

# Лабораторная работа № 9. Текстовой редактор emacs

---

Leysan R. Abdullina

NEC-2022, 17 May

RUDN University, Moscow, Russian Federation

# Лабораторная работа № 9.

## Текстовой редактор emacs

---

## Цель работы

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

**Определение 1.** Буфер—объект, представляющий какой-либо текст.

Буфер может содержать что угодно, например, результаты компиляции программы или встроенные подсказки. Практически всё взаимодействие с пользователем, в том числе интерактивное, происходит посредством буферов.

**Определение 2.** Фрейм соответствует окну в обычном понимании этого слова. Каждый фрейм содержит область вывода и одно или несколько окон Emacs.

**Определение 3.** Окно—прямоугольная область фрейма, отображающая один из буферов. Каждое окно имеет свою строку состояния, в которой выводится следующая информация: название буфера, его основной режим, изменялся текст буфера и как далеко вниз по буферу расположен курсор. Каждый буфер находится только в одном из возможных основных режимов. Существующие основные режимы включают режим Fundamental (наименее специализированный), режим Text, режим Lisp, режим C, режим Texinfo и другие. Под второстепенными режимами понимается список режимов, которые включены в данный момент в буфере выбранного окна.

**Определение 4.** Область вывода—одна или несколько строк внизу фрейма, в которой Emacs выводит различные сообщения, а также запрашивает подтверждения и дополнительную информацию от пользователя.

**Определение 5.** Минибуфер используется для ввода дополнительной информации всегда отображается в области вывода.

**Определение 6.** Точка вставки—место вставки (удаления) данных в буфере.

**Определение 7.** Режим — пакет расширений, изменяющий поведение буфера Emacs при редактировании и просмотре кста (например, для редактирования исходного текста программ на языках C или Perl).

# Выполнение работы

1. Откроем emacs с помощью команды emacs (предварительно установив все через sudo dnf install emacs). (скриншот 1)

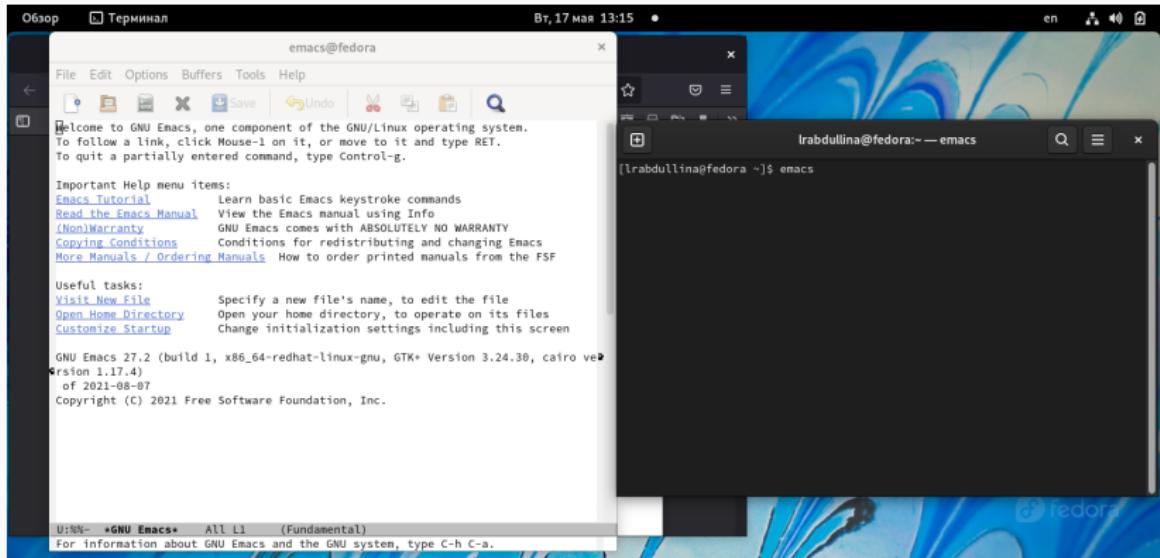


Figure 1: Открытие редактора emacs

# Выполнение работы

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

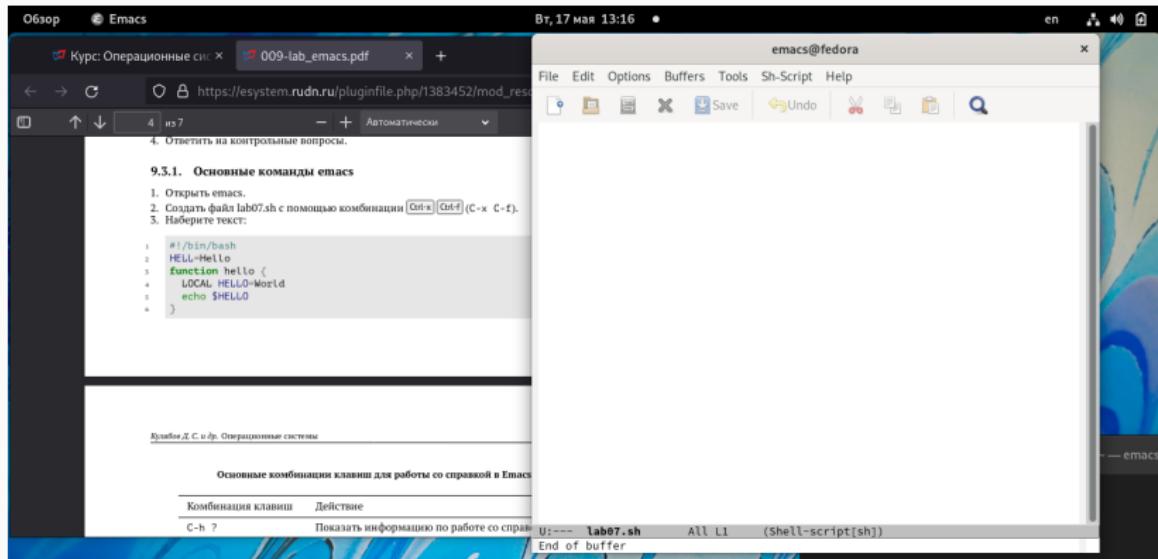


Figure 2: Создание файла lab07.sh

### 3. Наберем текст: (скриншот 3)

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

# Выполнение работы

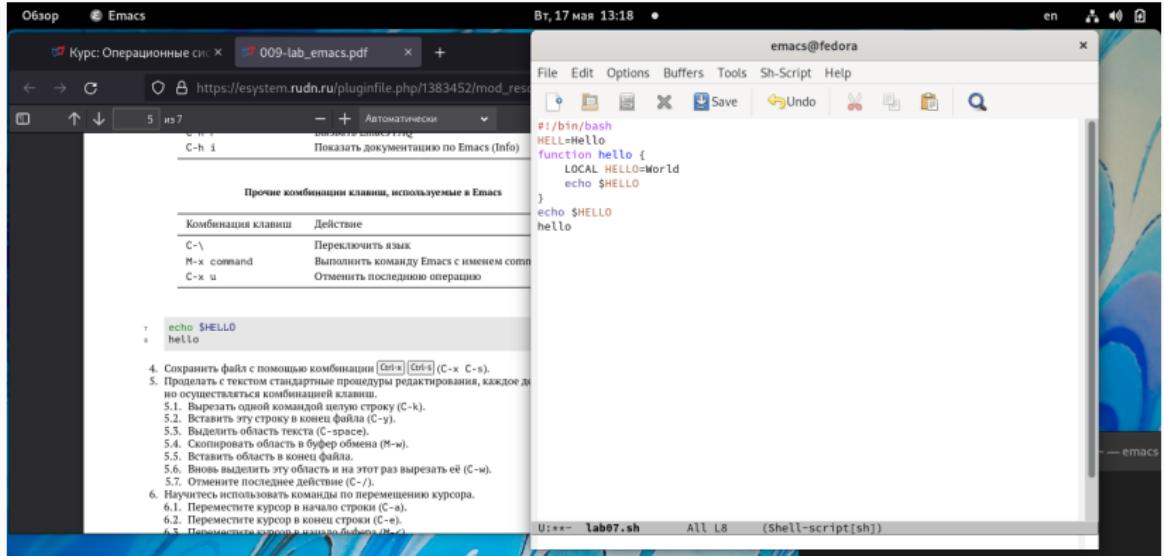


Figure 3: Набор текста в файле

# Выполнение работы

- Сохраним файл с помощью комбинации Ctrl-x Ctrl-s (обратим внимание на “погасший” занчик сохранения). (скриншот 4)

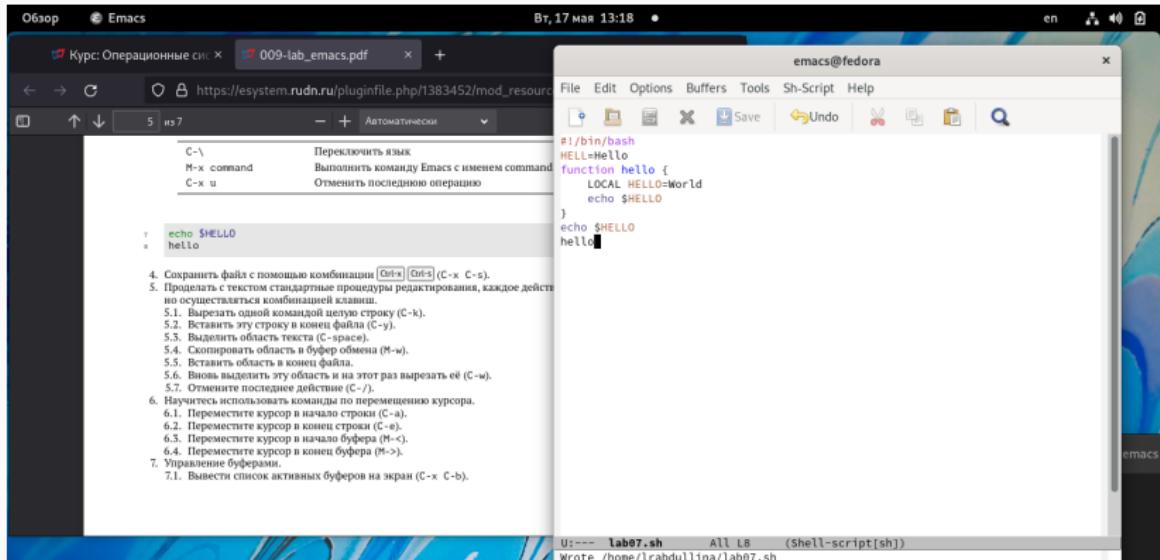


Figure 4: Сохранение файла

## Выполнение работы

5. Проделаем с текстом стандартные процедуры редактирования, каждое действие осуществляется комбинацией клавиш.
  - Вырезаем одной командой целую строку (C-k).(скриншот 5)

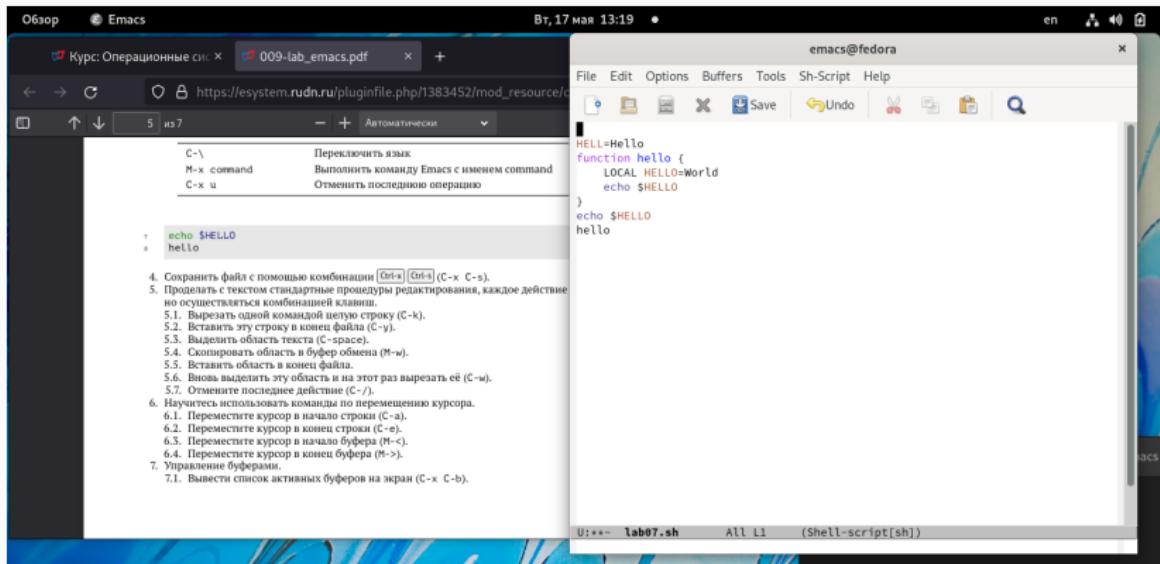


Figure 5: Вырезаем одной командой целую строку

# Выполнение работы

- Вставляем эту строку в конец файла (C-y).(скриншот 6)

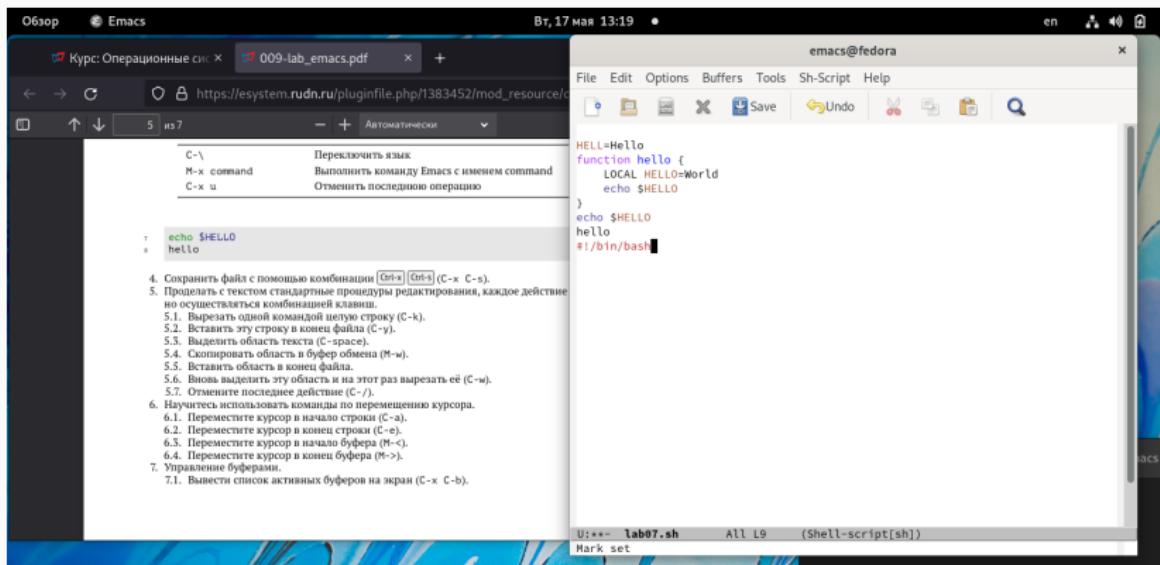


Figure 6: Вставляем эту строку в конец файла

# Выполнение работы

- Выделим область текста (C-space), а именно последние 2 строчки.
- Скопируем область в буфер обмена (M-w).
- Вставим область в конец файла.(скриншот 7)

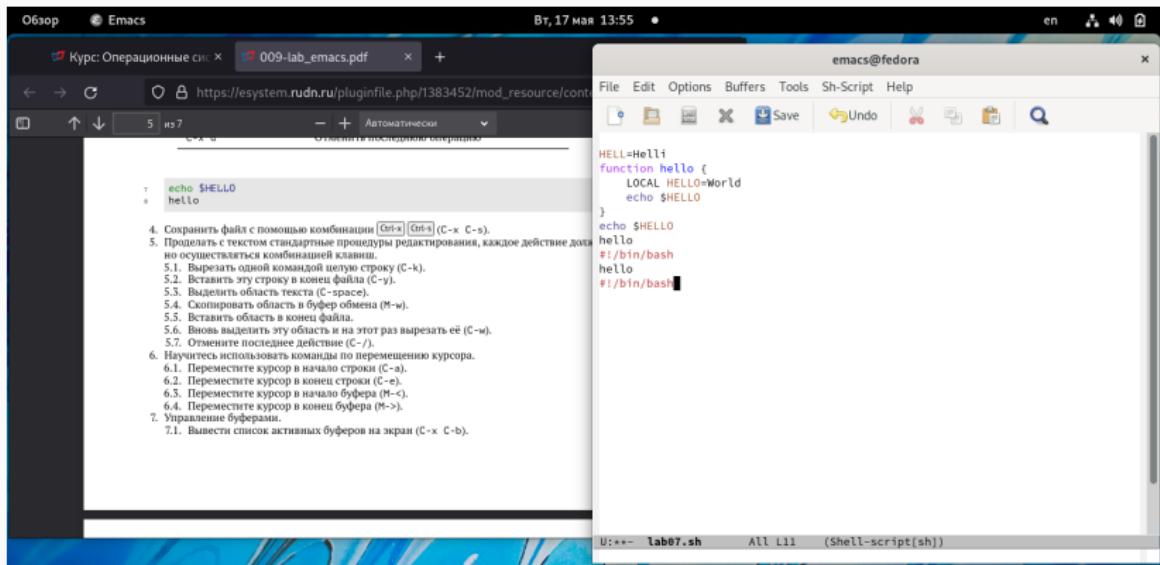


Figure 7: Вставляем область в конец файла

# Выполнение работы

- Вновь выделим эту область и на этот раз вырежем её (C-w).(скриншот 8)

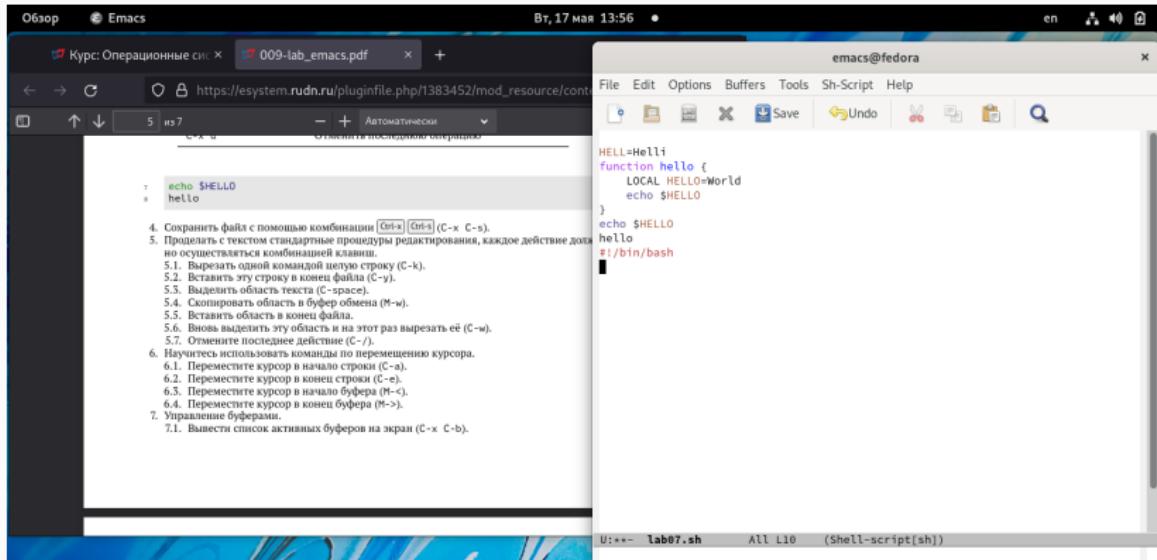


Figure 8: Вырезаем одной командой целую строку

# Выполнение работы

- Отменим последнее действие (C-/.)(скриншот 9)

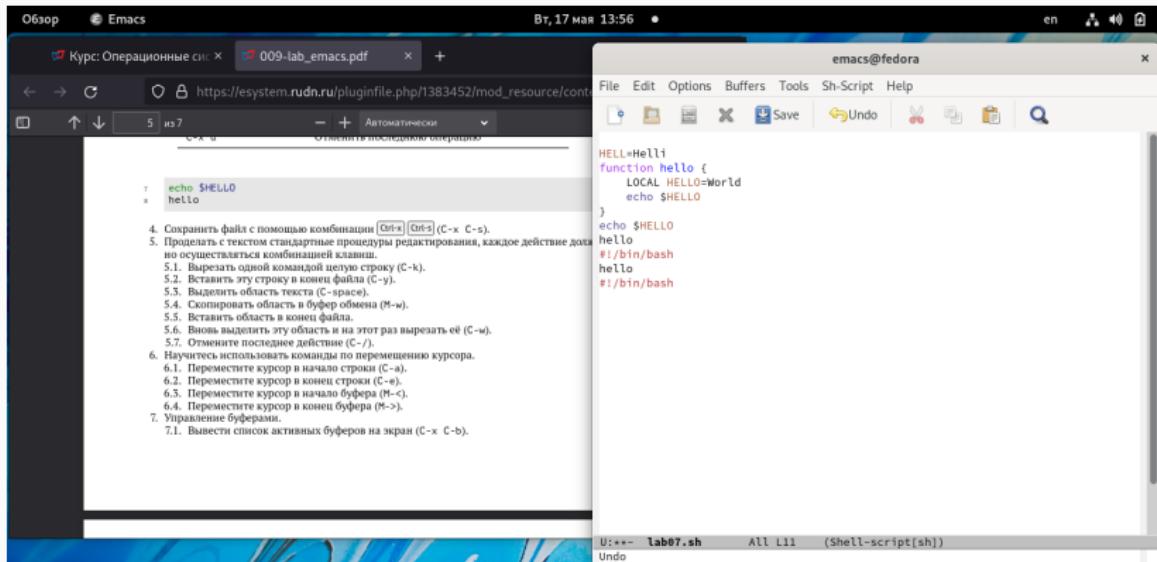


Figure 9: Отмена последнего действия

# Выполнение работы

## 6. Научимся использовать команды по перемещению курсора.

- Переместите курсор в начало строки (C-a). (скриншот 10)

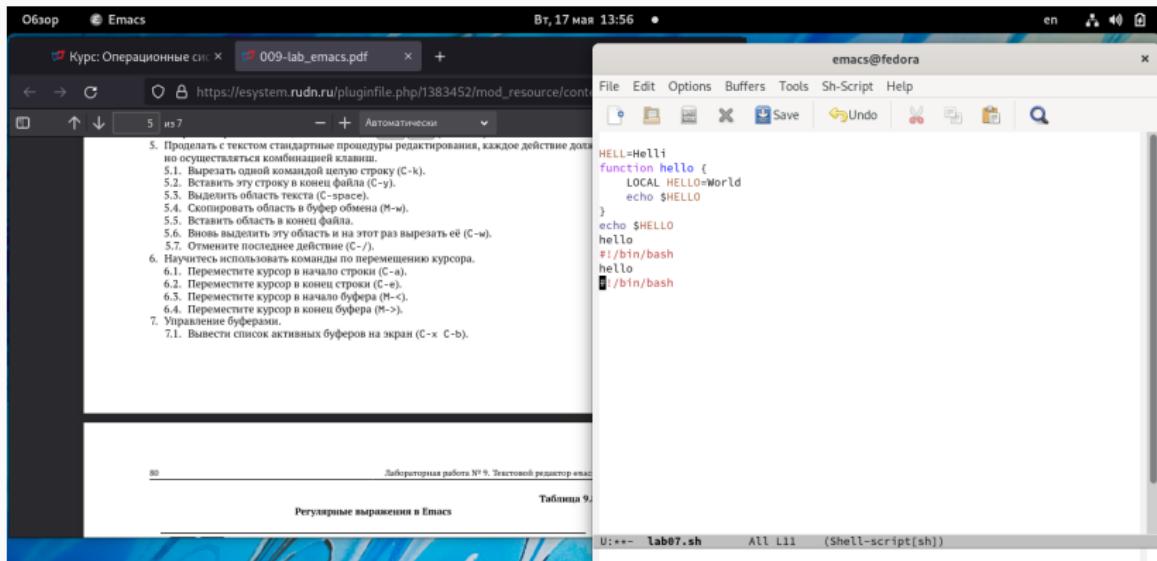


Figure 10: Перемещение курсора в начало строки

# Выполнение работы

- Переместите курсор в конец строки (C-e).(скриншот 11)

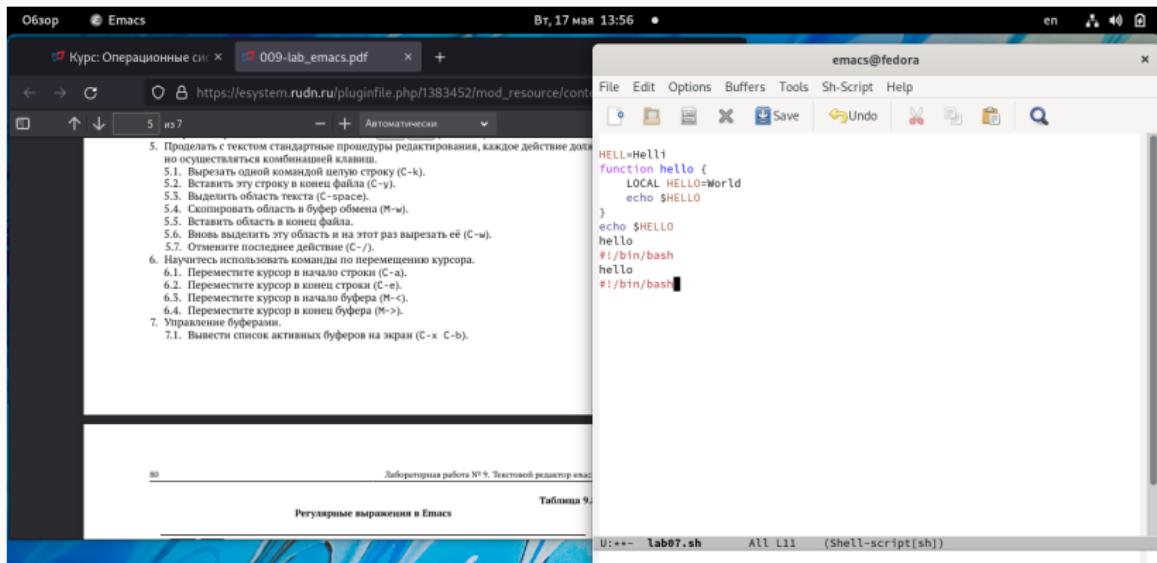


Figure 11: Перемещение курсора в конец строки

# Выполнение работы

- Переместите курсор в начало буфера ( $M-<$ ).(скриншот 12)

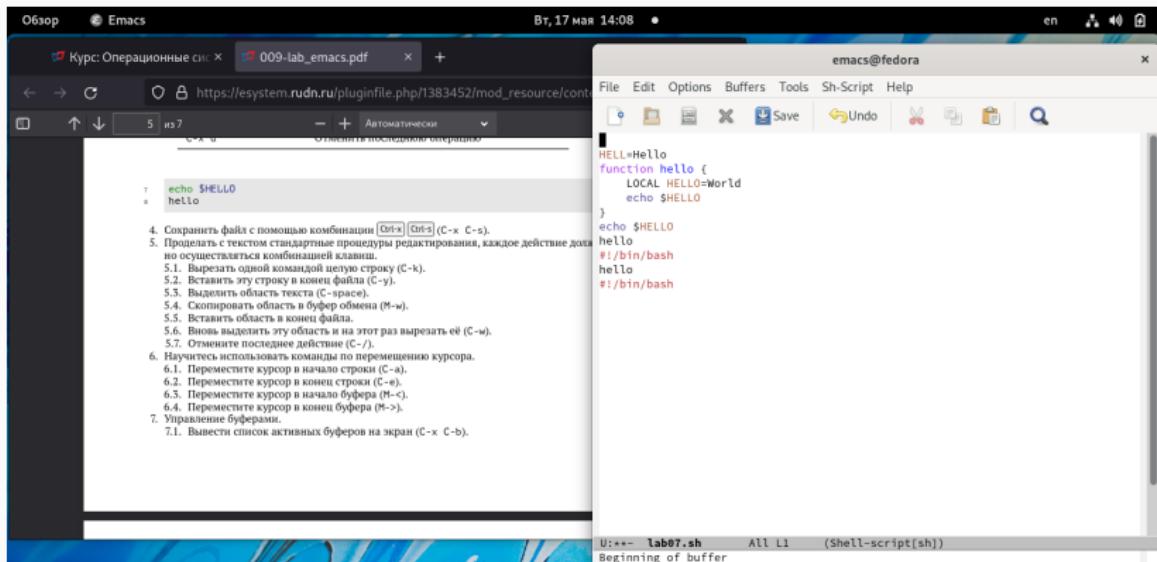


Figure 12: Перемещене курсора в начало буфера

## Выполнение работы

---

- Переместите курсор в конец буфера ( $M->$ ).

# Выполнение работы

## 7. Управление буферами.

- Выведим список активных буферов на экран (C-x C-b) (скриншот 13)

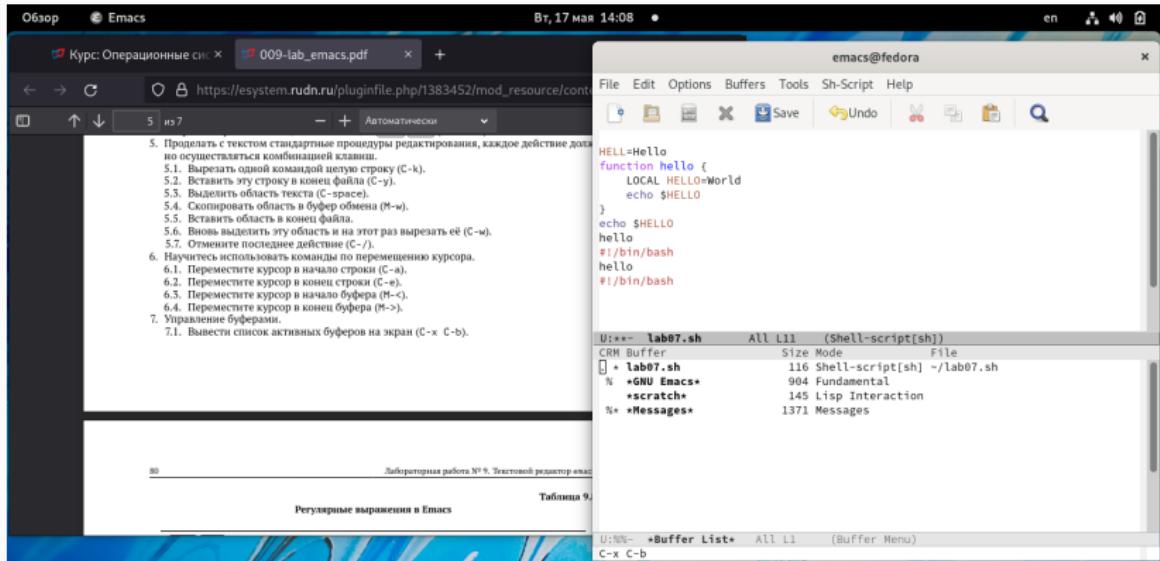


Figure 13: Список активных буферов

# Выполнение работы

- Переместимся во вновь открытое окно ((C-x) o) со списком открытых буферов и переключимся на другой буфер.(скриншот 14)

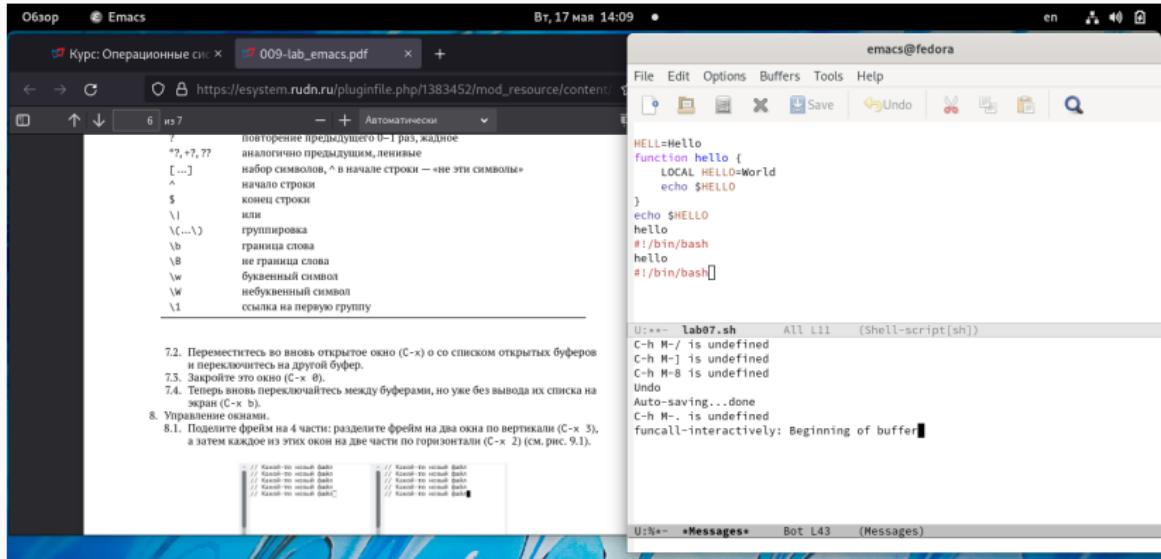


Figure 14: Переключение на другой буфер

# Выполнение работы

- Закроем это окно ( $C-x 0$ ). (скриншот 15)

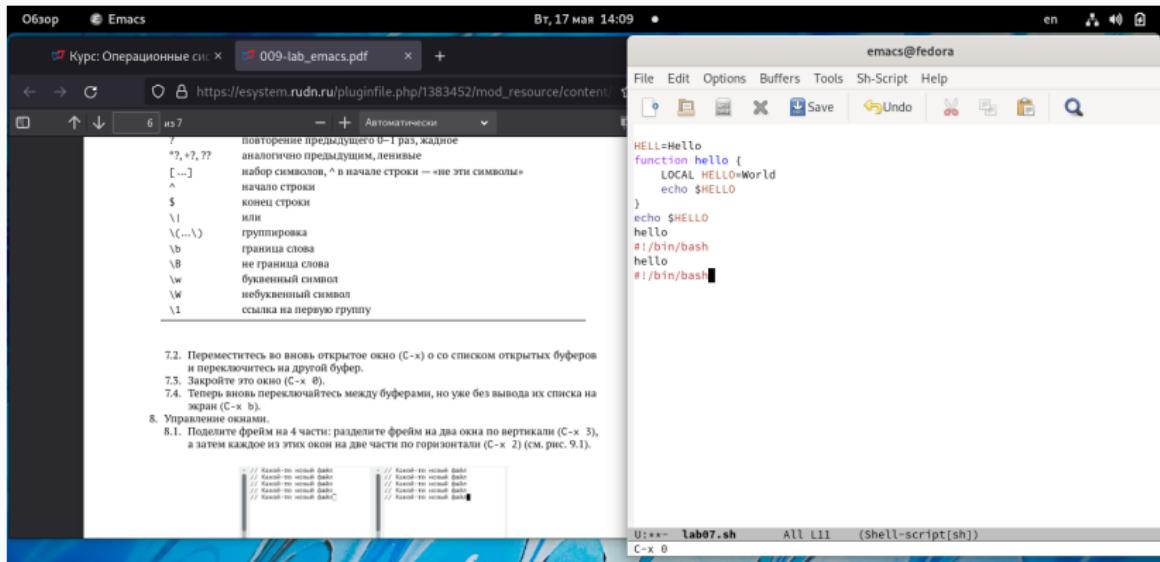


Figure 15: Закрытие окна

# Выполнение работы

- Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран ( $C-x b$ ). (скриншоты 16, 17)

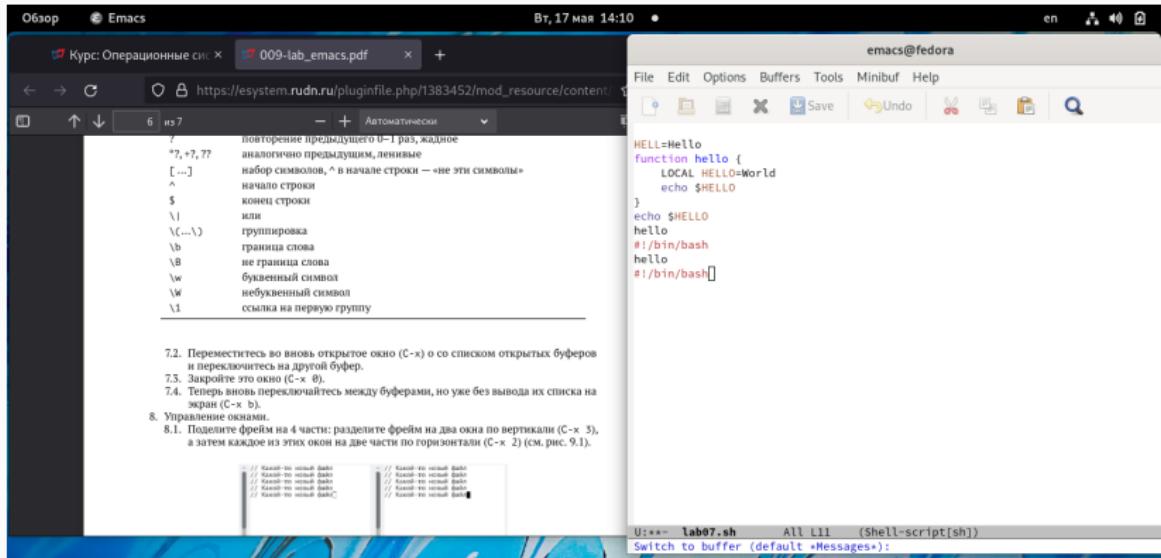


Figure 16: Ввод (внизу экрана) нужного нам буфера

# Выполнение работы

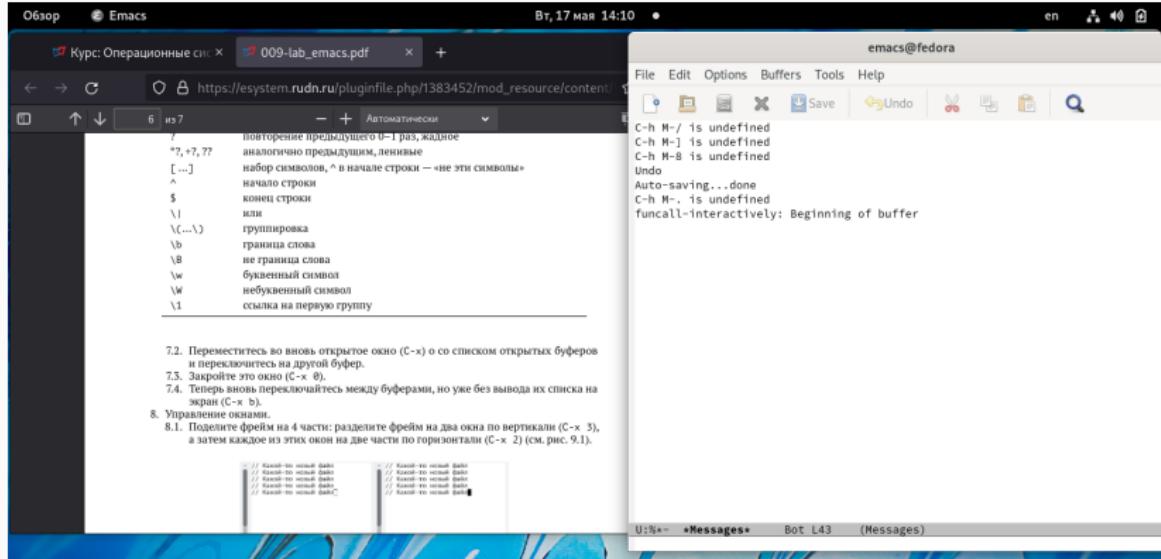


Figure 17: Переключение буфера

# Выполнение работы

## 8. Управление окнами.

- Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали ( $C-x 3$ ), а затем каждое из этих окон на две части по горизонтали ( $C-x 2$ ). (скриншот 18)

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

U:\*\*\*- lab07.sh All 111 (Shell-script[sh])

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

U:\*\*\*- lab07.sh All 111 (Shell-script[sh])

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

U:\*\*\*- lab07.sh All 111 (Shell-script[sh])

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

# Выполнение работы

- В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста.(скриншот 19)

The screenshot shows four separate Emacs windows, each titled "lab07.sh". Each window contains the same shell script code:

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

Figure 19: Ввод текста

# Выполнение работы

## 9. Режим поиска

- Переключимся в режим поиска (C-s) и найдем несколько слов, присутствующих в тексте (слово: кто).(скриншот 20)

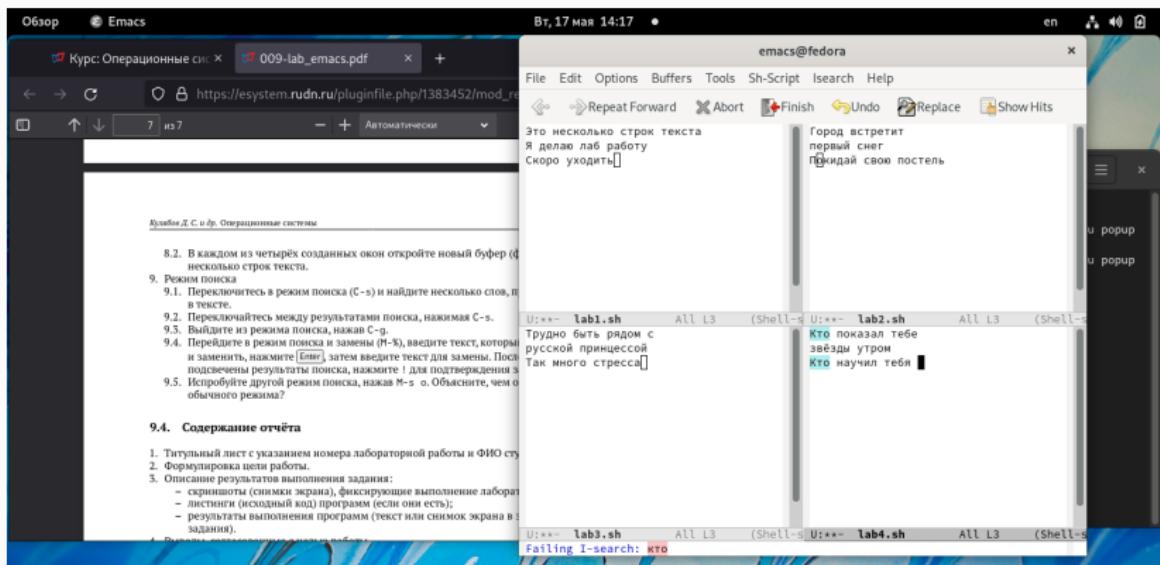


Figure 20: Поиск слова “кто”

# Выполнение работы

- Переключаем между результатами поиска, нажимая С-с.(скриншот 21)

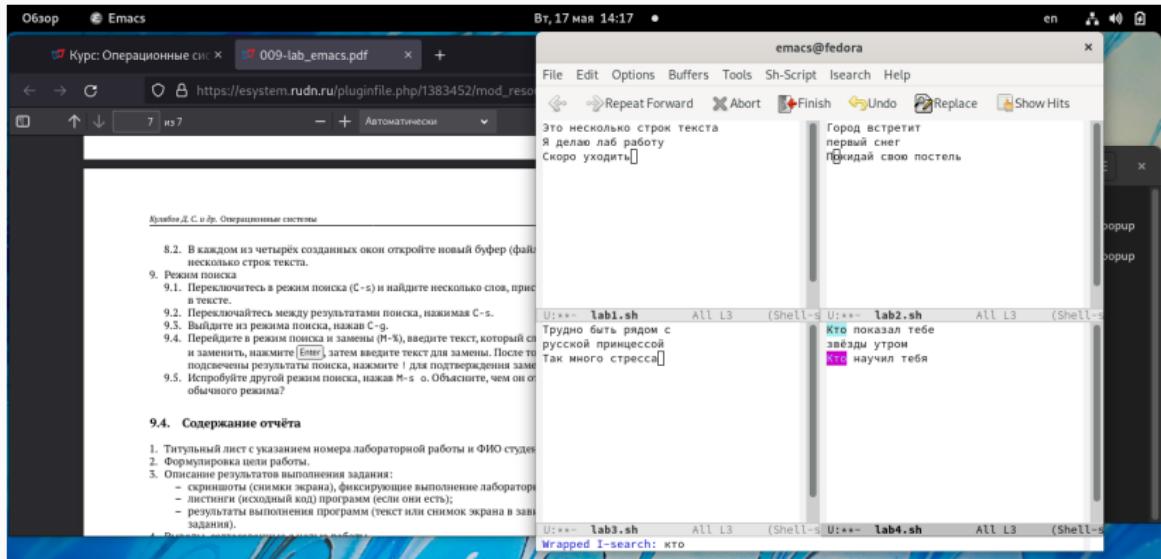


Figure 21: Переключения между словами “кто”

# Выполнение работы

- Выйдите из режима поиска, нажав C-g. (скриншот 22)

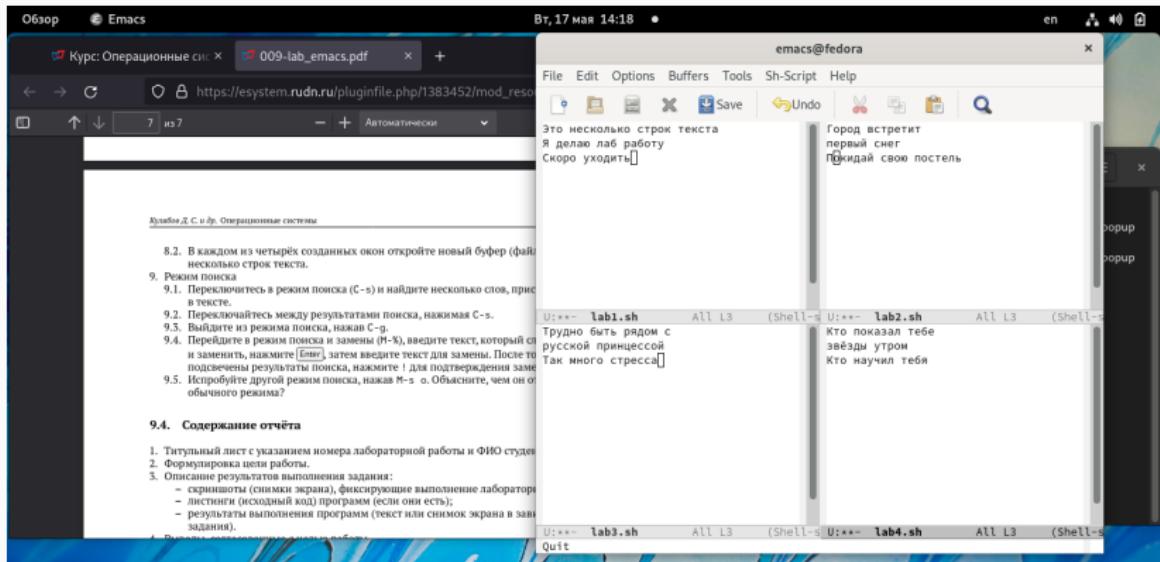


Figure 22: Выход из режима поиска

# Выполнение работы

- Перейдем в режим поиска и замены (**M-%**), введем текст, который следует найти и заменить, нажмем **Enter**, затем введем текст для замены. После того как будут подсвечены результаты поиска, нажмем **!** для подтверждения замены. (скриншот 23)

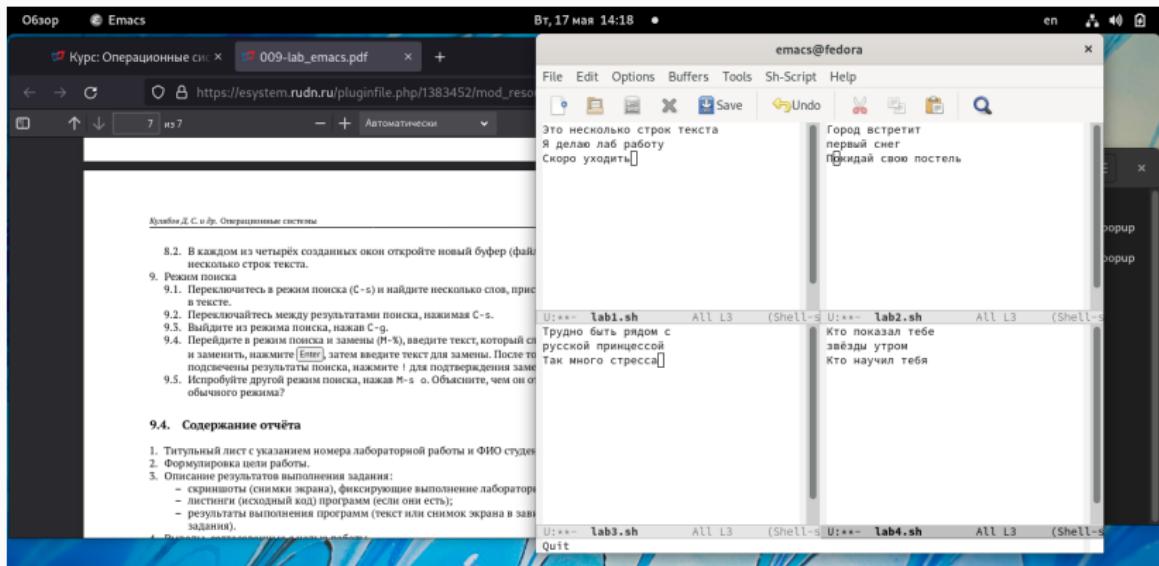


Figure 23: Замена текста (кто на он)

# Выполнение работы

- Испробуем другой режим поиска, нажав M-s о. Объясним, чем он отличается от обычного режима? Отдельно выводит строчку и показывает документ, где нашлось соответствие (скриншот 24)

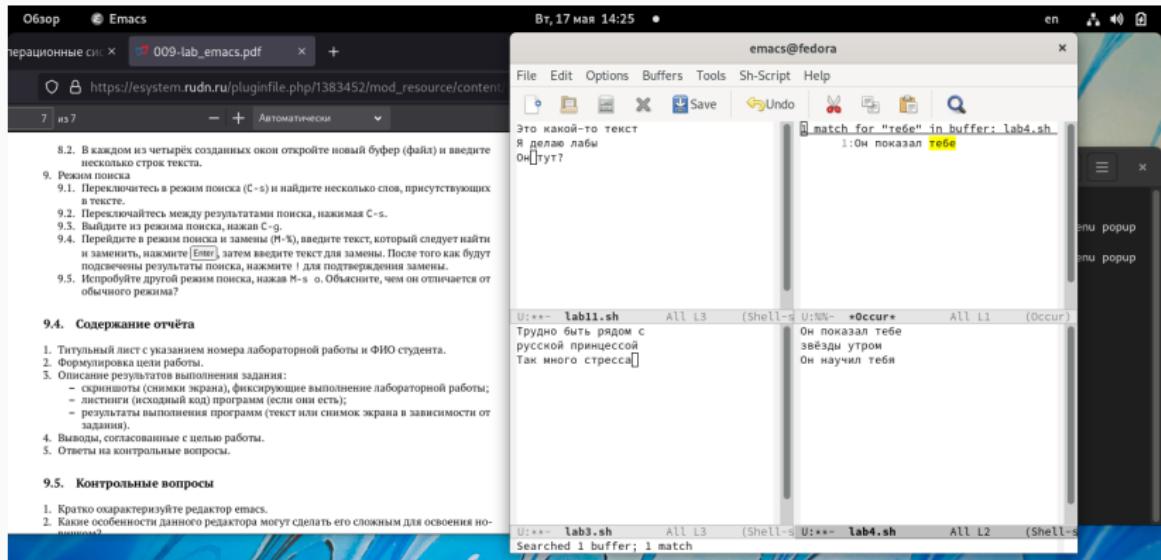


Figure 24: Другой режим поиска

## Выводы

В ходе лабораторной работы мы ознакомились с операционной системой Linux. Получили практические навыки работы с редактором Emacs.