Отчёт по лабораторной работе №11

Операционные системы

Луангсуваннавонг Сайпхачан

Содержание

Список иллюстраций

Список таблиц

# 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 Теоретическое введение

Emacs (Editor MACroS) — один из самых гибких и настраиваемых текстовых редакторов, изначально разработанный Ричардом Столлманом в 1970-х годах в рамках проекта GNU. Он быстро вышел за рамки обычного редактора, превратившись в полноценную рабочую среду, включающую инструменты для программирования, написания документов, чтения почты, управления файлами и даже игр.

Основу архитектуры Emacs составляет язык Emacs Lisp — диалект Lisp, который позволяет пользователям настраивать, автоматизировать и расширять практически любой аспект редактора. Большая часть интерфейса и функций написана именно на Elisp, тогда как базовая система реализована на C.

Ключевые особенности:

Буферно-ориентированная работа — данные обрабатываются в буферах, которые могут содержать файлы, результаты команд, терминалы и другое.

Режимы (modes) — специальные режимы под разные типы задач: от org-mode для заметок до python-mode для программирования.

Интегрированная среда — поддержка компиляции, контроля версий (например, через Magit), терминала и отладки в рамках одного окна.

Несмотря на высокую кривую обучения (необычные сочетания клавиш и необходимость изучения Elisp), Emacs остаётся любимым инструментом разработчиков и продвинутых пользователей благодаря своей гибкости, расширяемости и философии «редактор, подстраивающийся под пользователя». Он объединяет принципы UNIX-инструментария в рамках одной мощной, программируемой среды.

# 4 Выполнение лабораторной работы

## 4.1 Основные команды emacs

Я устанавливаю и запускаю emacs через терминал (рис. 1)

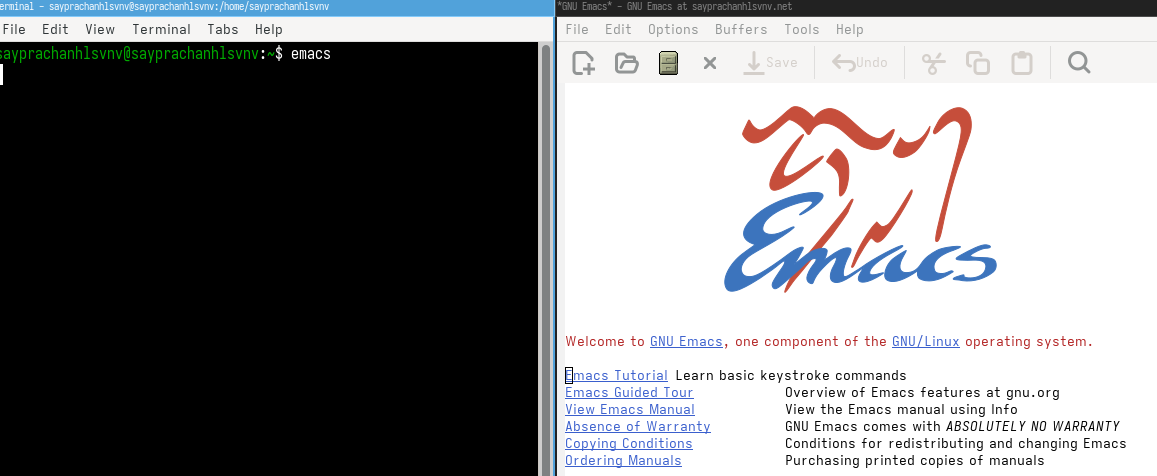


Рис. 1: Окно Emacs

Используя комбинацию клавиш Ctrl-x Ctrl-f (C-x C-f), я создаю файл lab07.sh (Поскольку он не находит файл, создается новый) (рис. 2)

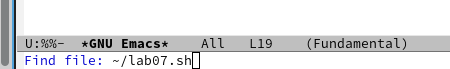


Рис. 2: Создание нового файла

Добавляю текстовый код в файл lab07.sh (рис. 3)

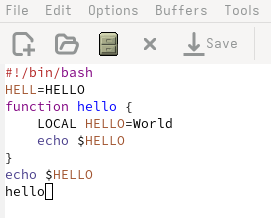


Рис. 3: Добавление текстового кода

Сохраняю файл lab07.sh комбинацией Ctrl-x Ctrl-s (C-x C-s) (рис. 4)

Рис. 4: Сохранение файла

Рис. 4: Сохранение файла

Вырезаю целую строку командой Ctrl-k (C-k) (рис. 5)

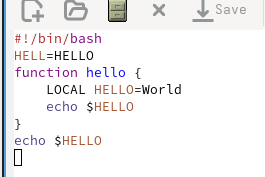


Рис. 5: Вырезание целой линии

Пытаюсь вырезать другую строку, затем вставляю её в конец файла lab07.sh командой Ctrl-y (C-y) (рис. 6)

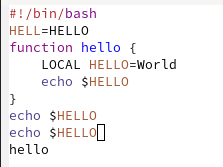


Рис. 6: Вставка строки

Для выделения области текста использую комбинацию Ctrl-space (C-space), затем копирую выделенный текст командой Alt-w или End-w (M-w) (рис. 7)

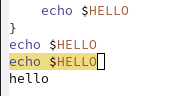


Рис. 7: Выделение текста и его копирование

Снова используя Ctrl-y (C-y), вставляю скопированный текст в конец файла (рис. 8)



Рис. 8: Вставка строки

Выделяю вставленный текст в конце файла и вырезаю его командой Ctrl-w (C-w) (рис. 9)

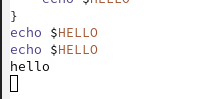


Рис. 9: Вырезание целой линии

Отменяю предыдущее действие комбинацией Ctrl-/ (C-/), и мы видим, что текст возвращается к исходному состоянию до вырезания (рис. 10 и рис. 11)

Рис. 10: Отмена последнего действия

Рис. 10: Отмена последнего действия



Рис. 11: Текстовый код

Перемещаю курсор в начало строки командой Ctrl-a (C-a) (рис. 12)

Рис. 12: Курсор в начале строки

Рис. 12: Курсор в начале строки

Затем перемещаю курсор в конец строки комбинацией Ctrl-e (C-e) (рис. 13)

Рис. 13: Курсор в конце строки

Рис. 13: Курсор в конце строки

Перемещаю курсор в начало буфера (файла) с помощью Alt-< или Esc-< (M-<) (рис. 14)

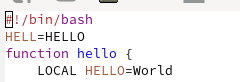


Рис. 14: Курсор в начале файла

Далее, используя Alt-> или Esc-> (M->), перемещаю курсор в конец файла (рис. 15)

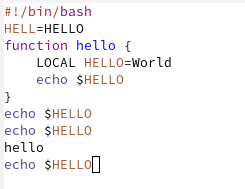


Рис. 15: Курсор в конце файла

## 4.2 Управление буферами

Отображаю список активных буферов на экране комбинацией Ctrl-x Ctrl-b (C-x C-b), затем переключаюсь между буфером lab07.sh и списком активных буферов командой Ctrl-x и буквой o (Ctrl-x o) (рис. 16)

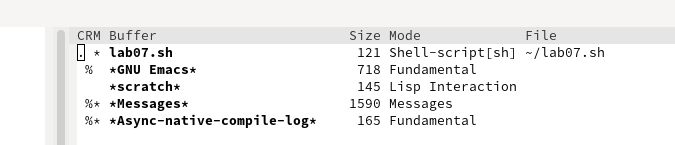


Рис. 16: Список активных буферов

Закрываю вкладку со списком буферов комбинацией Ctrl-x и цифрой 0 (Ctrl-x 0) (рис. 17)

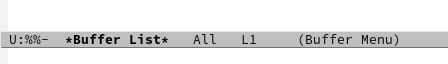


Рис. 17: Закрытие буфера

Переключаюсь между буферами без отображения их списка на экране с помощью Ctrl-x и буквы b (C-x b), выбираю нужный буфер (Buffer List) (рис. 18 и рис. 19)

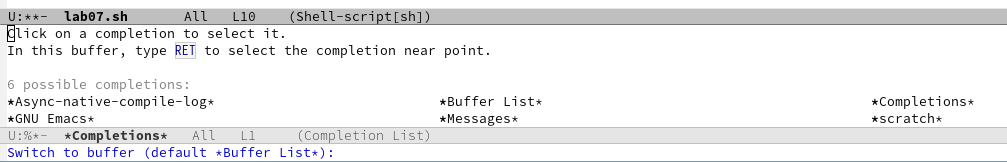


Рис. 18: Переключение на буфер

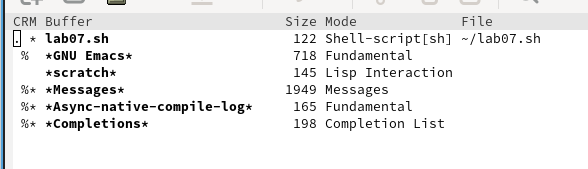


Рис. 19: Буфер Buffer List

## 4.3 Управление окнами

Возвращаюсь к буферу lab07.sh, затем разделяю фрейм на 4 части: комбинацией Ctrl-x с цифрой 3 (C-x 3) делю окна по вертикали, а с помощью Ctrl-x с цифрой 2 (C-x 2) — по горизонтали (рис. 20)



Рис. 20: Разделение буферного кадра

Открываю новый буфер в каждом из четырех созданных окон и ввожу в них текст (рис. 21)



Рис. 21: Создание новых буферов

## 4.4 Режим поиска

Переключаюсь в режим поиска командой Ctrl-s (C-s), нахожу несколько слов в тексте, также добавляю еще текст для дополнительного поиска (рис. 22 и рис. 23)



Рис. 22: Режим поиска



Рис. 23: Режим поиска

Переключаюсь между результатами поиска комбинацией Ctrl-S (C-S) (рис. 24)



Рис. 24: Переключение между результатами

Выхожу из режима поиска командой Ctrl-g (C-g) (рис. 25)

Рис. 25: Выход из режима поиска

Рис. 25: Выход из режима поиска

Активирую режим поиска и замены комбинацией Alt-% или Esc-% (M-%), ввожу текст для поиска и замены, нажимаю Enter для применения изменений (рис. 26)

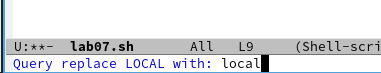


Рис. 26: Режим поиска и замены

Проверяю результат — видно, что слово заменилось на новое (я заменил слово ‘LOCAL’ на ‘local’) (рис. 27)

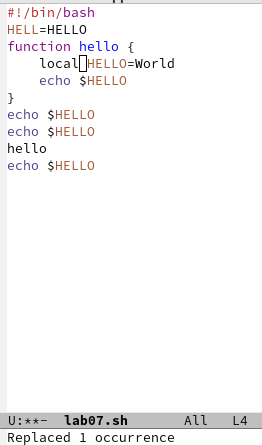


Рис. 27: Замена слова

Пробую другой режим поиска: Alt-s или Esc-s с буквой o (M-s o), ввожу искомый текст, и результаты отображаются в отдельном окне (в отличие от обычного поиска) (рис. 28)

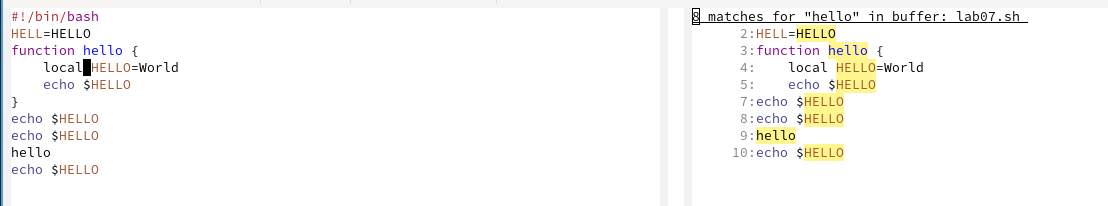


Рис. 28: Другой режим поиска

# 5 Выводы

Я познакомился с операционной системой Linux и получил практические навыки работы с редактором Emacs

# 6 Ответы на контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — это мощный, расширяемый текстовый редактор с большим количеством функций, включая подсветку синтаксиса, работу с файлами, управление буферами, встроенный терминал, поддержку языков программирования и возможность кастомизации.

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

Необычные сочетания клавиш (например, C-x C-s для сохранения).

Модальность (разные режимы ввода и команд).

Огромное количество функций, которые могут перегружать пользователя.

Настройка через Lisp (может быть сложной для непрограммистов).

1. Своими словами опишите, что такое буфер и окно в терминологии emacs’а.

Буфер — это объект Emacs, представляющий открытый файл, текстовые данные или процесс (например, терминал). Окно — это область на экране, в котором отображается буфер. В одном графическом окне Emacs может быть несколько внутренних окон (фреймов).

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

Да, мы можем открыть столько буферов, сколько захотим, но только некоторые из них отображаются в одном окне за раз (в зависимости от разбивки на подокна).

1. Какие буферы создаются по умолчанию при запуске emacs?

При запуске создаются буферы:

\*scratch\* (временный буфер для тестирования кода)

\*Messages\* (лог системных сообщений)

\*GNU Emacs\* (справка или приветствие)

1. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?

Для ввода:

C-c | → Зажать Ctrl, нажать c, отпустить, затем нажать | (Shift + ).

C-c C-| → Зажать Ctrl, нажать c, затем, не отпуская Ctrl, нажать |.

1. Как поделить текущее окно на две части?

Поделить окно:

Горизонтально: C-x 2 (Ctrl + x, затем 2).

Вертикально: C-x 3 (Ctrl + x, затем 3)

1. В каком файле хранятся настройки редактора emacs?

Настройки хранятся в файле ~/.emacs или ~/.emacs.d/init.el.

1. Какую функцию выполняет клавиша (<-) и можно ли её переназначить?

Клавиша <- (Backspace) обычно удаляет символ перед курсором. Её можно переназначить через настройки (например, в .emacs).

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

Emacs удобен для тех, кто любит кастомизацию, работу с разными режимами и Lisp.

Vi/Vim проще для быстрого редактирования благодаря модальности.

Если привыкнуть к сочетаниям клавиш, Emacs может быть удобнее из-за гибкости, но Vim быстрее для простого редактирования.

# Список литературы

[Лабораторная работа №11](https://esystem.rudn.ru/pluginfile.php/2586874/mod_resource/content/5/009-lab_emacs.pdf)