

Лабораторная работа № 11

Простейший вариант

Лупупа Чилеше

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Контрольные вопросы	17
6	Выводы	21

Список иллюстраций

4.1	emacs	8
4.2	emacs	9
4.3	lab07.sh	10
4.4	C-space	10
4.5	M-w	11
4.6	C-a	11
4.7	C-e	11
4.8	M-<	12
4.9	M->	12
4.10	C-x C-b	13
4.11	переключение на другой буфер	13
4.12	C-x 3, C-x 2	14
4.13	Keep Calm And Code	15
4.14	C-s	16
4.15	M-%	16

Список таблиц

1 Цель работы

Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

2 Задание

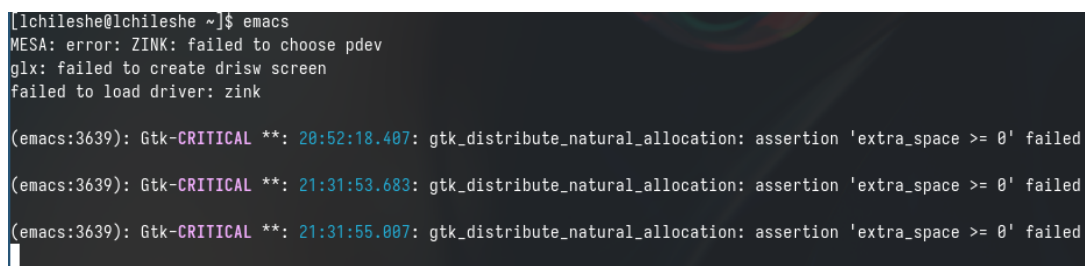
Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Теоретическое введение

Emacs — это легко расширяемый и настраиваемый текстовый редактор и вычислительная среда, которая была краеугольным камнем движения за свободное программное обеспечение с момента его создания. Он был создан Ричардом Столлманом в 1976 году как часть проекта GNU с целью предоставить бесплатную альтернативу проприетарным редакторам того времени. Emacs построен на основе Emacs Lisp, диалекта языка программирования Lisp, который позволяет пользователям расширять и настраивать редактор в соответствии со своими потребностями. Это делает Emacs не просто редактором, а мощной вычислительной средой, в которой пользователи могут писать, отлаживать и запускать код непосредственно в редакторе.

4 Выполнение лабораторной работы

1. Открыть emacs.



```
[lchileshe@lchileshe ~]$ emacs
MESA: error: ZINK: failed to choose pdev
glx: failed to create driw screen
failed to load driver: zink

(emacs:3639): Gtk-CRITICAL **: 20:52:18.407: gtk_distribute_natural_allocation: assertion 'extra_space >= 0' failed
(emacs:3639): Gtk-CRITICAL **: 21:31:53.683: gtk_distribute_natural_allocation: assertion 'extra_space >= 0' failed
(emacs:3639): Gtk-CRITICAL **: 21:31:55.007: gtk_distribute_natural_allocation: assertion 'extra_space >= 0' failed
```

Рис. 4.1: emacs

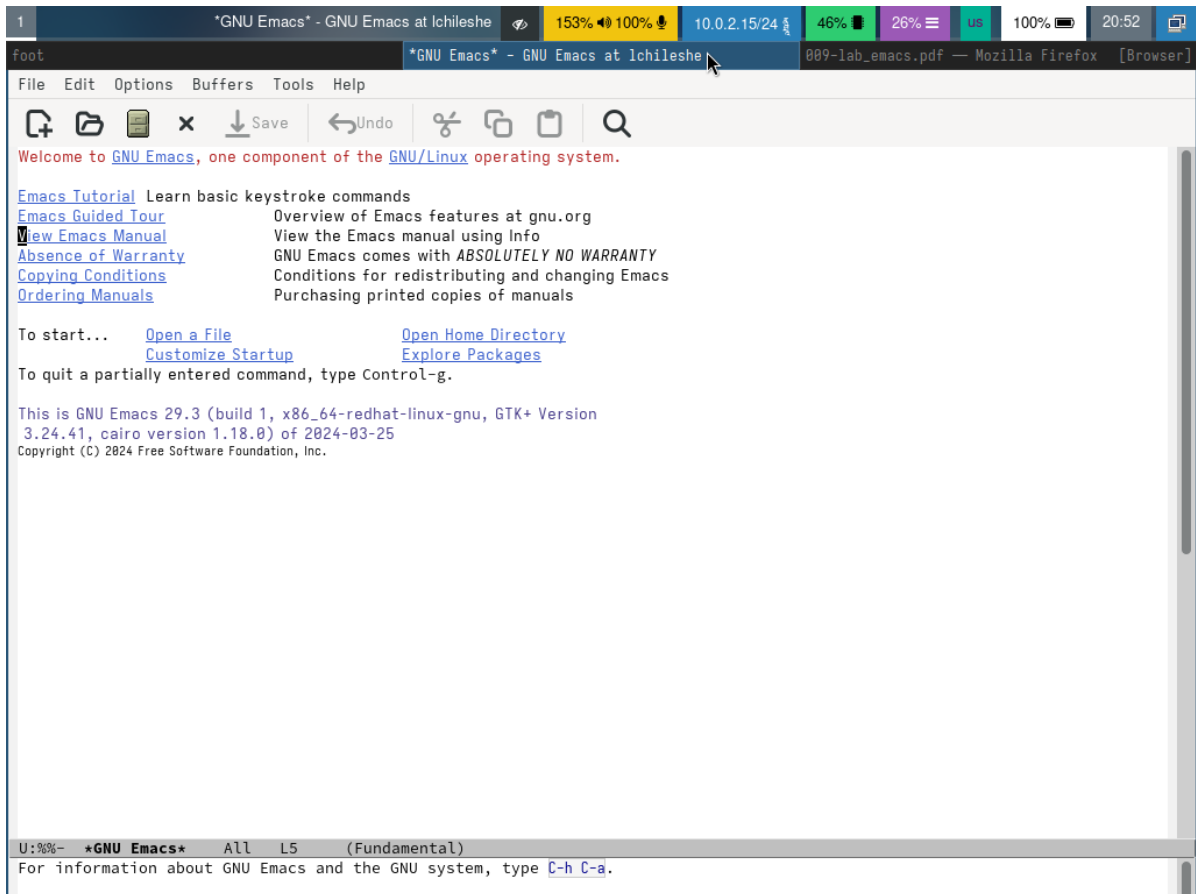


Рис. 4.2: emacs

2. Создать файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).

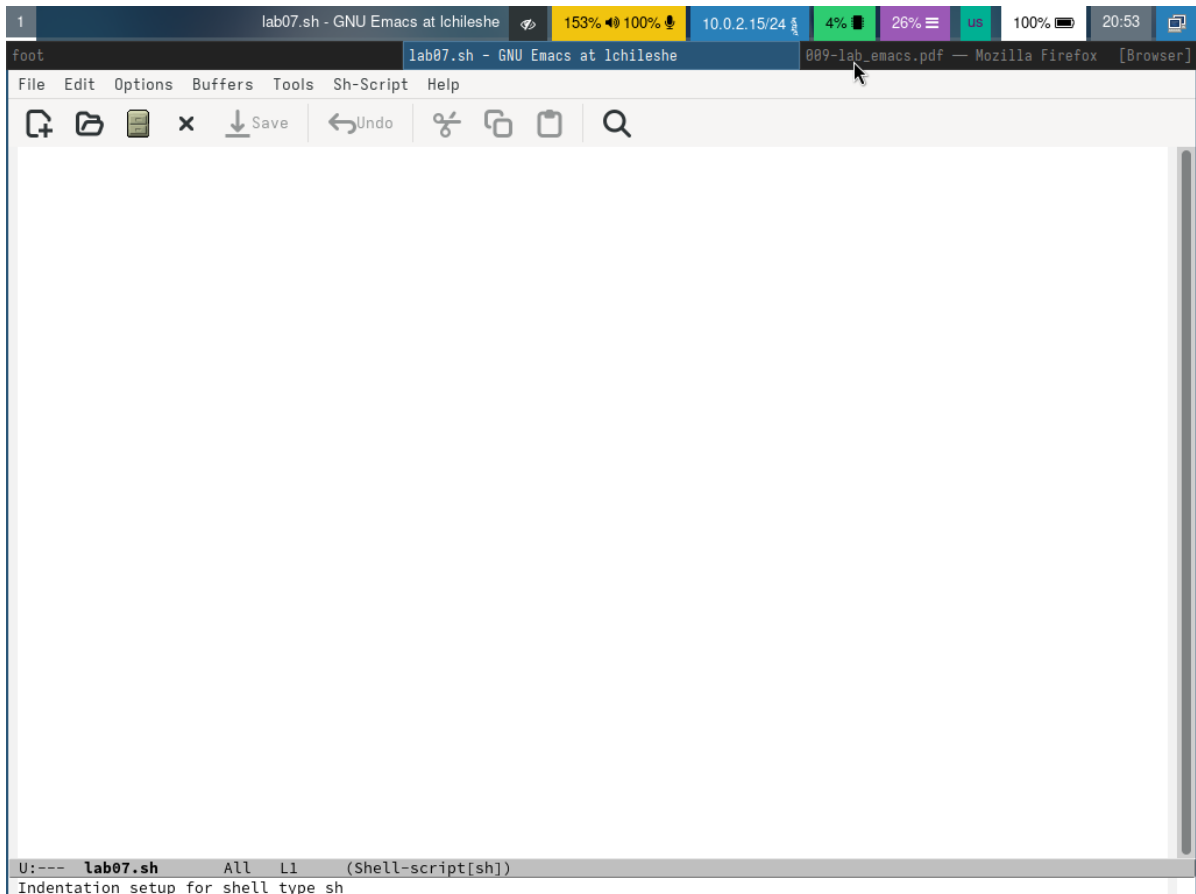


Рис. 4.3: lab07.sh

Наберите текст:

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

2. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.

- Выделить область текста (C-space).

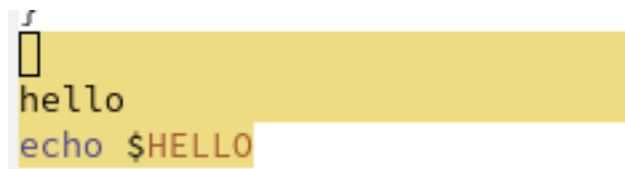


Рис. 4.4: C-space

- Скопировать область в буфер обмена (M-w).
- Вставить область в конец файла

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

hello
echo $HELLO

hello
echo $HELLO
```

Рис. 4.5: M-w

3. Научитесь использовать команды по перемещению курсора.

- Переместите курсор в начало строки (C-a).

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

Рис. 4.6: C-a

- Переместите курсор в конец строки (C-e)

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

Рис. 4.7: C-e

- Переместите курсор в начало буфера (M-<)

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

hello
echo $HELLO

hello
echo $HELLO
```

Рис. 4.8: M-<

- Переместите курсор в конец буфера (M->)

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

hello
echo $HELLO

hello
echo $HELLO
```

Рис. 4.9: M->

4. Управление буферами.

- Вывести список активных буферов на экран (C-x C-b)

U:*** lab07.sh All L13 (Shell-script[sh])					
CRM	Buffer		Size	Mode	File
1	* lab07.sh		110	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*		714	Fundamental	
	scratch		145	Lisp Interaction	
%*	*Messages*		1821	Messages	
%*	*Async-native-compile-log*		165	Fundamental	

Рис. 4.10: C-x C-b

- Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер.



Рис. 4.11: переключение на другой буфер

5. Управление окнами.

- Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2) (см. рис. 9.1).

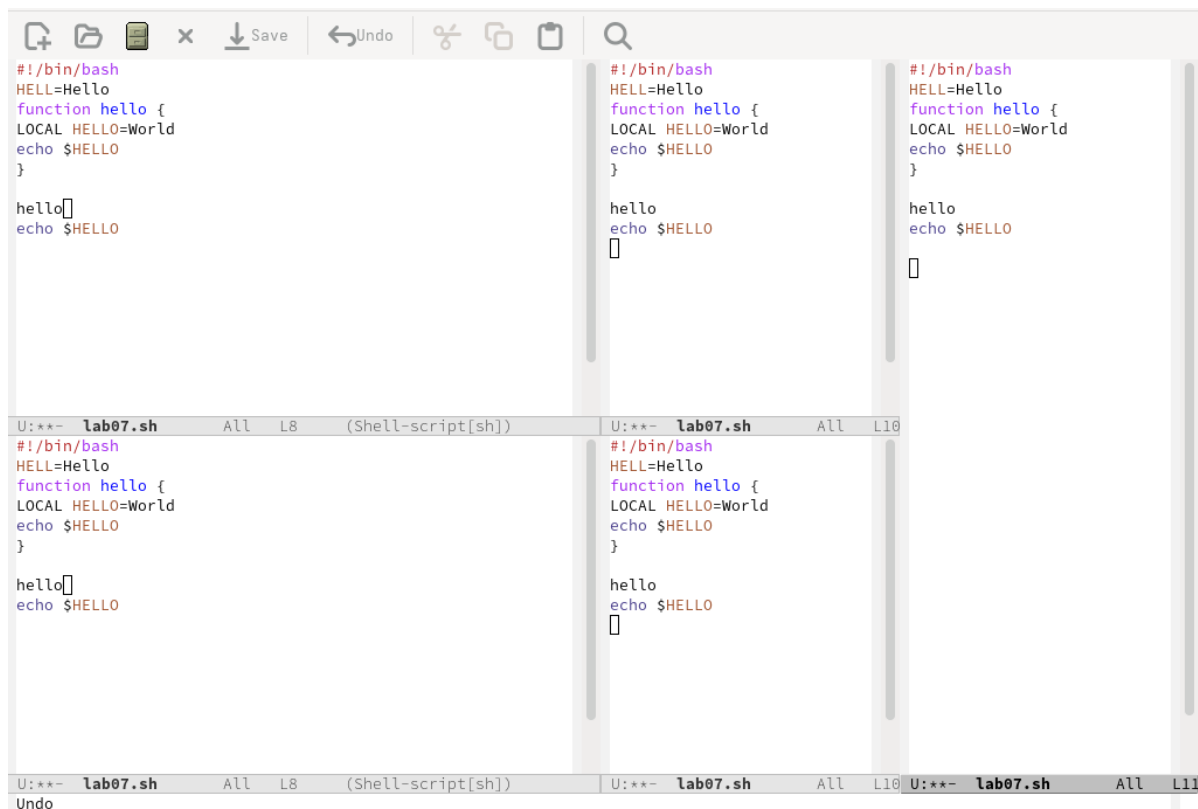


Рис. 4.12: C-x 3, C-x 2

- В каждом из четырёх созданных окон откройте новый буфер (файл) и введите несколько строк текста.

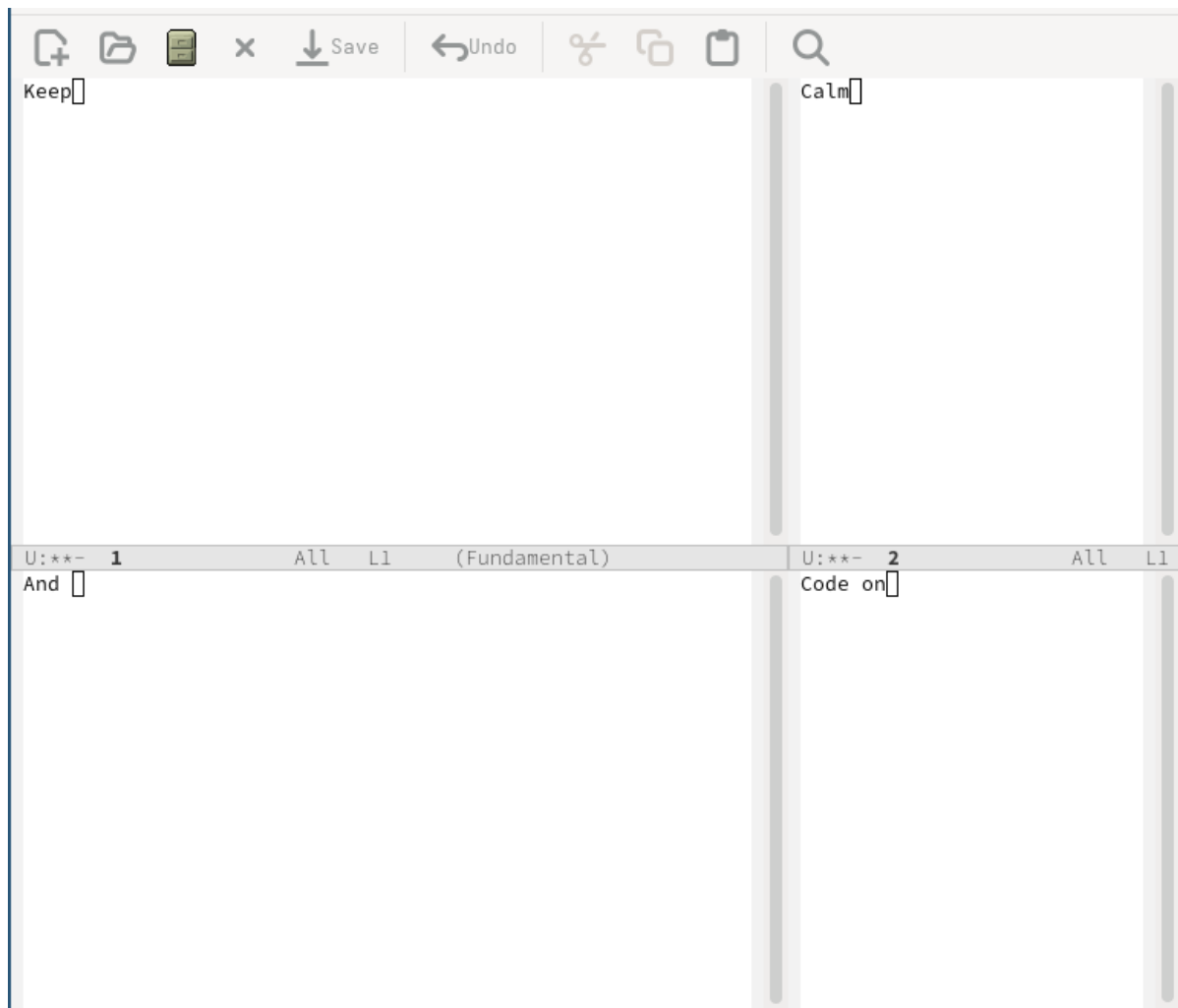


Рис. 4.13: Keep Calm And Code

6. Режим поиска

- Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте.



Рис. 4.14: C-s

Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter , затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены.

```
real musicKreal musicereal musicereal musicpreal music  
real music  
real music  
real music□
```

Рис. 4.15: M-%

5 Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

- Emacs - это продвинутый, расширяемый, настраиваемый и самодокументируемый редактор, который существует уже почти 40 лет. Он отличается от современных программных редакторов своей легкостью потребления системных ресурсов, благодаря чему его можно считать относительно легким инструментом для выполнения задач, которые он выполняет. Несмотря на то, что Emacs был раньше воспринимаем как необычно большой инструмент для своих возможностей, современные редакторы, использующие веб-браузерные движки и серверы на JavaScript в качестве своих бэкендов, делают его использование еще более непримечательным. Однако, Emacs делает гораздо больше, чем просто редактирование текста

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

- **Широкий спектр команд и горячих клавиш:** Emacs обладает сотнями команд и горячих клавиш, что делает его очень мощным инструментом, но также увеличивает сложность его освоения. Новички могут испытывать трудности с запоминанием всех этих команд и пониманием, как и когда их использовать
- **Настройка и конфигурация:** Emacs предлагает множество настроек, что позволяет пользователям адаптировать редактор под свои нужды. Однако,

это также означает, что новичкам может быть сложно понять, как и где настраивать различные параметры, чтобы получить желаемый результат

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

- В терминологии Emacs, буфер - это основная единица редактирования, представляющая собой кусок редактируемого текста. Буфер может быть связан с файлом, в котором он хранит данные, или быть создан в результате выполнения команд, содержащими вывод команды, список для выбора вариантов и другую информацию. В любой момент времени вы можете редактировать только один выбранный буфер, но при использовании нескольких окон видимыми могут быть несколько буферов
- Окно в Emacs - это визуальное представление буфера. Окно позволяет пользователю видеть и взаимодействовать с содержимым буфера. В Emacs можно иметь несколько окон, каждое из которых может отображать разные буферы. Это позволяет пользователям работать с несколькими файлами или частями файла одновременно, улучшая продуктивность и удобство работы

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

- В Emacs, вы можете открыть больше 10 буферов, но в одном окне отображается только один буфер за раз. Окно в Emacs - это визуальное представление буфера, позволяющее пользователю видеть и взаимодействовать с содержимым буфера. Если у вас есть несколько окон, каждое из них может отображать разные буферы, но в каждом окне видим только один буфер

5. Какие буферы создаются по умолчанию при запуске emacs?

- Буфер scratch: Этот буфер предназначен для временных заметок и обычно отображается при старте Emacs, если он существует и пуст

- уферы, указанные через аргументы командной строки: Если при запуске Emacs указаны файлы через аргументы командной строки, то для каждого из этих файлов создается новый буфер
 - Буфер, определенный через initial-buffer-choice: Этот механизм позволяет настроить, какой буфер будет отображаться при старте Emacs
6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?
- Чтобы ввести комбинацию клавиш C-c | в Emacs, вы должны нажать клавишу Control (Ctrl) и одновременно нажать клавишу c, а затем отпустить обе клавиши и нажать клавишу |.

Чтобы ввести комбинацию клавиш C-c C-| (где C-| обозначает нажатие клавиши Control (Ctrl) и клавиши | одновременно), вы должны нажать клавишу Control (Ctrl) и одновременно нажать клавишу c, а затем отпустить обе клавиши и снова нажать клавишу Control (Ctrl) и клавишу | одновременно.

7. Как поделить текущее окно на две части?

- В Linux, в зависимости от используемого оконного менеджера, могут быть разные комбинации клавиш. Например, в GNOME вы можете использовать Super + Up Arrow или Super + Down Arrow для разделения окна по горизонтали или вертикали.

8. В каком файле хранятся настройки редактора emacs?

- астройки редактора Emacs хранятся в файле конфигурации, который может находиться в нескольких местах в зависимости от вашей операционной системы и предпочтений. Вот основные места, где Emacs ищет файл конфигурации:

~/.emacs или ~/.emacs.el - старые местоположения файла конфигурации (не рекомендуется). ~/.emacs.d/init.el - основной файл конфигурации

в папке конфигурации Emacs (рекомендуется на macOS и Windows).
~/.config/emacs/init.el - следует руководству Linux для рабочего окружения (рекомендуется на Linux).

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

- Клавиша Fn (функциональная клавиша) на клавиатуре обычно используется для доступа к дополнительным функциям или комбинациям клавиш, которые не могут быть достигнуты с помощью стандартных клавиш. В контексте Emacs, клавиша Fn может не работать так, как ожидается, особенно на macOS, где она может конфликтовать с функциональностью Emacs.

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

- Выбор между редакторами vi и Emacs во многом зависит от личных предпочтений, стиля работы и требований к функциональности. Оба редактора имеют свои уникальные особенности и преимущества, которые могут сделать один из них более предпочтительным в определенных ситуациях

Emacs:

Расширяемость и настройка: Emacs предлагает широкие возможности для настройки и расширения

Поддержка языков и скриптов: Emacs полностью поддерживает все системы письма, совместимые с Unicode, и позволяет свободно смешивать различные скрипты

Vi (или Vim):

Простота и скорость: Vi (и его расширенная версия Vim) известны своей простотой и скоростью

Поддержка языков и скриптов: Vi имеет базовую поддержку языков, отличных от английского

6 Выводы

Погрузившись в мир Emacs, я оценил его мощь и гибкость как текстового редактора и вычислительной среды. Способность Emacs адаптироваться к моим потребностям, движимая мышечной памятью и глубоким пониманием его возможностей, изменила мои рабочие процессы написания и программирования. Широкие возможности настройки в сочетании с эффективностью операций с клавиатурой сделали Emacs незаменимым инструментом в моем арсенале разработчиков. Будь то написание книги, редактирование документов или просто навигация по файлам, Emacs оказался очень эффективным и приятным в использовании инструментом. Путь изучения Emacs был одновременно трудным и плодотворным, и я с нетерпением жду продолжения изучения и освоения этого мощного редактора.