

Отчёт по лабораторной работе №9

Текстовый редактор emacs

Мокочунина Влада сергеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	27
5	Выводы	29

Список иллюстраций

3.1	emacs	7
3.2	файл	8
3.3	текст	8
3.4	сохранение	9
3.5	строка	10
3.6	вставка	11
3.7	выделение	12
3.8	копирование	13
3.9	вставка	14
3.10	вырезание	15
3.11	Отмена	16
3.12	перемещение	17
3.13	перемещение	18
3.14	перемещение	19
3.15	перемещение	20
3.16	список	21
3.17	переключение	22
3.18	закрытие	23
3.19	переключение	23
3.20	деление	24
3.21	текст	25
3.22	поиск	26
3.23	переключение	26

Список таблиц

1 Цель работы

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

2 Задание

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

3 Выполнение лабораторной работы

1. Открыла emacs. (рис. [3.1]).

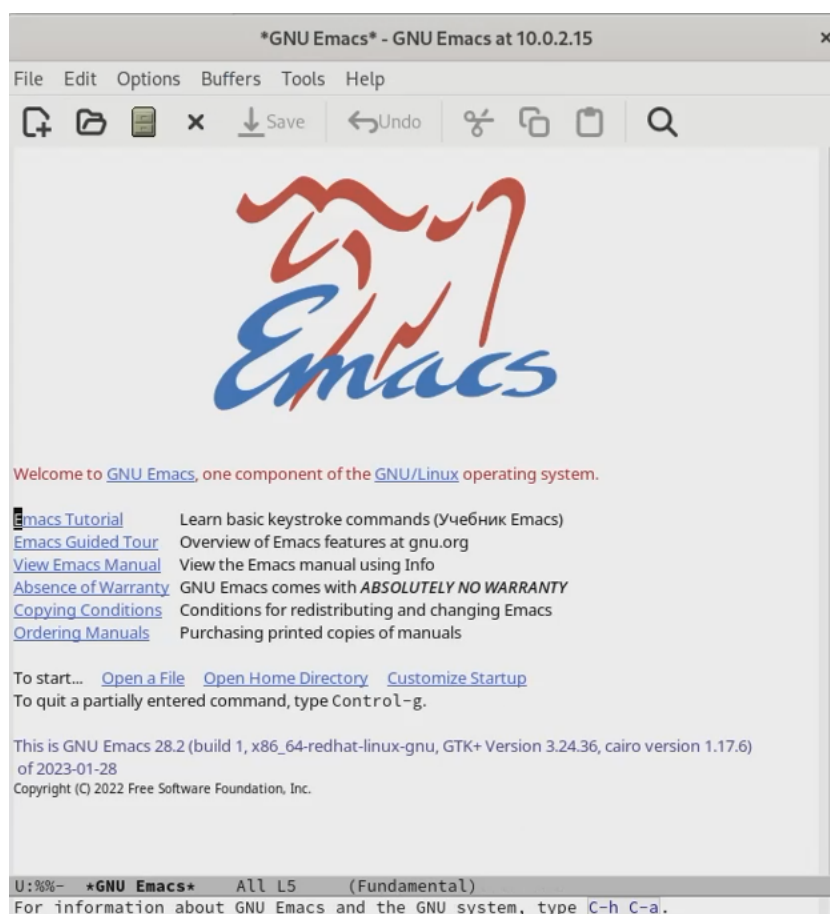


Рис. 3.1: emacs

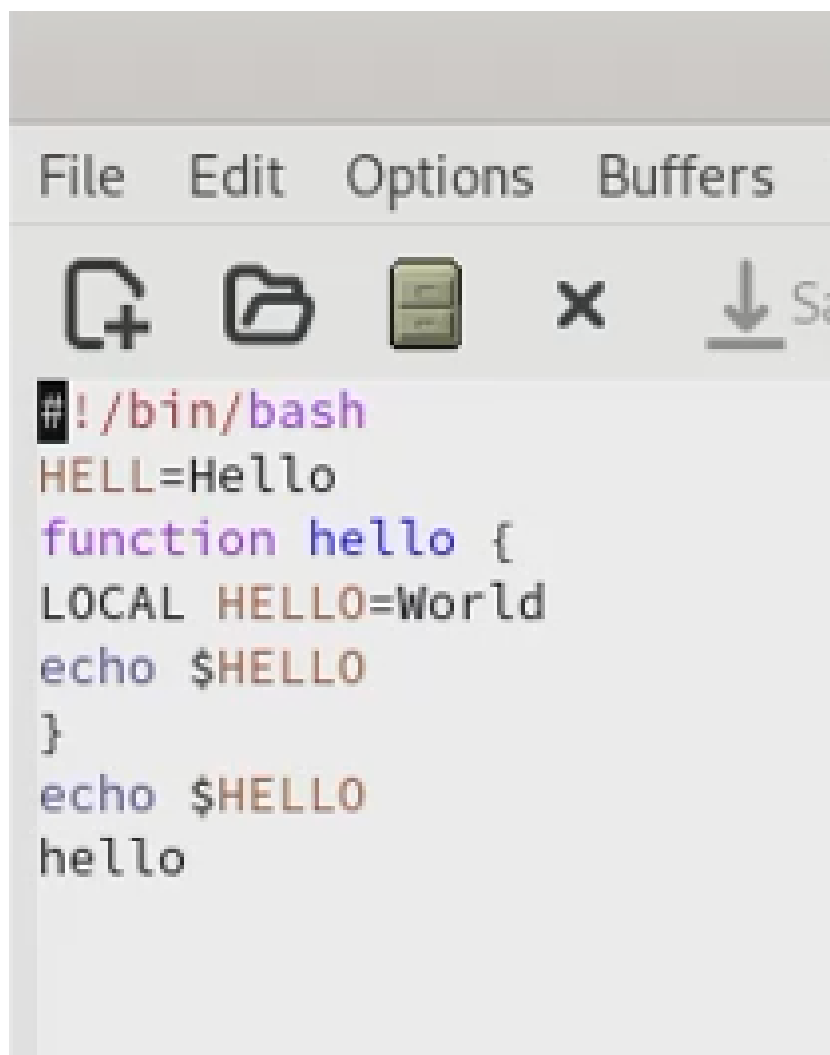
2. Создала файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).



```
Find file: ~/lab07.sh
```

Рис. 3.2: файл

3. Набрала текст



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

Рис. 3.3: текст

4. Сохранила файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).



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

Рис. 3.4: сохранение

5. Вырезала одной командой целую строку (C-k).

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

Рис. 3.5: строка

6. Вставила эту строку в конец файла (С-у).

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

hello
echo $HELLO
```

Рис. 3.6: вставка

7. Выделила область текста (C-space).

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

hello
echo $HELLO
LOCAL HELLO=World
echo $HELLO
```

Рис. 3.7: выделение

8. Скопировала область в буфер обмена (M-w).

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

hello
echo $HELLO
```

Рис. 3.8: копирование

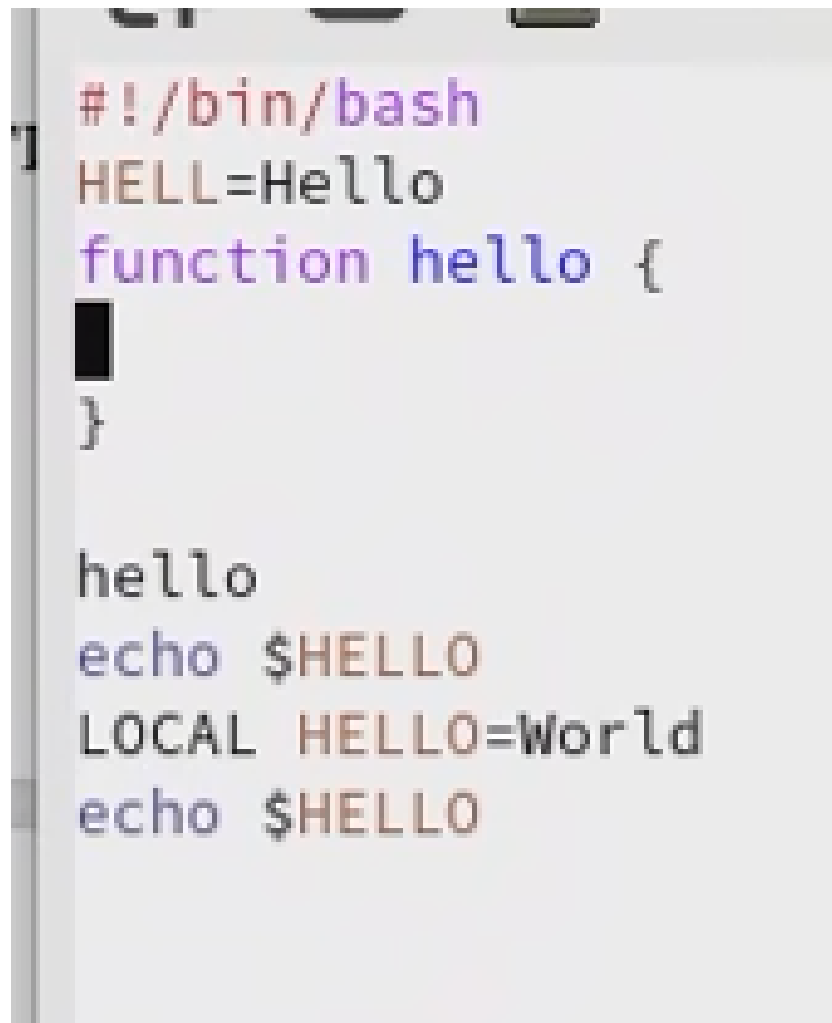
9. Вставила область в конец файла.

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

hello
echo $HELLO
LOCAL HELLO=World
echo $HELLO
```

Рис. 3.9: вставка

10. Вновь выделила эту область и на этот раз вырезала её (C-w).



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

hello
```

Рис. 3.10: вырезание

11. Отменила последнее действие (C-/).

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

hello
echo $HELLO
LOCAL HELLO=World
echo $HELLO
```

Рис. 3.11: Отмена

12. Переместила курсор в начало строки (C-a).


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

hello
echo $HELLO
LOCAL HELLO=World
echo $HELLO
```

Рис. 3.12: перемещение

13. Переместила курсор в конец строки (C-e).

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

hello
echo $HELLO
LOCAL HELLO=World
echo $HELLO
```

Рис. 3.13: перемещение

14. Переместила курсор в начало буфера (M-<).

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

hello
echo $HELLO
LOCAL HELLO=World
echo $HELLO
```

Рис. 3.14: перемещение

15. Переместила курсор в конец буфера (M->).

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

hello
echo $HELLO
LOCAL HELLO=World
```

Рис. 3.15: перемещение

16. Вывела список активных буферов на экран (C-x C-b).

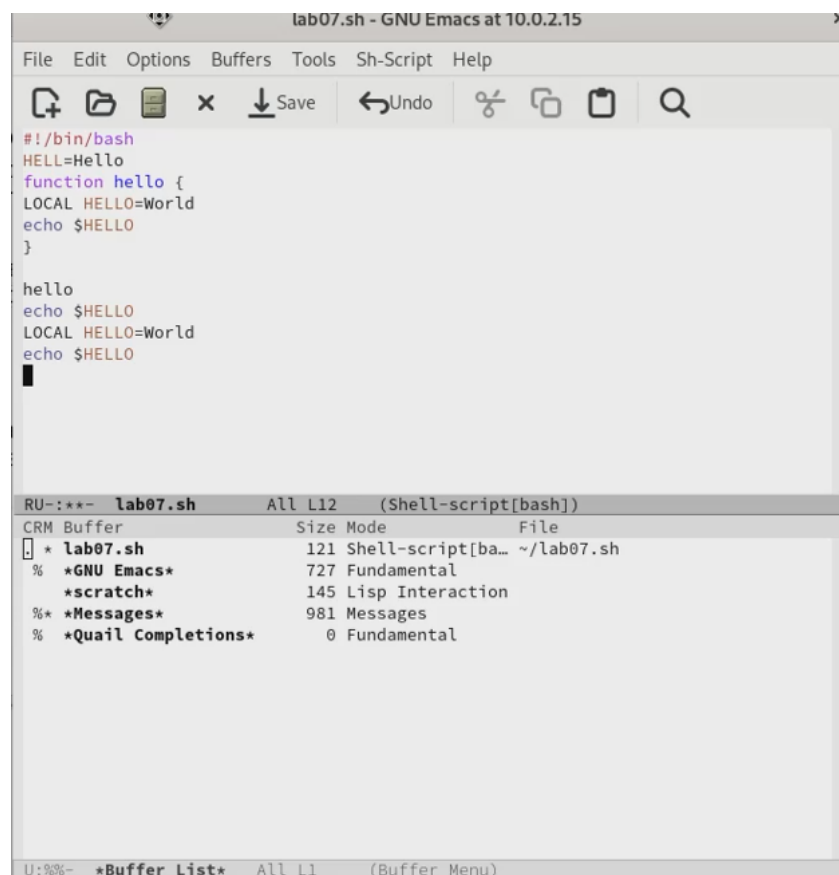


Рис. 3.16: список

17. Переместилась во вновь открытое окно (С-х) о со списком открытых буферов и переключилась на другой буфер.

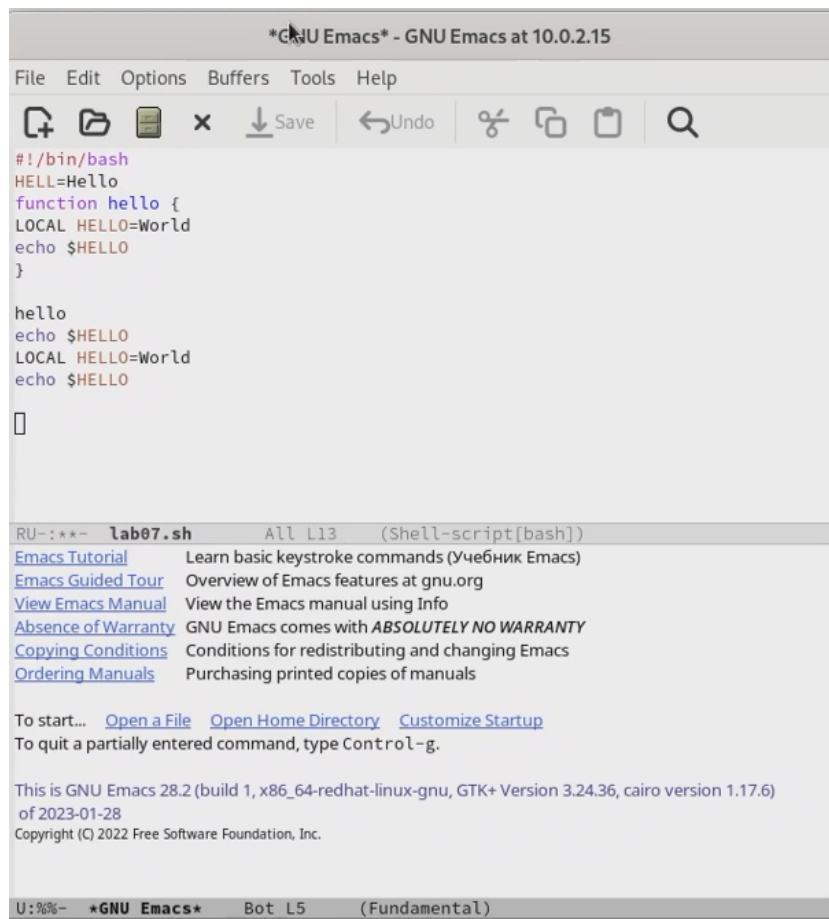


Рис. 3.17: переключение

18. Закрывает это окно (C-x 0).

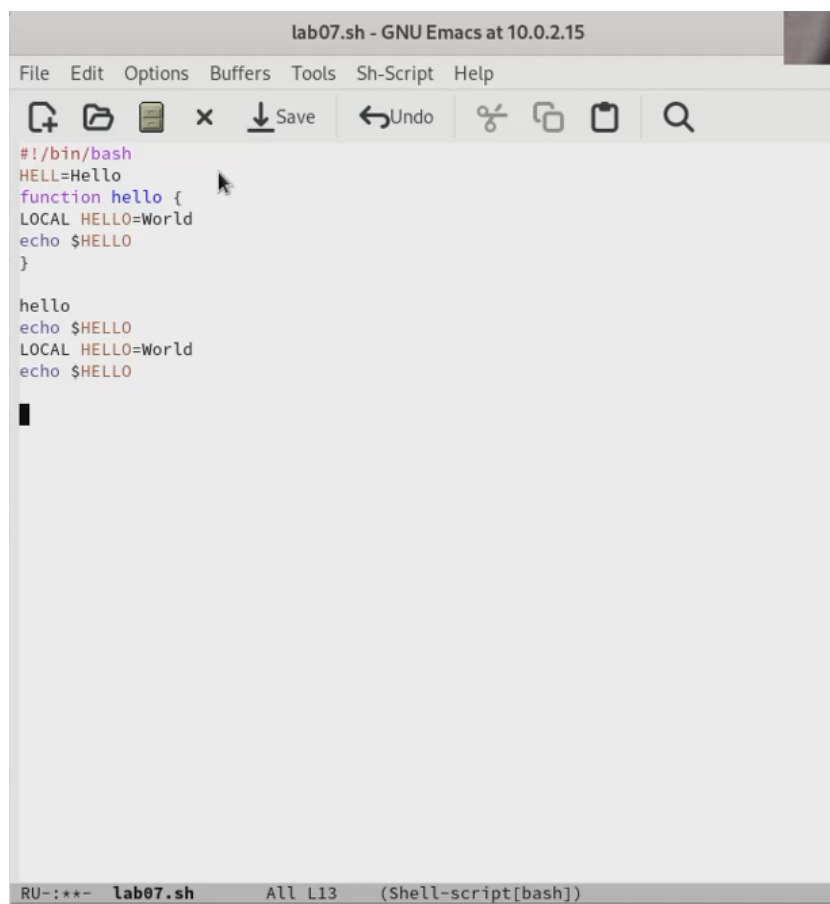


Рис. 3.18: закрытие

19. Теперь вновь переключилась между буферами, но уже без вывода их списка на экран (C-x b).



Рис. 3.19: переключение

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

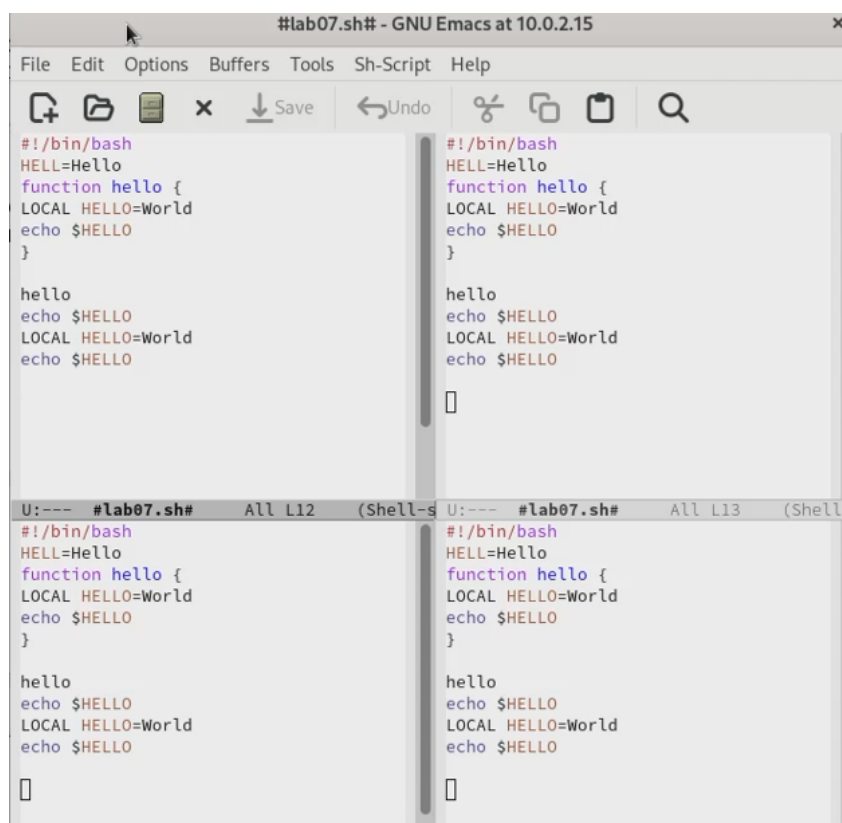


Рис. 3.20: деление

21. В каждом из четырёх созданных окон открыла новый буфер (файл) и ввела несколько строк текста.

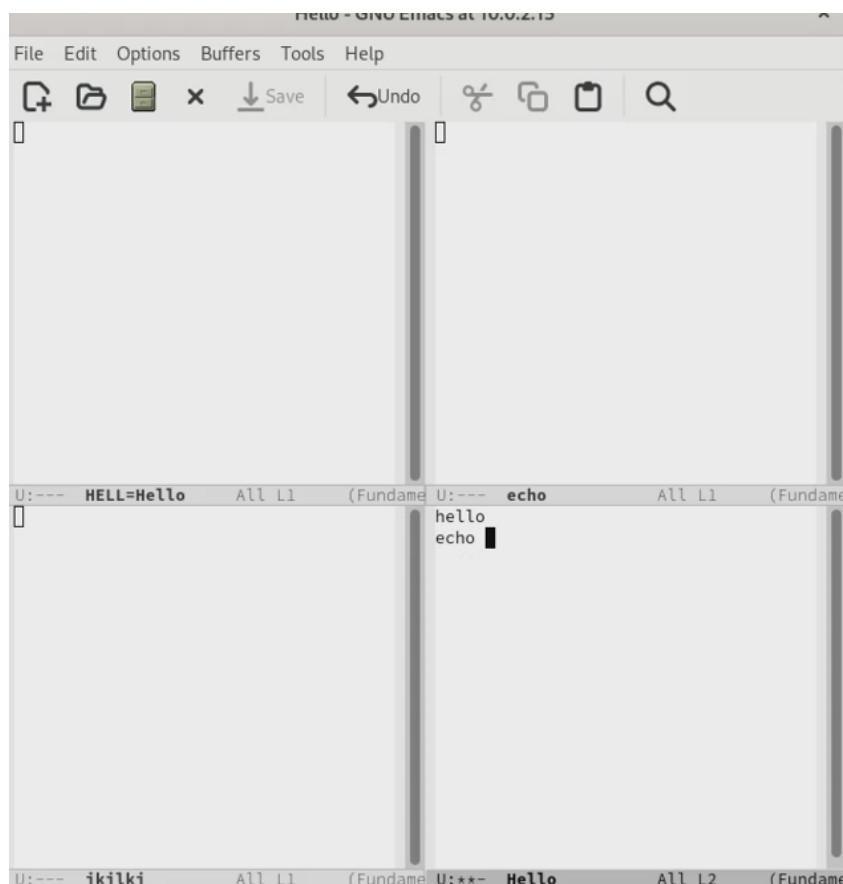


Рис. 3.21: текст

22. Переключилась в режим поиска (C-s) и нашла несколько слов, присутствующих в тексте.

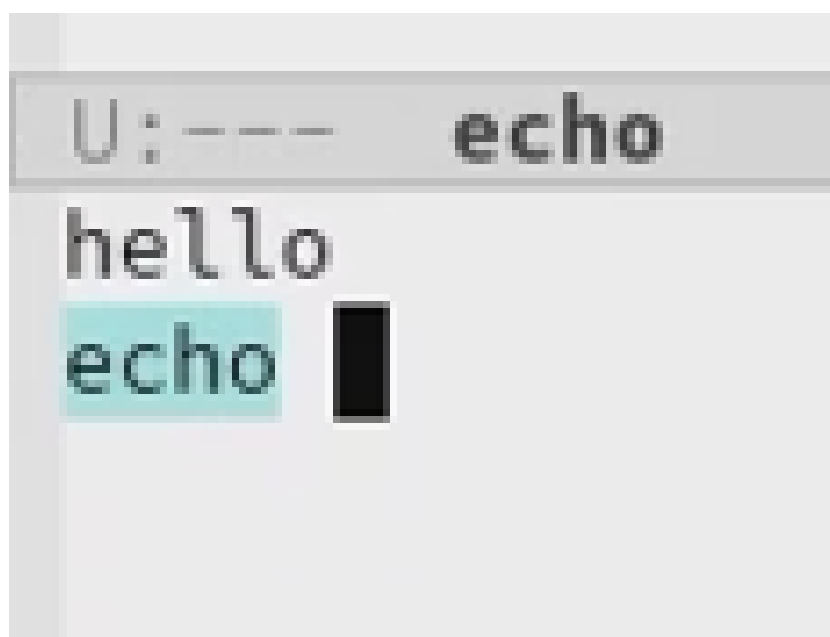


Рис. 3.22: поиск

23. Переключилась между результатами поиска, нажимая C-s



Рис. 3.23: переключение

4 Контрольные вопросы

- 1) 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 и т.д. 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.
- 2) Основную трудность для новичков при освоении данного редактора могут составлять большое количество команд, комбинаций клавиш, которые не получится все запомнить с первого раза и поэтому придется часто обращаться к справочным материалам.
- 3) Буфер – это объект, представляющий собой текст. Если имеется несколько буферов, то редактировать можно только один. Обычно буфер считывает данные из файла или записывает в файл данные из буфера. Окно – это область экрана, отображающая буфер. При запуске редактора отображается одно окно, но при обращении к некоторым функциям могут открыться

дополнительные окна. Окна Emacs и окна графической среды X Window – разные вещи. Одно окно X Window может быть разбито на несколько окон в смысле Emacs, в каждом из которых отображается отдельный буфер.

- 4) Да, можно.
- 5) При запуске Emacs по умолчанию создаются следующие буферы: ☒ «scratch» (буфер для несохраненного текста) ☒ «Messages» (журнал ошибок, включающий также информацию, которая появляется в области EchoArea) ☒ «GNU Emacs» (справочный буфер о редакторе)
- 6) C-c | сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и нажимаю «|» C-c C-| сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и, удерживая «ctrl», нажимаю «|»
- 7) Чтобы поделить окно на две части необходимо воспользоваться комбинацией «Ctrl-x 3» (по вертикали) или «Ctrl-x 2» (по горизонтали)
- 8) Настройки Emacs хранятся в файле .emacs.
- 9) По умолчанию клавиша «☒» удаляет символ перед курсором, но в редакторе её можно переназначить. Для этого необходимо изменить конфигурацию файла .emacs.
- 10) Более удобным я считаю редактор emacs, потому что в нем проще открывать другие файлы, можно использовать сразу несколько окон, нет «Командного режима», «Режима ввода», «Режима командной строки», которые являются немного непривычными и в какой-то степени неудобными.

5 Выводы

Я познакомилась с операционной системой Linux, получила практические навыки работы с редактором Emacs.