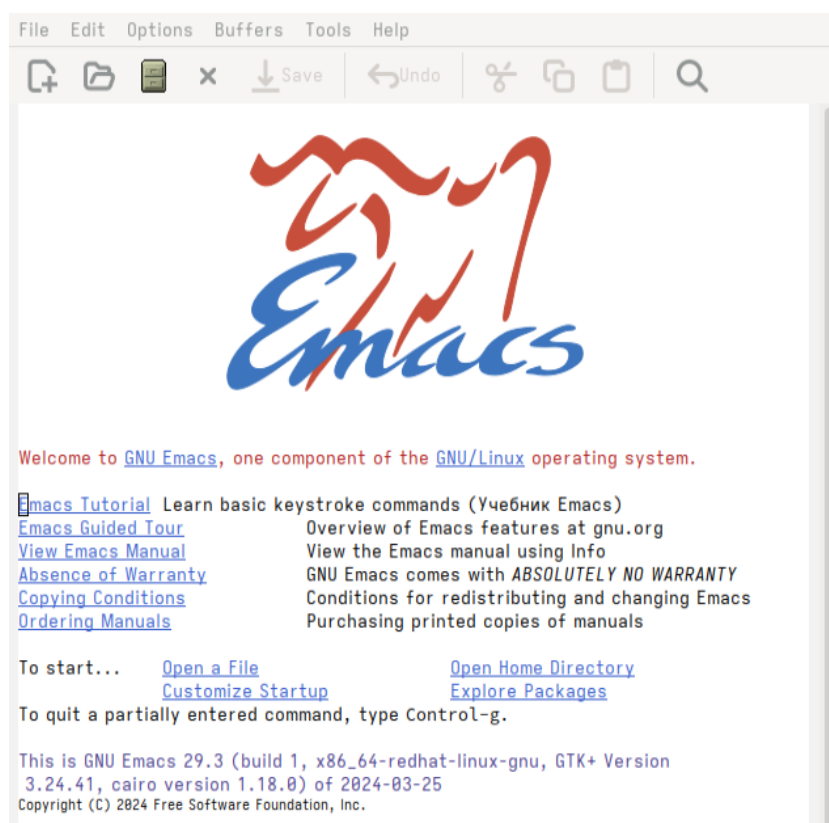


Рис. 3.2: Emacs

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

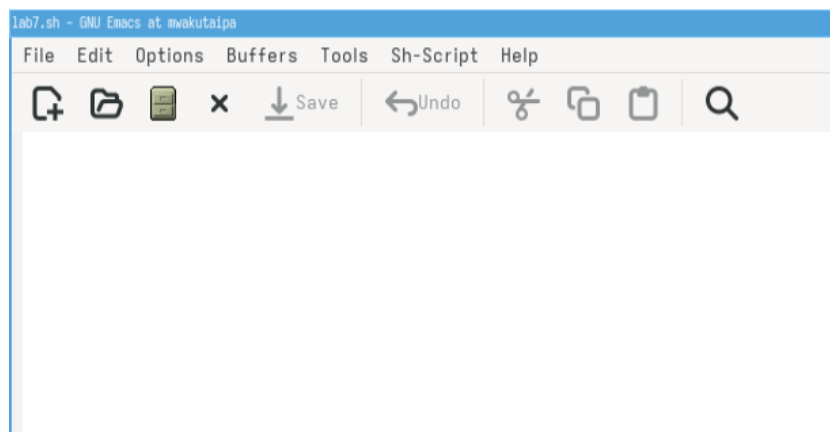


Рис. 3.3: Созданный файл

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

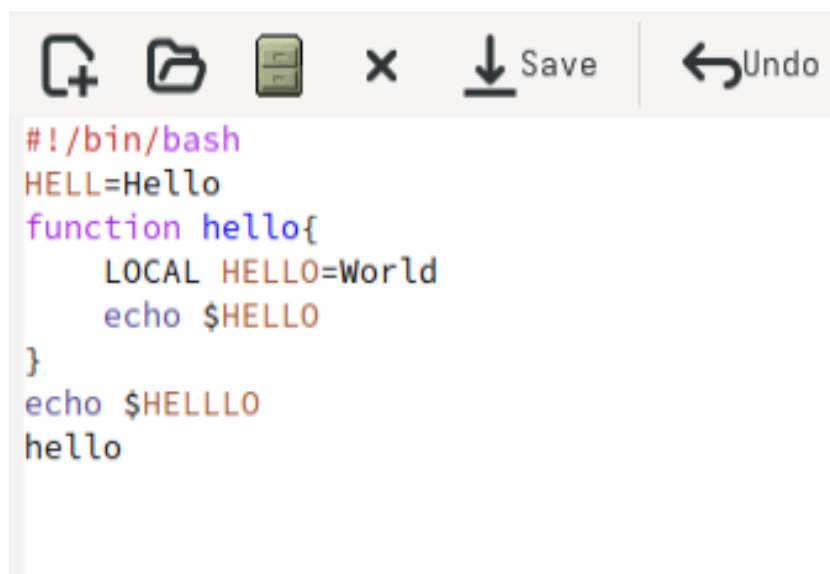


Рис. 3.4: текст в lab07.sh

Одной командой вырезала целую строку (C-k):


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

hello
echo $HELLO
```

Рис. 3.5: Вырезание строки

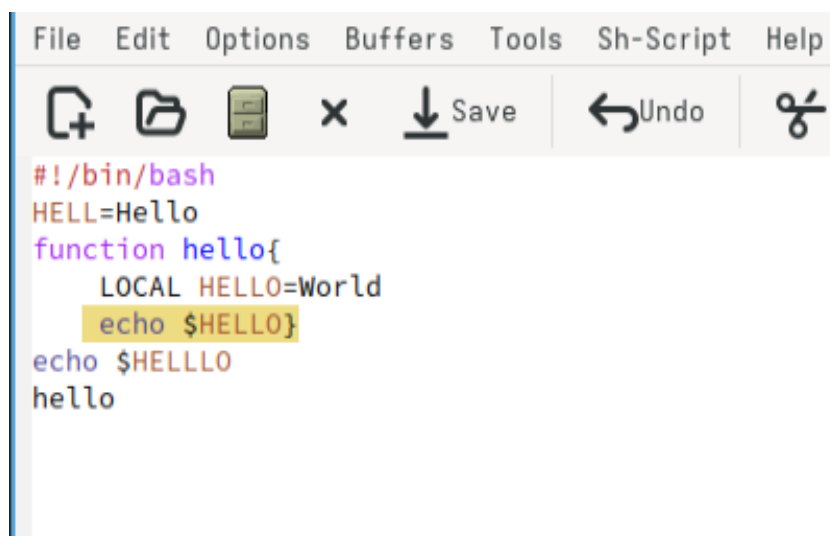
С помощью C-у вставила эту строку в конец файла:

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

hello
echo $HELLO
echo $HELLO
```

Рис. 3.6: Перемещение строку в конец файла

Выделила область текста (C-space):



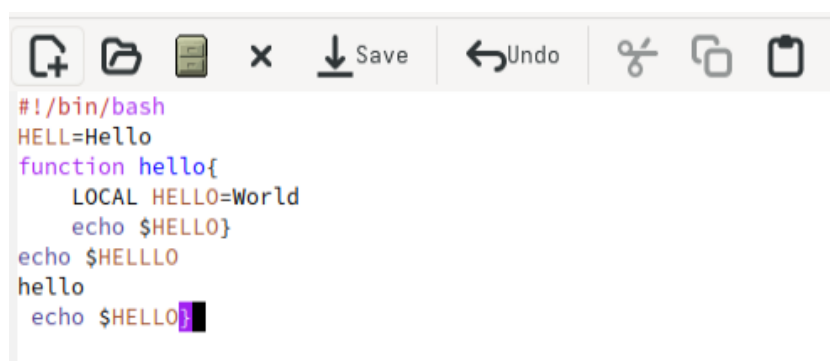
The screenshot shows a text editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for file operations and editing. The script content is as follows:

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

The line `echo $HELLO}` inside the function definition is highlighted in yellow.

Рис. 3.7: Выделенный текст

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



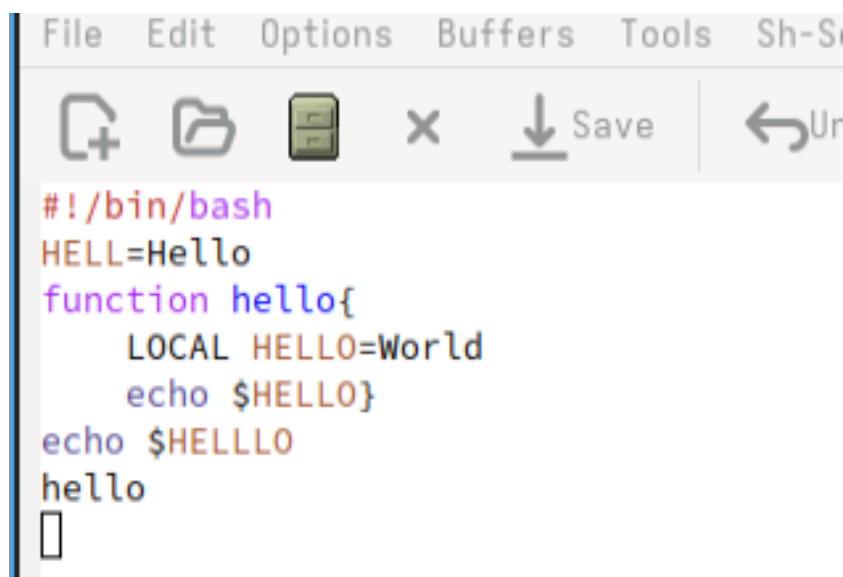
The screenshot shows the same text editor window. The script content is now:

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

The new line `echo $HELLO` at the end of the file is highlighted in purple.

Рис. 3.8: копирование и вставка

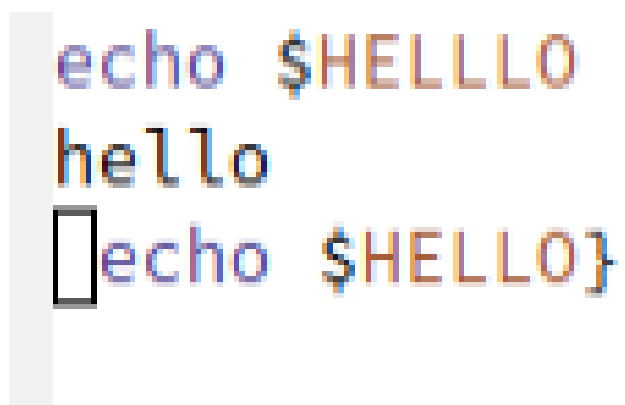
Выделила эту же область и на этот раз вырезала её (C-w):



```
File Edit Options Buffers Tools Sh-S
+  Save  Ur
#!/bin/bash
HELL=Hello
function hello{
    LOCAL HELLO=World
    echo $HELLO}
echo $HELLLO
hello
█
```

Рис. 3.9: Вырезанная область

С помощью C-/ отменила последнее действие:



```
echo $HELLLO
hello
█echo $HELLO}
```

Рис. 3.10: отмена действие

С помощью C-a переместила курсор в начало строки:

```
echo $HELLLO
hello
echo $HELLO}
```

Рис. 3.11: Перемещение курсор в начало строки

С помощью С-е переместила курсор в конец строки:

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

Рис. 3.12: Перемещение курсор в конец строки

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

```
U:***- lab7.sh All L4 (Shell-script[sh])
Visit tags table (default TAGS): ~/
```

Рис. 3.13: Перемещение курсор в буфере

Выводила список активных буферов на экран с помощью С-х С-b:

U:***	lab7.sh	All	L4	(Shell-script[sh])
CRM	Buffer	Size	Mode	File
□	* lab7.sh	111	Shell-script[sh]	~/lab7.sh
%	*GNU Emacs*	734	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	869	Messages	
%*	*Async-native-compile-...	165	Fundamental	

Рис. 3.14: Активные буферы

С помощью С-х о переместилась во вновь открытое окно со списком открытых буферов и переключилась на другой буфер:

U:***	lab7.sh	All	L9	(Shell-script[bash])
CRM	Buffer	Size	Mode	File
□	lab7.sh	111	Shell-script[ba...	~/lab7.sh
%	*GNU Emacs*	734	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	552	Messages	
%*	*Async-native-compile-...	165	Fundamental	

U:%%-	*Buffer List*	All	L1	(Buffer Menu)
-------	---------------	-----	----	---------------

Рис. 3.15: список открытых буферов

С помощью С-х 0 закрыла окно со списком открытых буферов:

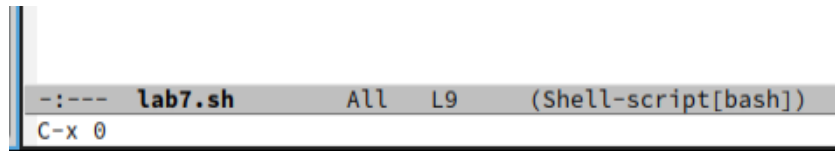


Рис. 3.16: Заккрытие окно

Без вывода списка буферов, я переключилась между буферами:

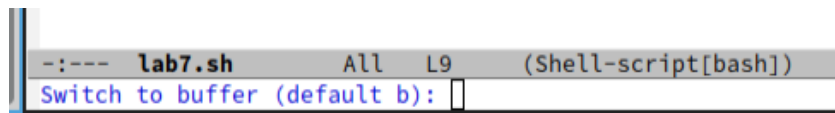


Рис. 3.17: Переключение между буферами

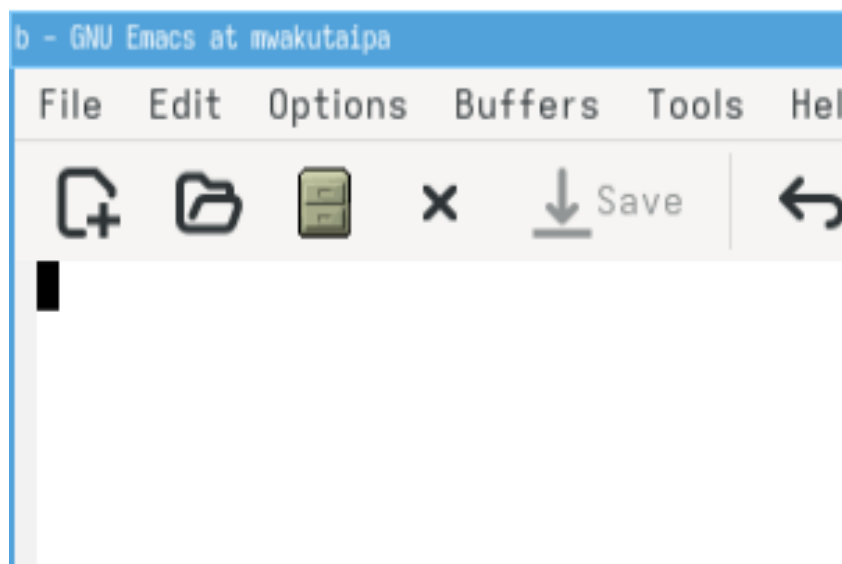


Рис. 3.18: Новый буфер

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):

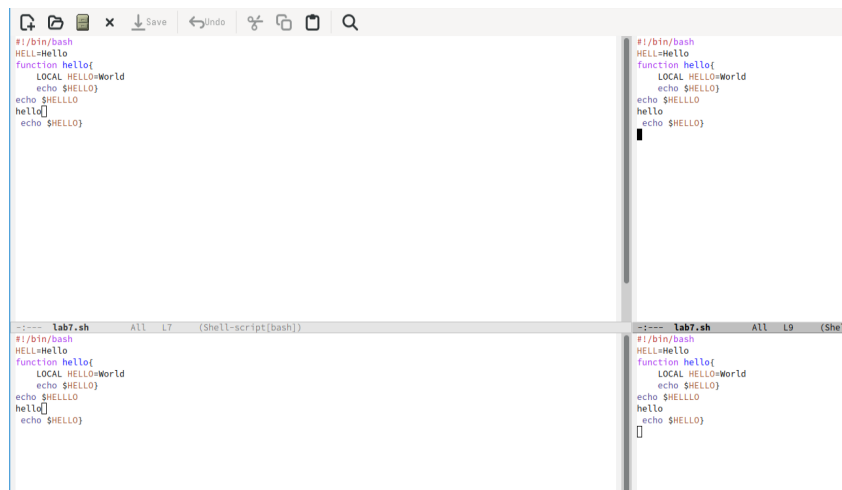


Рис. 3.19: Фрейм разделённый на 4 окна

В каждом из четырёх созданных окон открыла новый буфер:

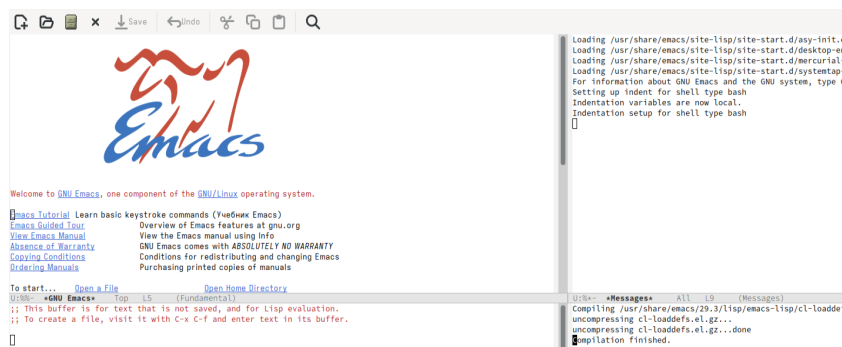


Рис. 3.20: Новые буферы

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

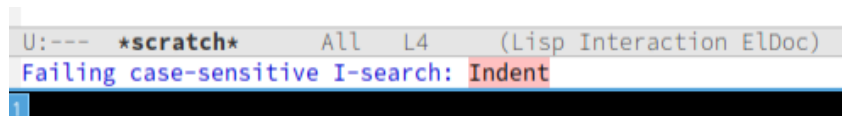


Рис. 3.21: Режим поиска

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

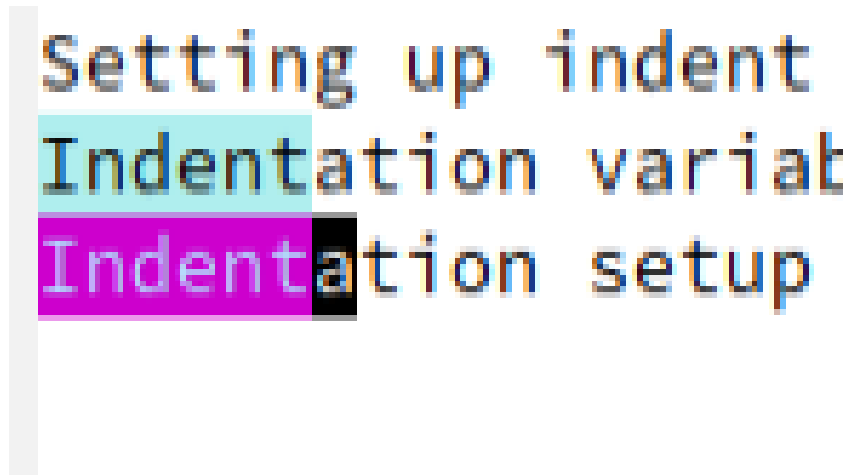


Рис. 3.22: Переключение между результатами

Перешла в режим поиска и замены (M-%), искала слово World, нажала Enter, и заменила на Planet:

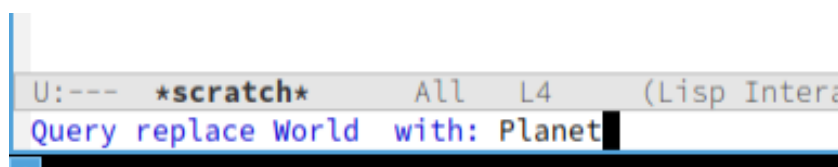


Рис. 3.23: Режим поиска

Нажав M-s o, я использовала другой режим поиска. Он отличается от предыдущего тем, что выводит результаты поиска в новом окне:

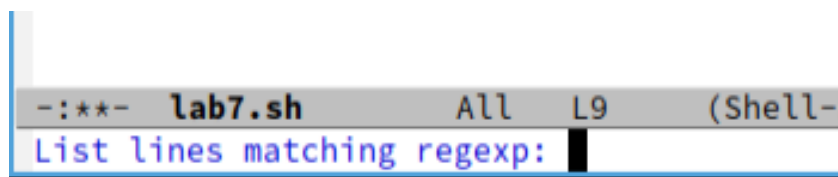
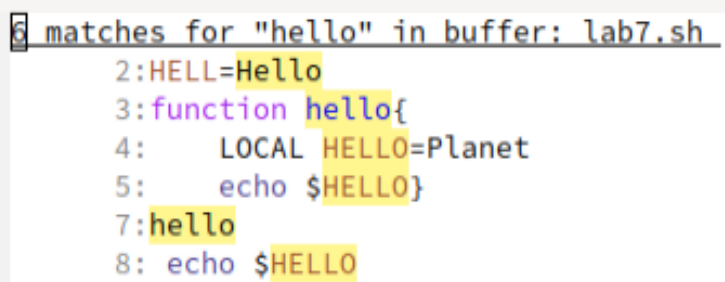


Рис. 3.24: другой режим поиска



6 matches for "hello" in buffer: lab7.sh

```
2:HELL=Hello
3:function hello{
4:    LOCAL HELLO=Planet
5:    echo $HELLO}
7:hello
8: echo $HELLO
```

Рис. 3.25: Результаты поиска

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 работа 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.