

Лабораторная работа №9

Текстовый редактор emacs

Парфенова Елизавета Евгеньевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	28
5	Контрольные вопросы	29

Список иллюстраций

3.1	Вызов текстового редактора	8
3.2	Создание файла	9
3.3	Запись текста в файл	10
3.4	Сохранение файла	11
3.5	Вырезание строки	12
3.6	Вставка вырезанной строки	12
3.7	Выделение текста	13
3.8	Вставка скопированного текста	13
3.9	Вырезание области текста	14
3.10	Отмена последней операции	14
3.11	Курсор в начало строки	15
3.12	Курсор в конец строки	16
3.13	Курсор в начало буфера	17
3.14	Курсор в конец буфера	18
3.15	Список активных буферов	19
3.16	Переход в другой буфер	20
3.17	Закрытие второго буфера	20
3.18	Переход в другой буфер	21
3.19	Фрейм на 4 части	22
3.20	4 новых файла	23
3.21	Посик слова	24
3.22	Замена слова	25
3.23	Результат замены	26
3.24	Поиск слова другим способом	27

Список таблиц

1 Цель работы

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

2 Задание

1. Открыть emacs.
2. Создать файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).
3. Наберите нужный текст
4. Сохранить файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
5. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.
 - 5.1. Вырезать одной командой целую строку (C-k).
 - 5.2. Вставить эту строку в конец файла (C-y).
 - 5.3. Выделить область текста (C-space).
 - 5.4. Скопировать область в буфер обмена (M-w).
 - 5.5. Вставить область в конец файла.
 - 5.6. Вновь выделить эту область и на этот раз вырезать её (C-w).
 - 5.7. Отмените последнее действие (C-/).
6. Научитесь использовать команды по перемещению курсора.
 - 6.1. Переместите курсор в начало строки (C-a).
 - 6.2. Переместите курсор в конец строки (C-e).
 - 6.3. Переместите курсор в начало буфера (M-<).
 - 6.4. Переместите курсор в конец буфера (M->).

7. Управление буферами.

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

7.2. Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер. 7.3. Закройте это окно (C-x 0).

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

8. Управление окнами.

8.1. 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)

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

9. Режим поиска

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

9.2. Переключайтесь между результатами поиска, нажимая C-s.

9.3. Выйдите из режима поиска, нажав C-g.

9.4. 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.

9.5. Испробуйте другой режим поиска, нажав M-s o. Объясните, чем он отличается от обычного режима?

3 Выполнение лабораторной работы

Первым шагом я вызвала в терминале текстовый редактор emacs, набрав одноименную команду. (рис. 3.1)

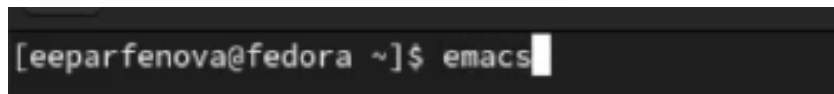


Рис. 3.1: Вызов текстового редактора

После я создала новый файл с помощью комбинации клавиш *Ctrl+x Ctrl+f*. Назвала его, как указано в файле Лабораторной работы, - lab07.sh. (рис. 3.2)

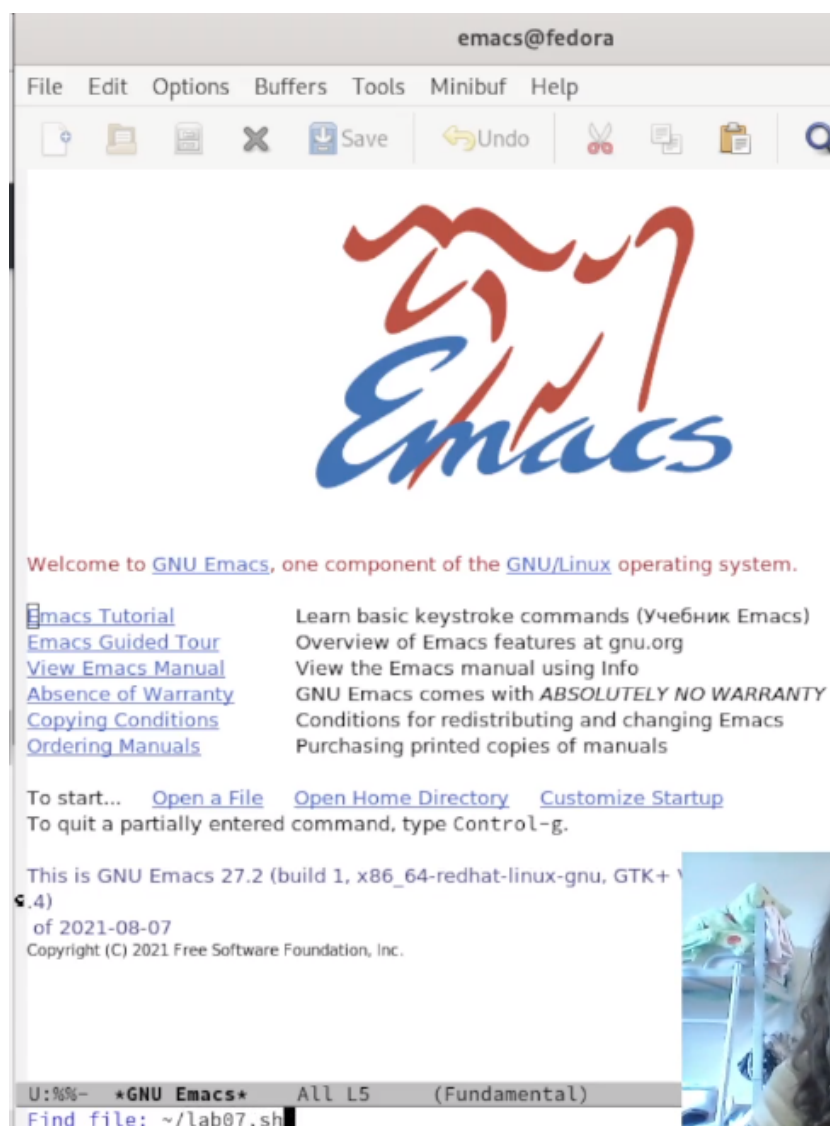


Рис. 3.2: Создание файла

Далее я написала в созданном файле нужный текст, который также был указан в файле Лабораторной работы. (рис. 3.3)

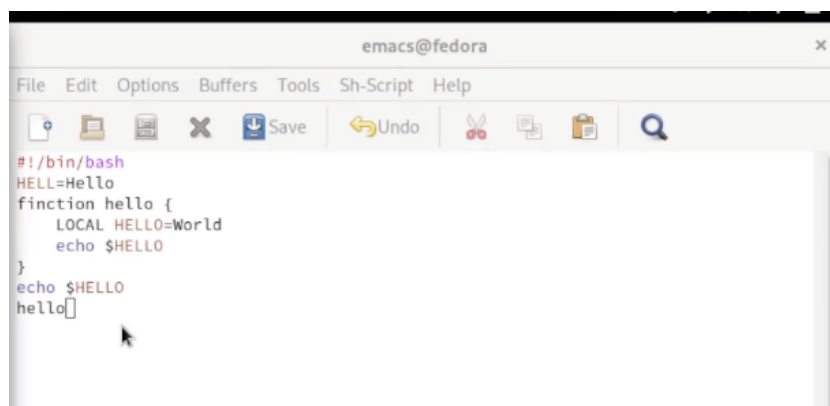


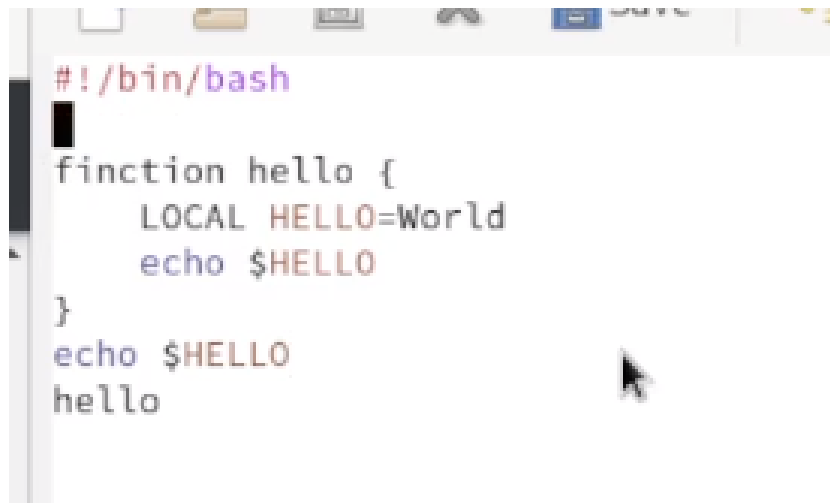
Рис. 3.3: Запись текста в файл

Далее я сохранила файл комбинацией клавиш *Ctrl+x Ctrl+s*. (рис. 3.4)



Рис. 3.4: Сохранение файла

Следующим заданием было проделать с текстом процедуры редактирования. Вначале я вырезала строку клавишами *Ctrl+k* (рис. 3.5)



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

Рис. 3.5: Вырезание строки

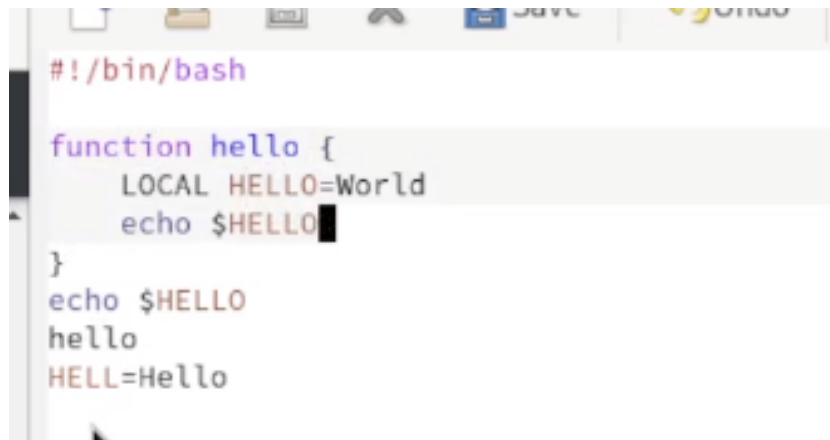
После я вставила вырезанную строку в конец текста клавишами *Ctrl+y*. (рис. 3.6)



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

Рис. 3.6: Вставка вырезанной строки

Затем я выделила область текста с помощью *Ctrl+Space* (рис. 3.7), клавишами *Alt+w* скопировала его и с помощью *Ctrl+y* втсавила в конец текста. (рис. 3.8)



```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
```

Рис. 3.7: Выделение текста



```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
```

Рис. 3.8: Вставка скопированного текста

Далее я вновь выделила эту же область *Ctrl+Space* и вырезала ее с помощью *Ctrl+w*. (рис. 3.9) Отменила последнюю операцию, используя *Ctrl+.* (рис. 3.10)

```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
```

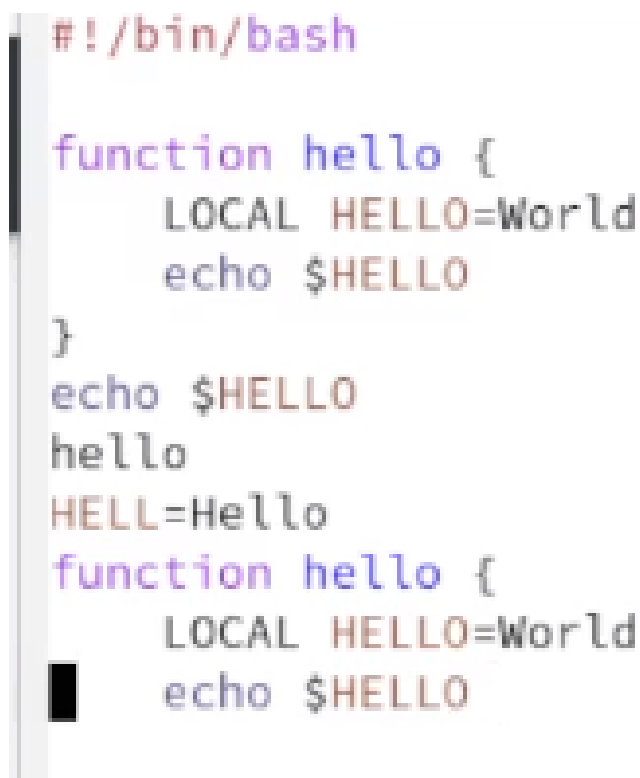
Рис. 3.9: Вырезание области текста

```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
```

Рис. 3.10: Отмена последней операции

После я перемещала курсор клавишами. Вначале переместила его в начало строки через *Ctrl+a*. (рис. 3.11) После - в конец строки, используя *Ctrl+e*. (рис. 3.12) Далее переместила курсор в начало буфера с помощью *Alt+-<* и в конец буфера - *Alt+->* (рис. 3.13) (рис. 3.14)



```
#!/bin/bash

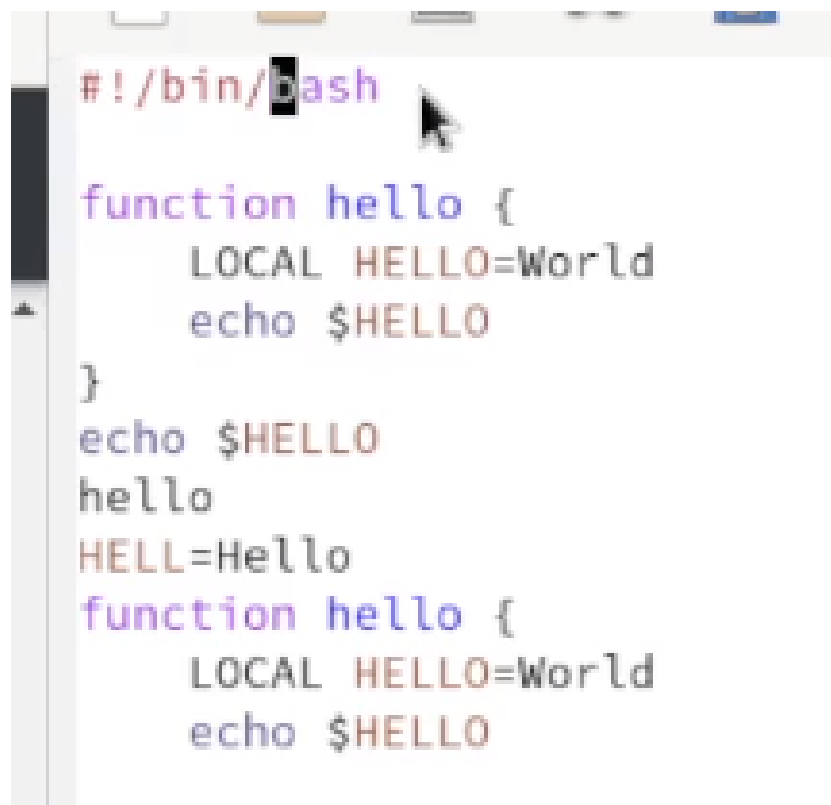
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
```

Рис. 3.11: Курсор в начало строки

```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
```

Рис. 3.12: Курсор в конец строки

A screenshot of a terminal window with a light gray background. The text is color-coded: purple for keywords like 'function', 'echo', and 'hello'; brown for variables like 'HELLO' and 'HELL'; and black for other text. A mouse cursor is positioned at the start of the first line. The script content is as follows:

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

Рис. 3.13: Курсор в начало буфера

```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
```

Рис. 3.14: Курсор в конец буфера

Следующим шагом я вывела список активных буферов на экран с помощью *Ctrl+x Ctrl+b*. (рис. 3.15)

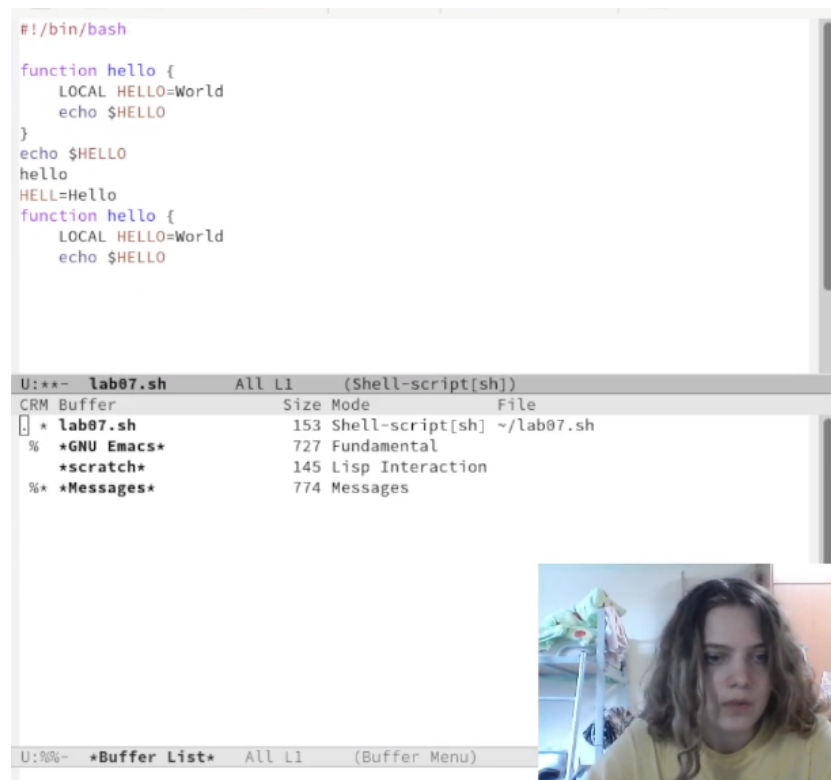


Рис. 3.15: Список активных буферов

После я перешла в один из активных буферов. Я выбрала '*GNU Emacs*'. (рис. 3.16) Далее я, используя *Ctrl+x 0*, закрыла буфер '*lab07.sh*'. (рис. 3.17)

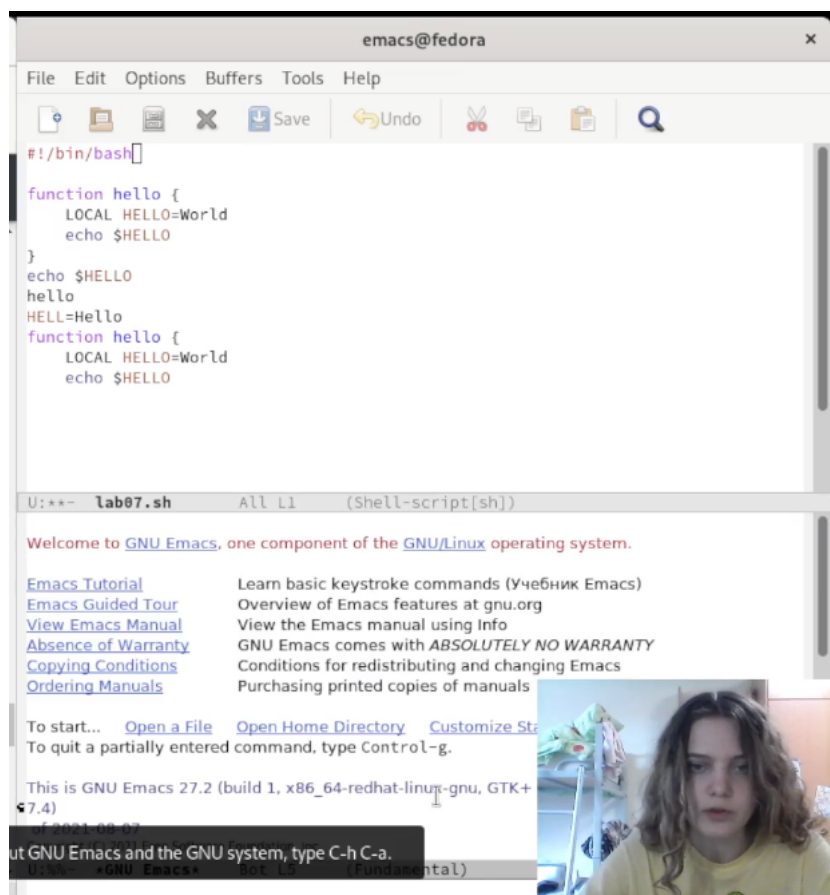


Рис. 3.16: Переход в другой буфер

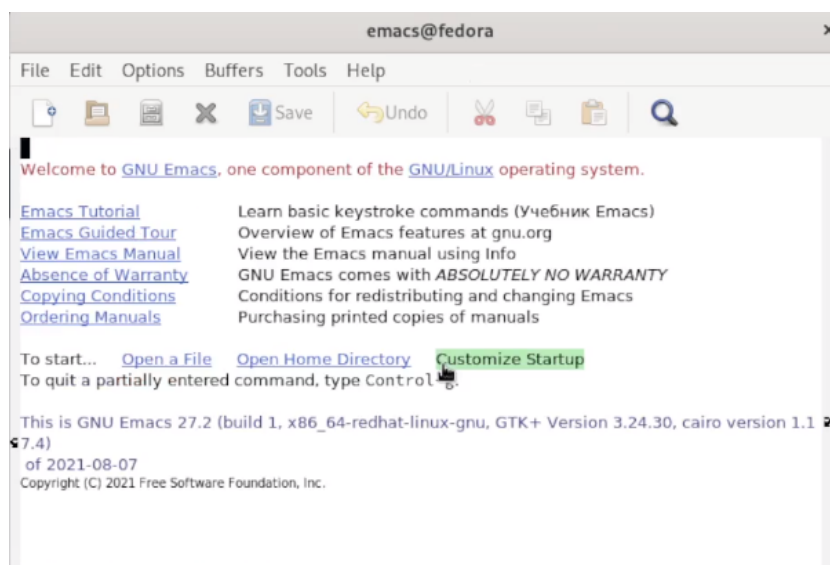


Рис. 3.17: Заккрытие второго буфера

После я переключилась на буфер *'GNU Emacs'* уже без открытия окна активных буферов, нажав *Ctrl+x b* и вписав нужное название. (рис. 3.18) Таким же образом я вернулась к исходному буферу.



Рис. 3.18: Переход в другой буфер

Следующим шагом я разделила фрейм на 4 части. Чтобы разделить его по вертикали я использовала *Ctrl+x 3*, а чтобы разделить по горизонтали - *Ctrl+x 2*, выполнив комбинацию на каждой из частей. (рис. 3.19)

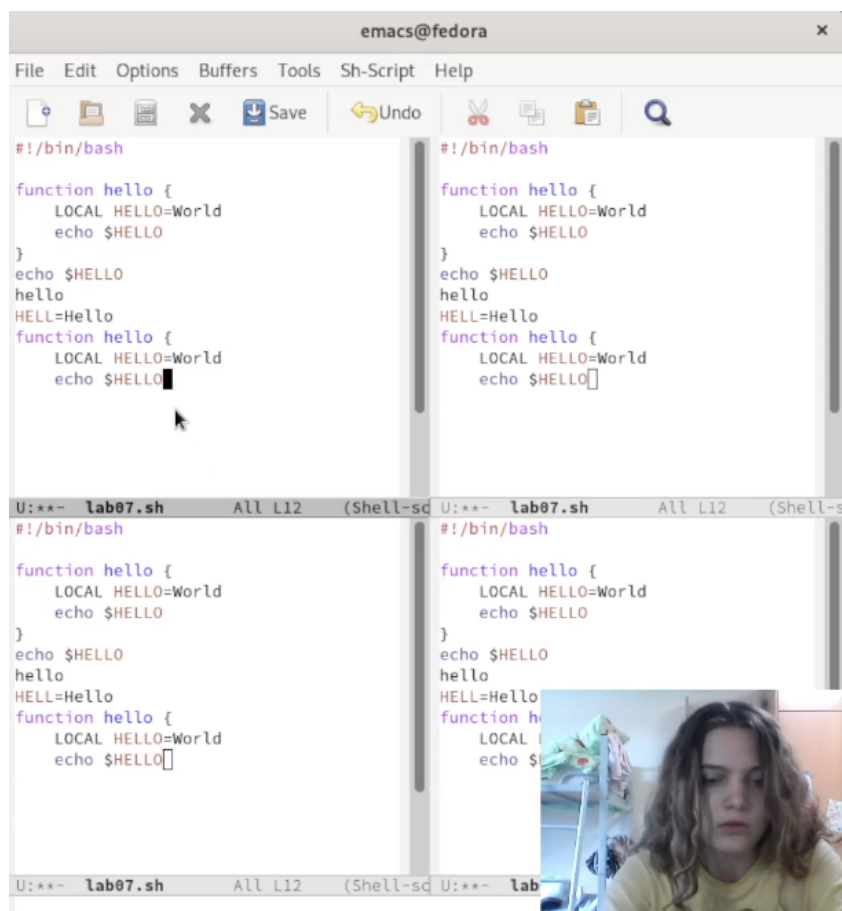


Рис. 3.19: Фрейм на 4 части

После в каждом из окон я создала новый буфер и записала по строчке текста.(рис. 3.20)

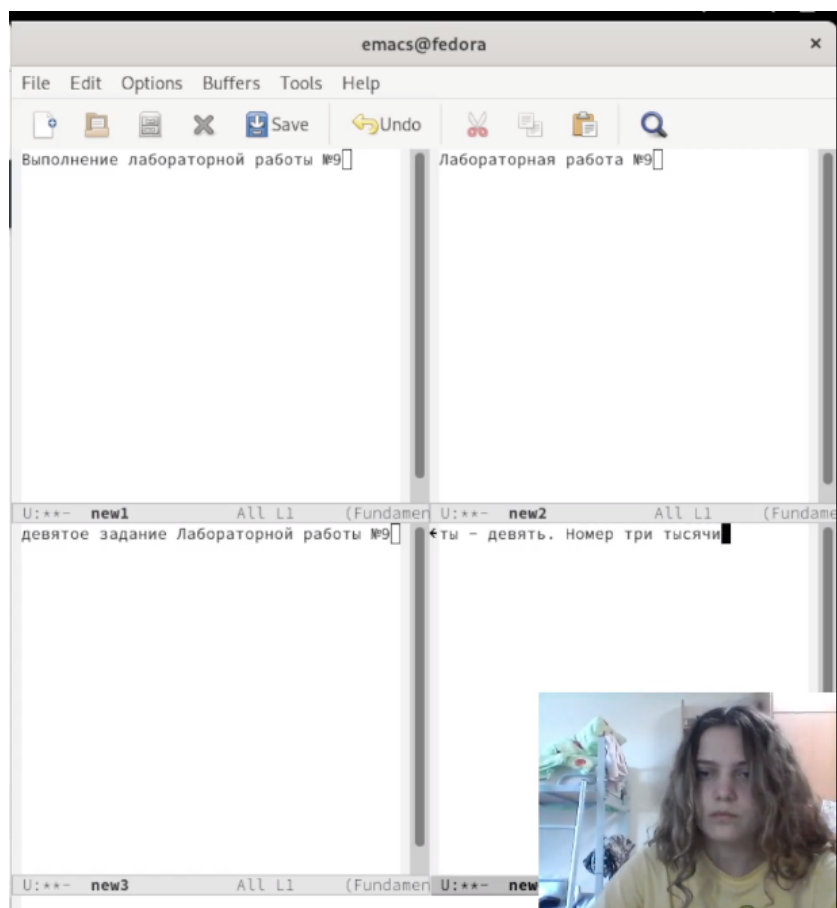


Рис. 3.20: 4 новых файла

Потом я переключилась в режим поиска с помощью *Ctrl+s* и в 4 файле запустила поиск слова “номер”. Найденные совпадения подсветились. Среди них можно переключаться с помощью того же *Ctrl+s*. (рис. 3.21) После я вышла из режима поиска, нажав *Ctrl+q*.

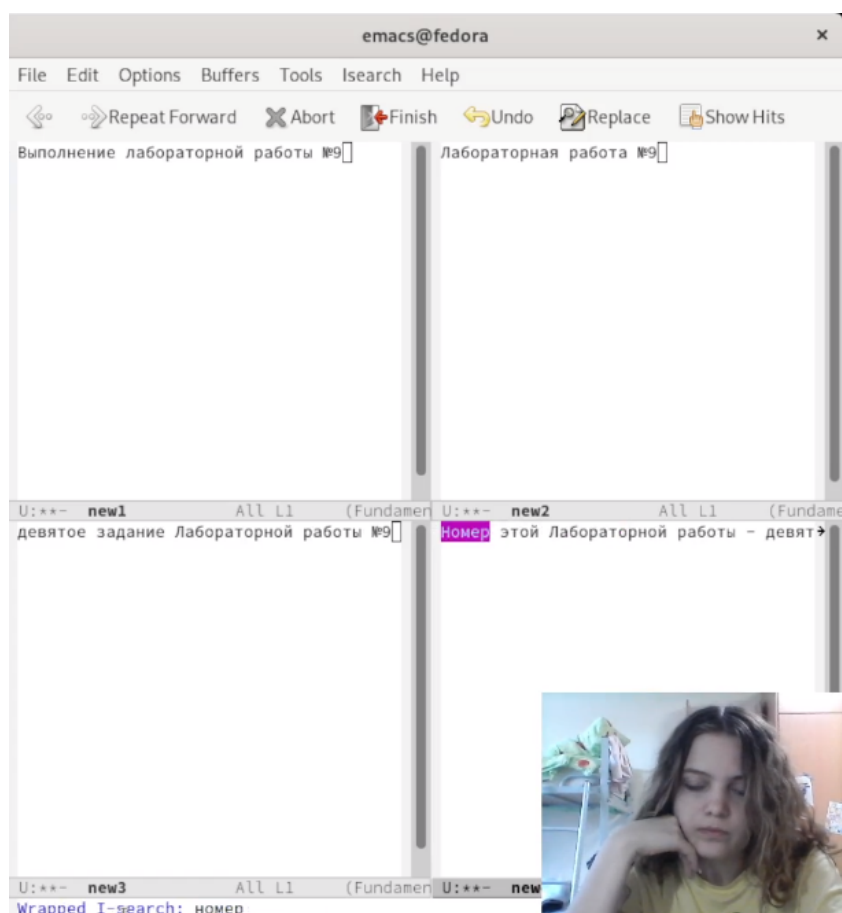


Рис. 3.21: Посик слова

Далее было необходимо перейти в режим поиска и замена, нажав *Alt+%*, однако у меня комбинация клавиш упрямо не хотела работать поэтому я перешла в режим поиска, нашла слово которое хочу заменить и, подсветив ее, выбрала опцию замены в меню в верхней части редактора. Я решила заменить слово “Выполнение” на слово “Подготовка” в 1 окне. Это слово я и ввела в нижней строке. (рис. 3.22) Затем я нажала *!* и замена была произведена. (рис. 3.23)

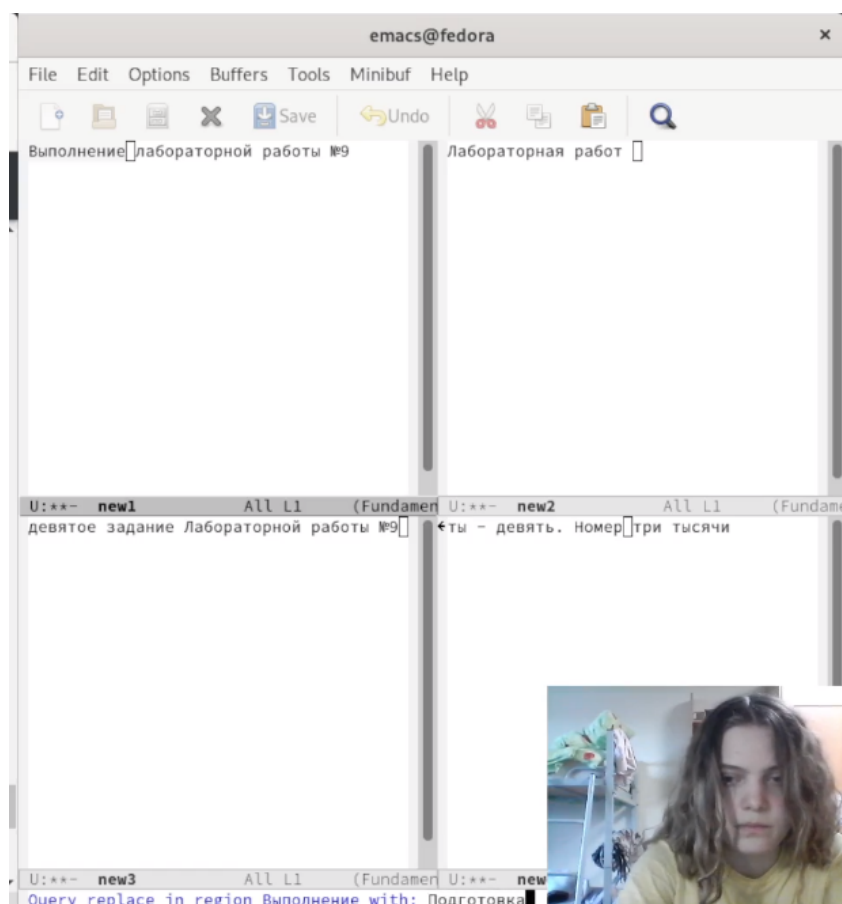


Рис. 3.22: Замена слова

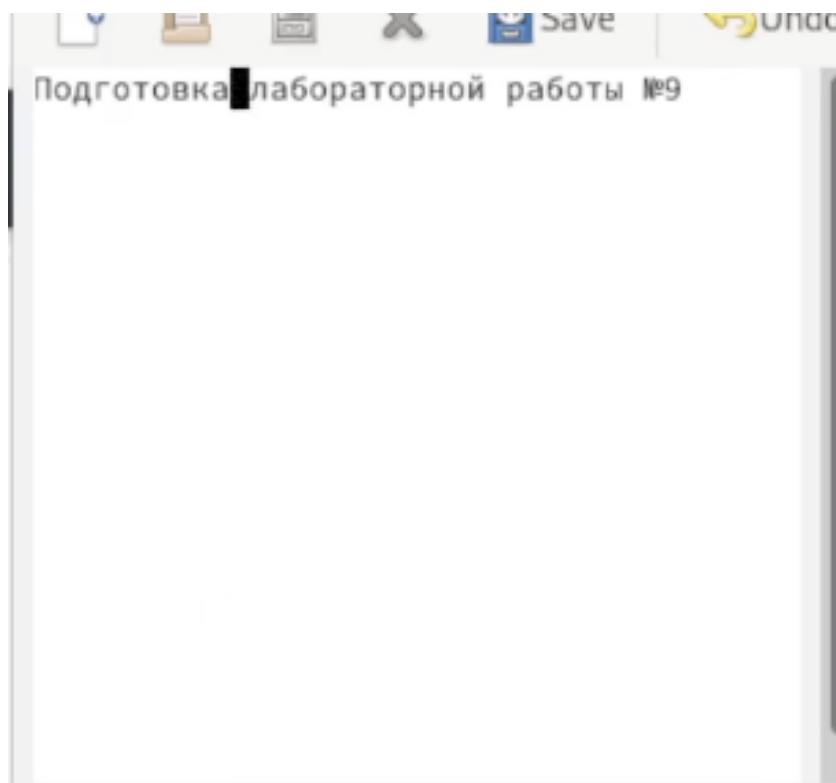


Рис. 3.23: Результат замены

Последним заданием было произвести поиск другим способом. Я искала в 4 окне все то же слово “номер”, но уже с помощью *Alt+s o*. Результаты этого поиска были выведены в другом буфере, причем с описанием, которое содержало информацию о том, сколько слов было найдено, в какой линии какого буфера. Также была приведена строчка, в которой нашлись совпадения. Это и есть существенное отличие двух режимов поиска. (рис. 3.24)

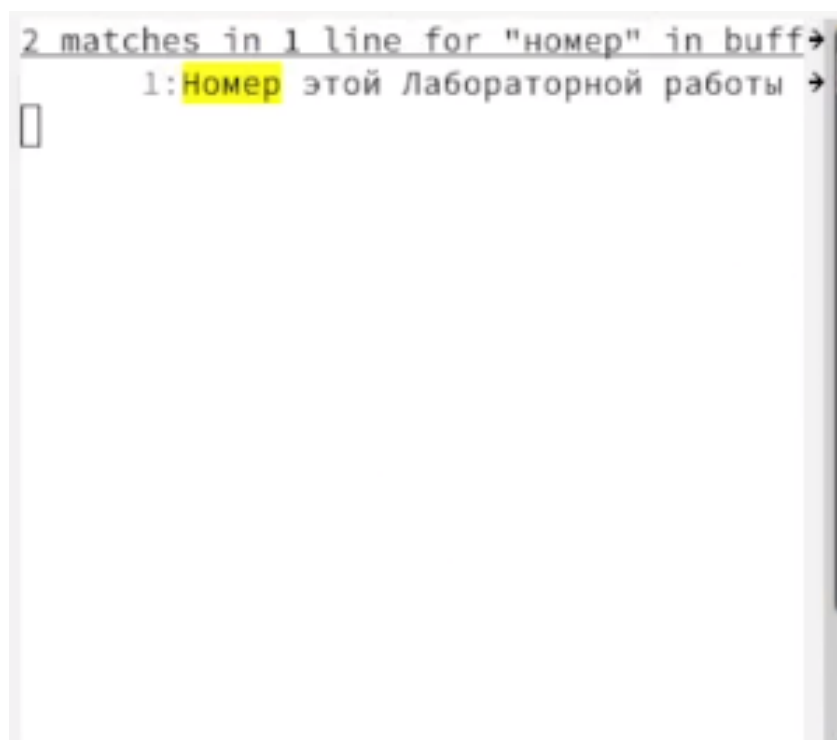


Рис. 3.24: Поиск слова другим способом

4 Выводы

Мы познакомились с операционной системой Linux и получили практические навыки работы с редактором Emacs.

5 Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs представляет собой мощный экраный редактор текста, написанный на языке высокого уровня Elisp.

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

Для работы с emacs требуется знать много о его функциях и о том, какими комбинациями клавиш они могут быть вызваны. Этих функций, как и комбинаций соответственно, достаточно много. Также сложность могут вызвать несовместимость кнопок вашего ПК и кнопок, заложенных в функциях редактора.

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

Буфер - это область, где пользователь набирает текст или как-то с ней работает, а окно - это область фрейма, в которой отображается этот буфер, либо же объединение буферов.

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

Да, можно.

5. Какие буферы создаются по умолчанию при запуске emacs?

’* GNU Emacs , Message , scratch *’

6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-с | и C-с C-|?

Ctrl+c Shift+ и Ctrl+c Ctrl + Shift+

7. Как поделить текущее окно на две части?

По вертикали - *Ctrl+x 3*, по горизонтали - *Ctrl+x 2*

8. В каком файле хранятся настройки редактора emacs?

Файл Emacs

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

Клавиша Backspace выполняет функцию клавиш Ctrl+k и ее можно переназначить.

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

Несмотря на то, что vi проще в использовании, мне больше понравился Emacs благодаря богатому интерфейсу и множеству разнообразных функций. Текст в нем редактируется лучше и интересней.