Лабораторная работа №11

Текстовой редактор emacs

Полякова Юлия Александровна

Содержание

# 1 Цель работы

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

# 2 Задание

1. Ознакомиться с теоретическим материалом.
2. Ознакомиться с редактором emacs.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы.

# 3 Выполнение лабораторной работы

1. Открываем emacs, написав в командной строке emacs. Создаем файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f). Набираем предложенный текст (рис. 1).

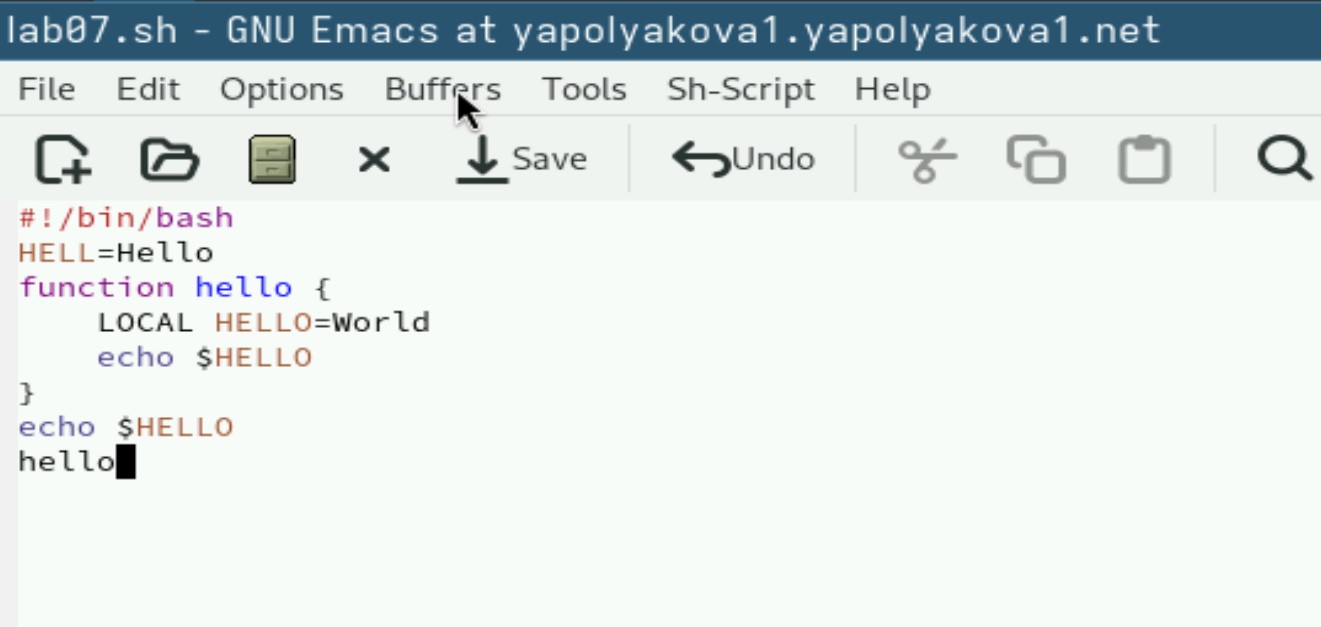


Рис. 1: Создание и заполнение файла

1. Сохранияем файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s). (рис. 2)

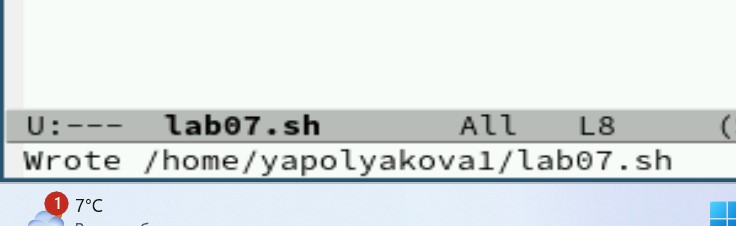


Рис. 2: Сохранение файла

1. Проделываем с текстом стандартные процедуры редактирования, каждое действие должно осуществляем комбинацией клавиш. Вырезаем одной командой целую строку (С-k), я вырезала echo $HELLO. Вставляем эту строку в конец файла (C-y). Выделяем область текста (C-space и я использовала стрелки), я выделила две строки hello и echo $HELLO. Копируем область в буфер обмена (Alt-w). Вставляем область в конец файла. Вновь выделяем эту область и на этот раз вырезаем её (C-w). Отменяем последнее действие (C-/). Результат на рисунке (рис. 3)

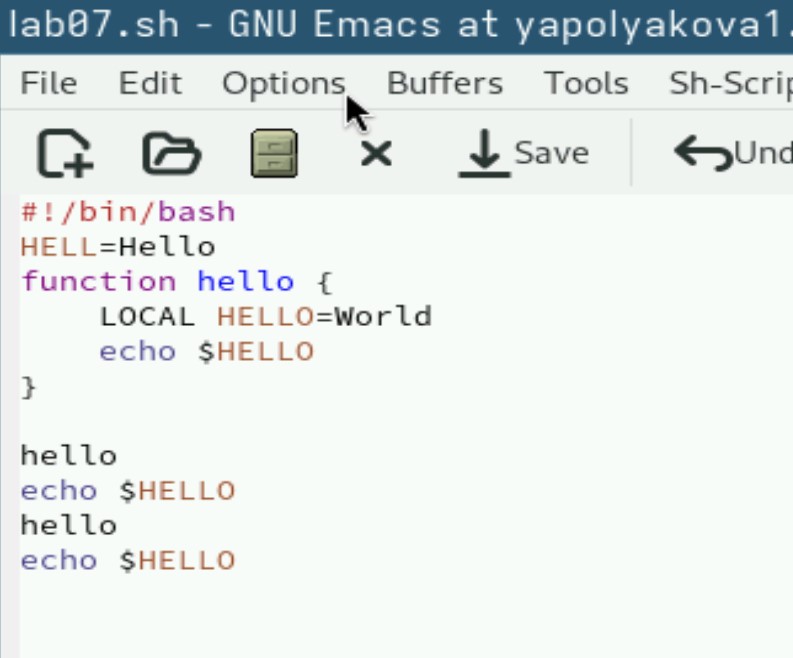


Рис. 3: Результат процедур редактирования текста

1. Учимся использовать команды по перемещению курсора. Перемещаем курсор в начало строки (C-a). Перемещаем курсор в конец строки (C-e). Перемещаем курсор в начало буфера (Alt-<). Перемещаем курсор в конец буфера (Alt->). (рис. 4)

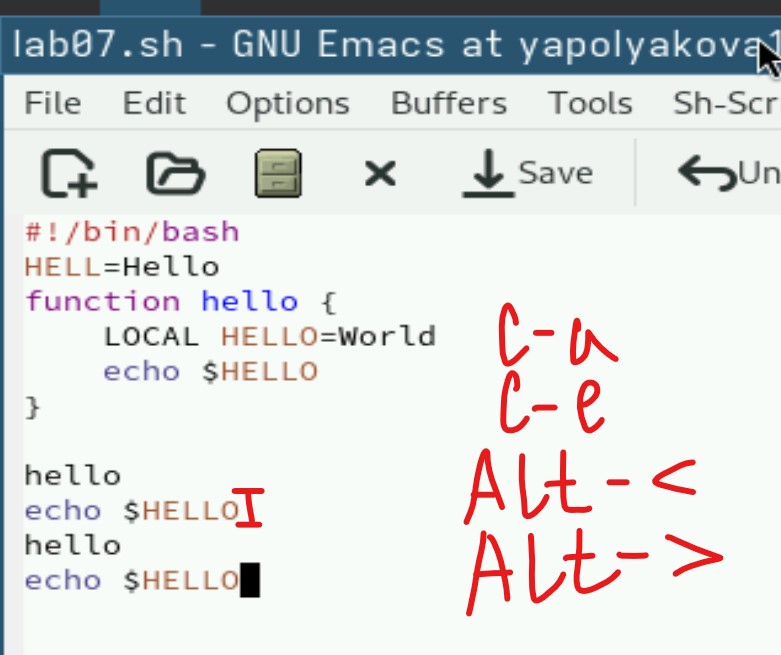


Рис. 4: Перемещение курсора клавишами

1. Выводим список активных буферов на экран (C-x C-b) (рис. 5)

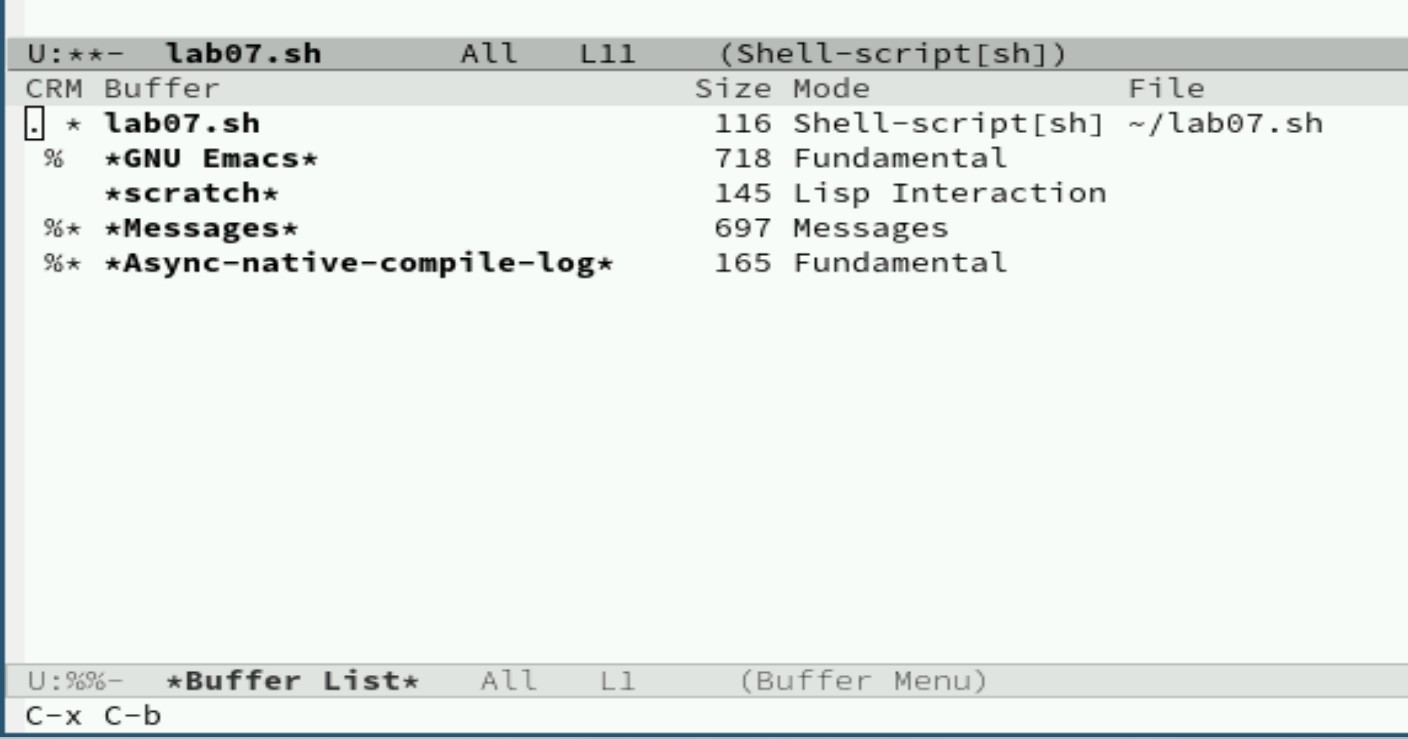


Рис. 5: Список активных буферов

1. Перемещаемся во вновь открытое окно (C-x o) со списком открытых буферов (рис. 6)

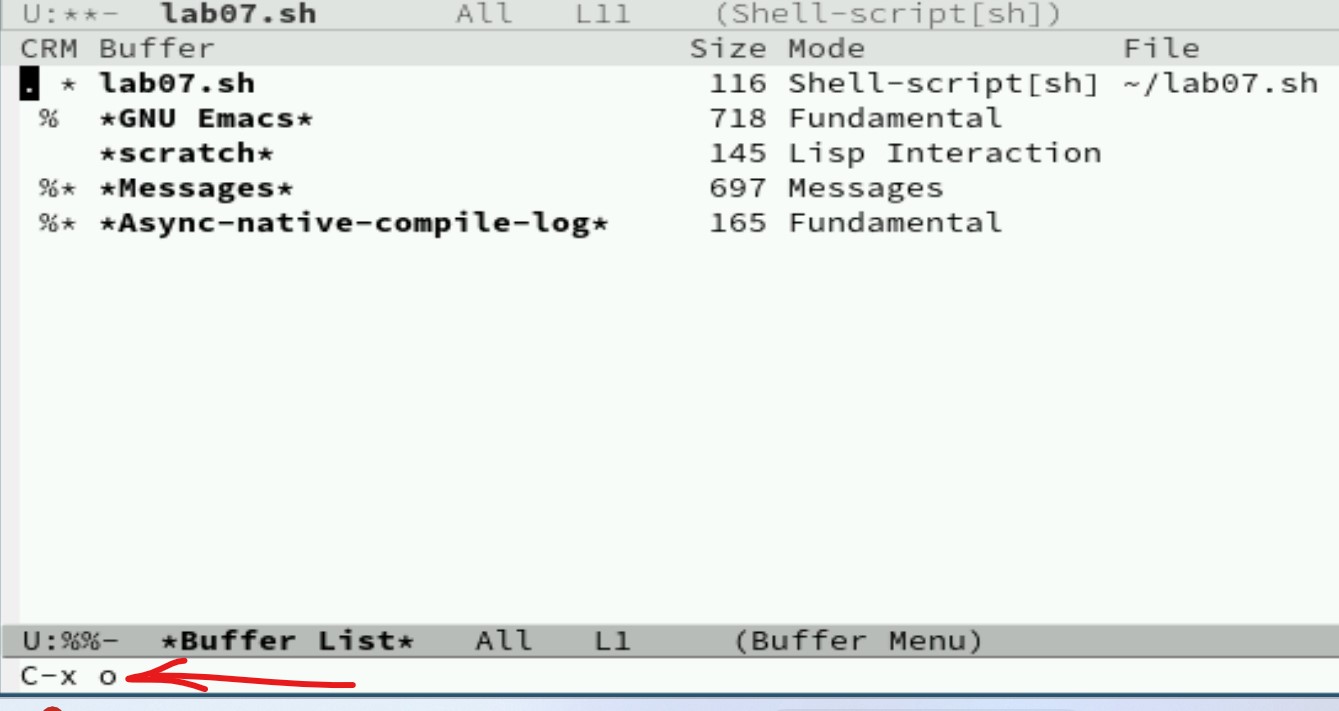


Рис. 6: Перемещаемся в открытое окно

1. Переключаемся на другой буфер (C-x b) (рис. 7)



Рис. 7: Другой буфер

1. Закрываем это окно (C-x 0) (рис. 8)



Рис. 8: Закрытие окна

1. Переключаемся между буферами, но уже без вывода их списка на экран (C-x b). (рис. 9)

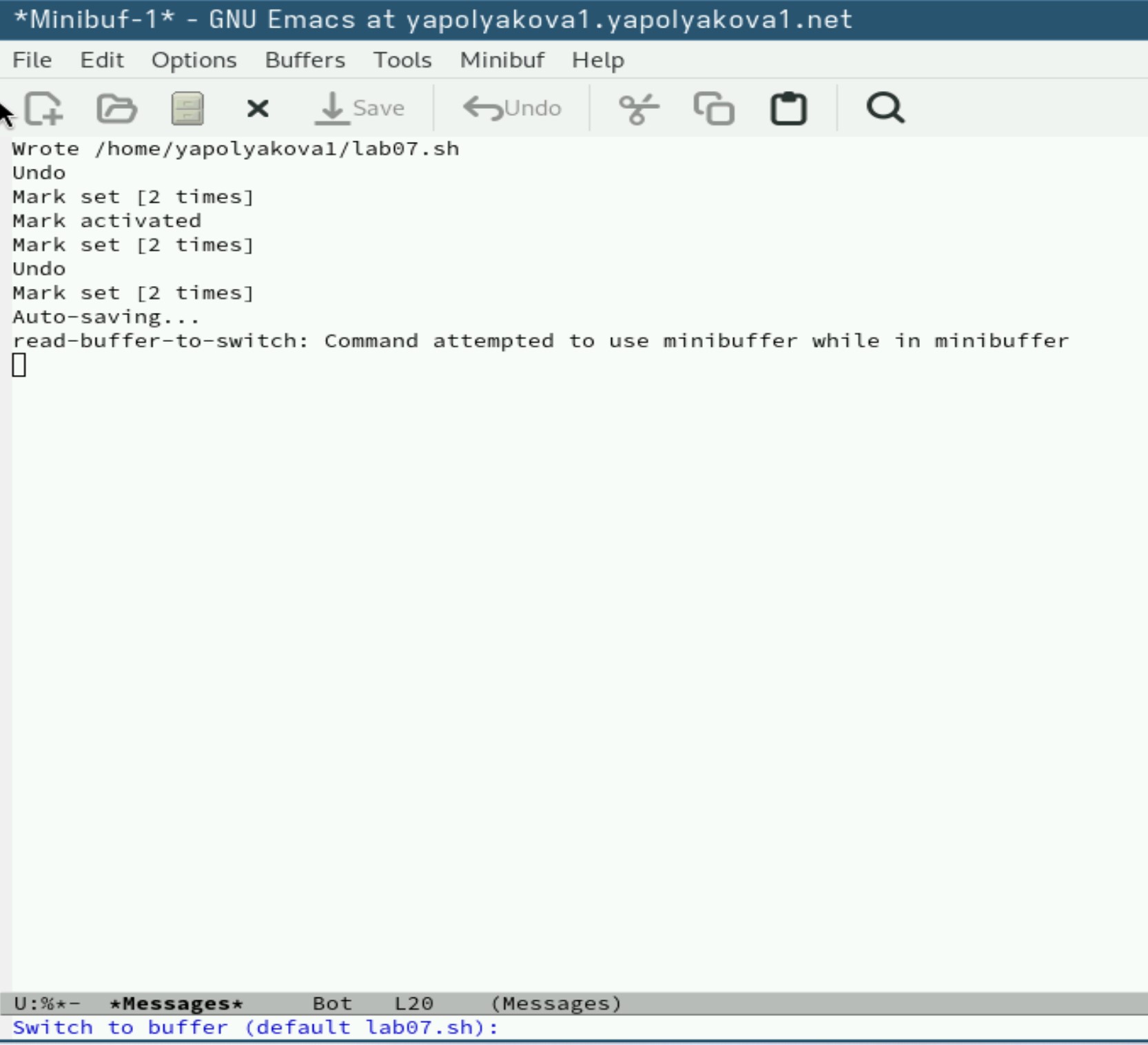


Рис. 9: Переключение без вывода списка

1. Делим фрейм на 4 части: делим на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2) (переключаемся между окнами C-x o). В каждом из четырёх созданных окон открываем новый буфер (C-x C-f) и вводим несколько строк текста. (рис. 10)

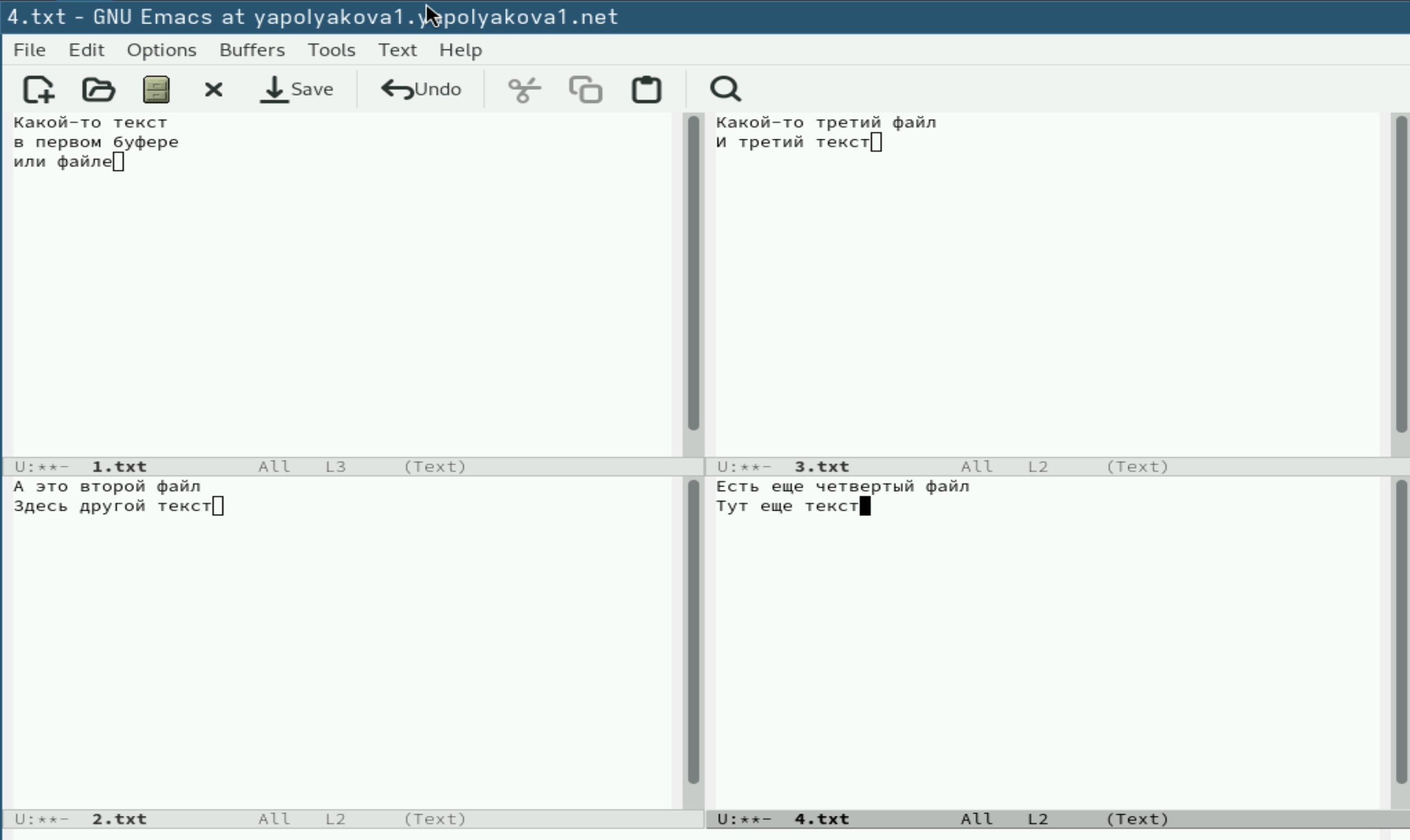


Рис. 10: Деление на 4 окна

1. Переключаемся в режим поиска (C-s) и ищем несколько слов, присутствующих в тексте. (рис. 11)

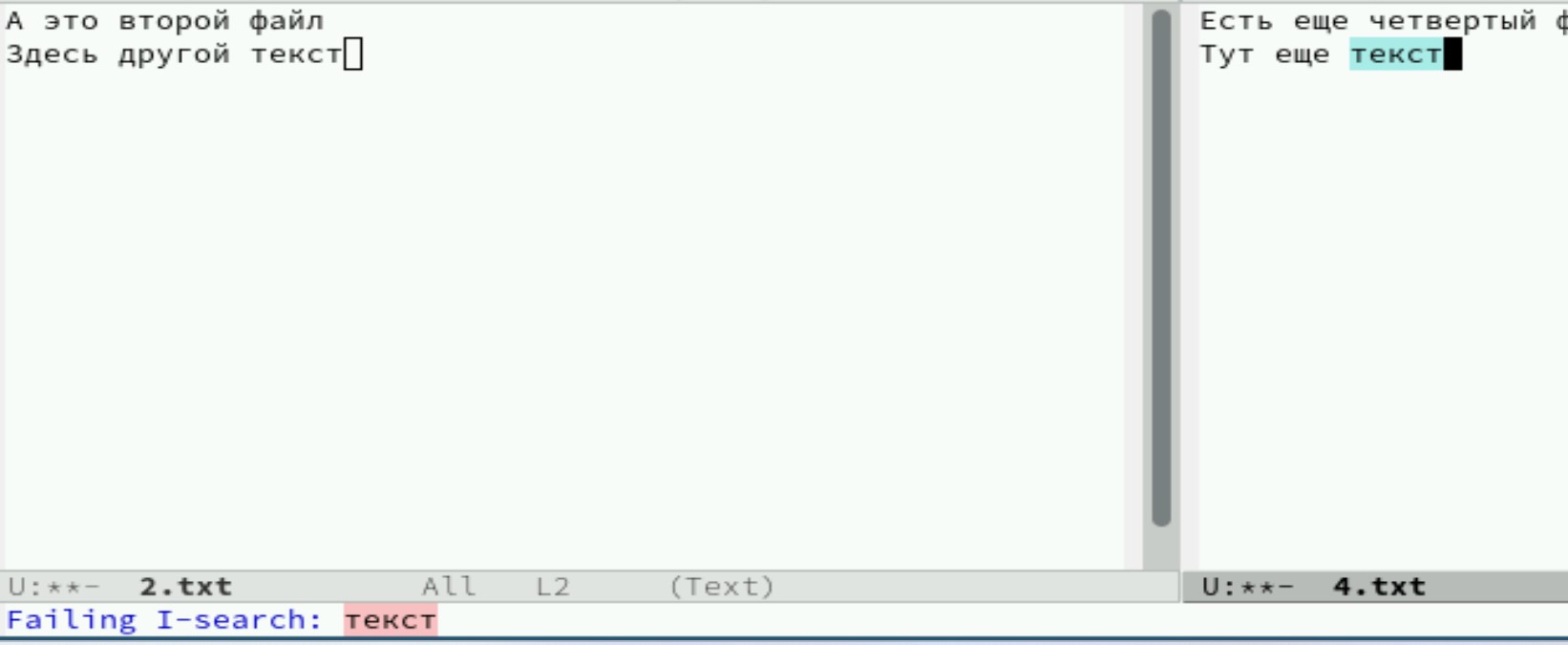


Рис. 11: Режим поиска

1. Переключаемся между результатами поиска, нажимая C-s. Выходим из режима поиска, нажав C-g (рис. 12)



Рис. 12: Переключение между результатами

1. Заменяем “еще” на “ещё”, используем режим поиска и замены (M-%), вводим текст, который следует найти и заменить, нажимаем Enter, затем вводим текст для замены. После того как будут подсвечены результаты поиска, нажимаем ! для подтверждения замены. (рис. 13)

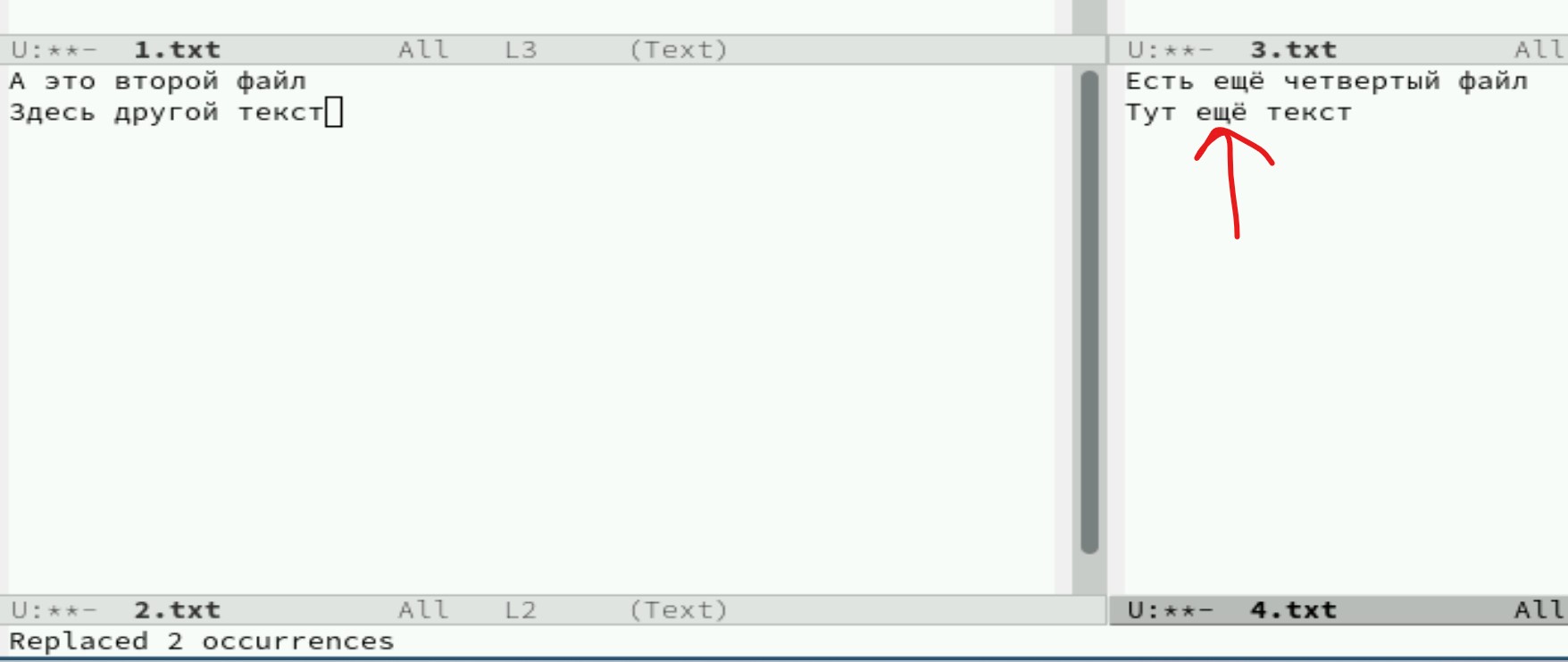


Рис. 13: Поиск с заменой

1. Пробуем другой режим поиска, нажав M-s o. Он отличается от предыдущих тем, что выводит результат поиска в отдельном окне. Он показывает кол-во совпадений, имя файла и место в файле (рис. 14)

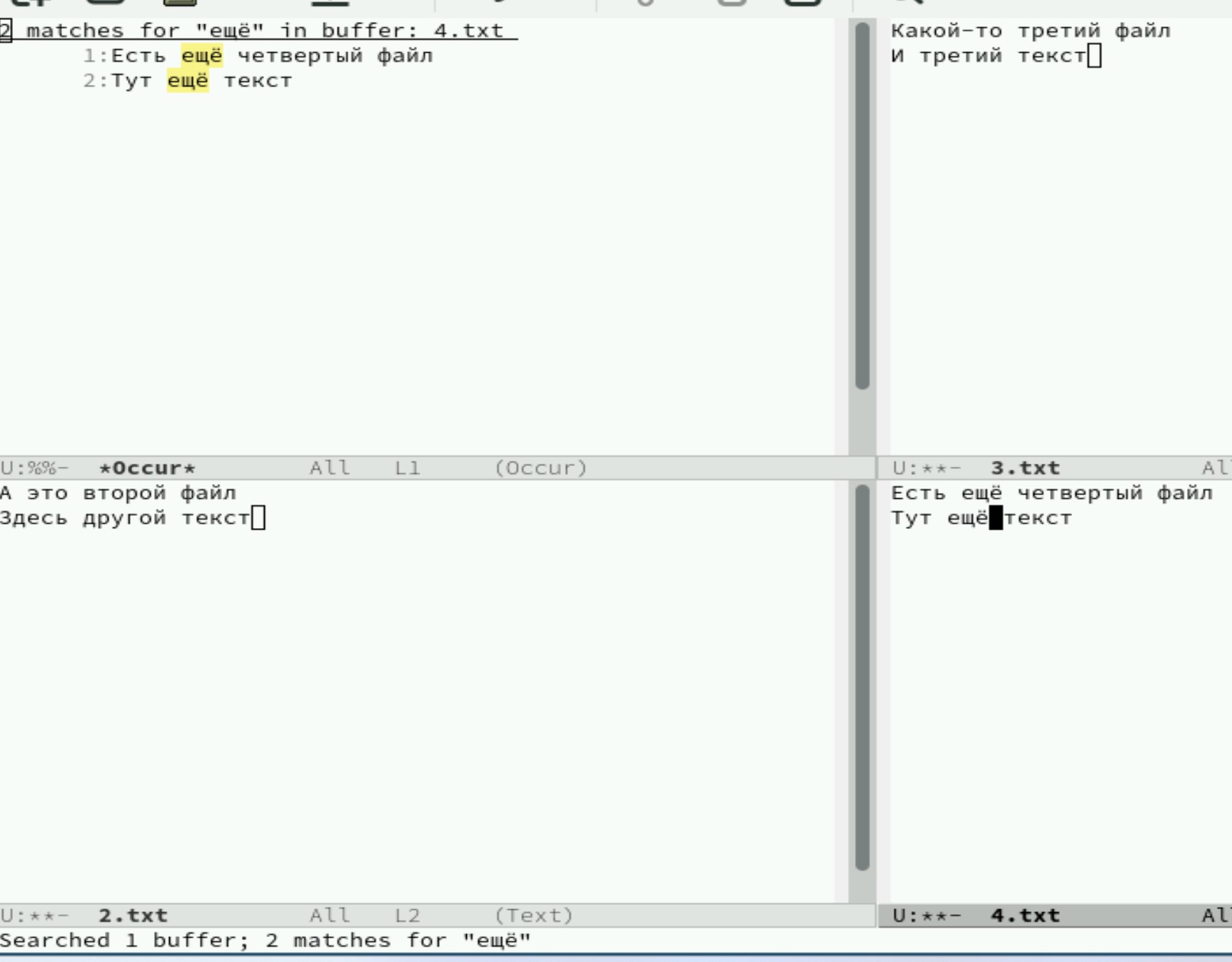


Рис. 14: Иной режим поиска

# 4 Контрольные вопросы

1. Краткая характеристика Emacs

Emacs — мощный расширяемый текстовый редактор с открытым исходным кодом, известный своей гибкостью и возможностями кастомизации. Поддерживает широкий спектр функций от простого редактирования текста до сложных операций программирования, организации рабочего процесса и даже электронной почты.

1. Особенности сложности для новичков:

* Многочисленные комбинации клавиш: основная функциональность требует запоминания множества сочетаний клавиш (C-x, C-s и др.).
* Строгая структура команд: некоторые команды требуют точного набора символов.
* Высокая степень настраиваемости: большое количество настроек и конфигураций делает начальное освоение непростой задачей.
* Необходимость изучения базовых концептов, таких как буферы, окна, режимы.

1. Буфер и окно в терминологии Emacs’а:

Буфер — это контейнер для хранения содержимого документа (текста), независимо от отображаемого окна. Окно же представляет собой область экрана, в которой отображается содержимое одного или нескольких буферов одновременно. Один буфер может отображаться сразу в нескольких окнах, а одно окно может показывать разные участки одного буфера.

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

Да, можно. Количество открытых буферов никак не ограничено количеством окон. Каждый буфер хранится отдельно, а окно лишь показывает определённый участок текущего активного буфера.

1. Какие буферы создаются по умолчанию при запуске Emacs?

При старте Emacs автоматически создаёт два основных буфера:

* \*scratch\*: специальный буфер для временного ввода и тестирования выражений Elisp.
* \*Messages\*: здесь выводятся информационные сообщения и предупреждения системы.

1. Комбинации клавиш:

Чтобы ввести последовательность C-c |, нажимаем сначала сочетание Ctrl+C, отпускаем, потом вводим символ вертикальной черты (“|”). Для последовательности C-c C-| выполняем аналогично, только вторую часть вводим удерживая Control и Shift: Ctrl+C → отпускаем → снова нажимаем Ctrl+Shift+“|”.

1. Разделение окна на две части:

Используя команду C-x 2 разделяет активное окно горизонтально пополам. Команда C-x 3 делит окно вертикально на две части.

1. Файл конфигурации Emacs:

Настройки сохраняются в файле .emacs.d/init.el. Этот файл находится в домашнем каталоге пользователя и хранит всю необходимую конфигурацию и расширения.

1. Клавиша Escape (или Meta):

Клавиша Esc (Meta) используется для запуска различных команд Emacs. Её часто применяют совместно с другими клавишами для активации специализированных действий. Эту клавишу можно переназначить путём изменения соответствующего макроса в настройках файла init.el.

1. Удобство Emacs vs Vi/Vim:

Выбор между Emacs и Vim зависит от личных предпочтений и целей использования. Например, если важна высокая степень настройки и возможность автоматизации процессов разработки, Emacs станет лучшим выбором благодаря своей мощной экосистеме плагинов и встроенному языку сценариев (Elisp). Если важнее скорость навигации по тексту и минимальные усилия на обучение клавиатурных сокращений, то Vim может показаться предпочтительнее. Оба инструмента имеют сильные стороны, и выбор идеального решения индивидуален. Удобнее мне показался emacs, так как он наиболее приближен к привычным редакторам, например, к Блокноту или gedit.

# 5 Вывод

Были получены практические навыки работы с редактором Emacs.