

Отчёт по лабораторной работе №11

Текстовый редактор emacs

Чекмарев Александр Дмитриевич | Группа НПИбд-02-23

Содержание

1	Цель работы	5
2	Задания	6
3	Выполнение лабораторной работы	7
3.1	Основные команды etacs	7
4	Контрольные вопросы	16
5	Выводы	18
	Список литературы	19

Список иллюстраций

3.1	Рис 1.1.1: команда для открытия	7
3.2	Рис 1.2.1: создание файла	7
3.3	Рис 1.3.1: введенный текст	8
3.4	Рис 1.5.1: вырезанная строка	8
3.5	Рис 1.5.2: вставка строки	8
3.6	Рис 1.5.3: выделение области	8
3.7	Рис 1.5.4: вставка текста	9
3.8	Рис 1.5.5: вырезанная область	9
3.9	Рис 1.5.6: отмена действия	9
3.10	Рис 1.7.1: активные буферы	10
3.11	Рис 1.7.2: другой буфер	10
3.12	Рис 1.7.3: переключение буфера	11
3.13	Рис 1.8.1: разделение на части	12
3.14	Рис 1.8.2: демонстрация введенного текста	13
3.15	Рис 1.9.1: переключение буфера	14
3.16	Рис 1.9.2: выделенное слово после действий команд	15
3.17	Рис 1.9.3: замена	15
3.18	Рис 1.9.4: демонстрация действия другого режима поиска	15

Список таблиц

1 Цель работы

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

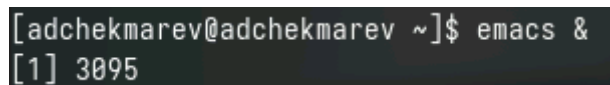
2 Задания

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

3 Выполнение лабораторной работы

3.1 Основные команды emacs

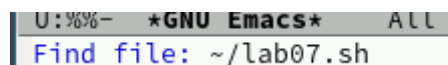
1. Откроем emacs.



```
[adchekmarev@adchekmarev ~]$ emacs &  
[1] 3095
```

Рис. 3.1: Рис 1.1.1: команда для открытия

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



```
U:%%~ *GNU Emacs* ALL  
Find file: ~/lab07.sh
```

Рис. 3.2: Рис 1.2.1: создание файла

3. Наберем текст:

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



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

Рис. 3.3: Рис 1.3.1: введенный текст

4. Сохраним файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
5. Прделаем с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.

Вырежем одной командой целую строку (C-k).

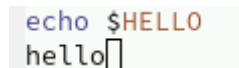


```
echo $HELLO

```

Рис. 3.4: Рис 1.5.1: вырезанная строка

Вставим эту строку в конец файла (C-y).



```
echo $HELLO
hello
```

Рис. 3.5: Рис 1.5.2: вставка строки

Выделим область текста (C-space).



```
echo $HELLO
}
echo $HELLO
hello
```

Рис. 3.6: Рис 1.5.3: выделение области

Скопируем область в буфер обмена (M-w). Вставим область в конец файла.


```
}  
echo $HELLO  
hello  
}  
echo $HELLO  
hello
```

Рис. 3.7: Рис 1.5.4: вставка текста

Вновь выделим эту область и на этот раз вырезать её (C-w).

```
}  
echo $HELLO  
hello  
█
```

Рис. 3.8: Рис 1.5.5: вырезанная область

Отменим последнее действие (C-/).

```
}  
echo $HELLO  
hello  
█  
echo $HELLO  
hello
```

Рис. 3.9: Рис 1.5.6: отмена действия

6. Научимся использовать команды по перемещению курсора

Переместим курсор в начало строки (C-a)

Переместим курсор в конец строки (C-e)

Переместим курсор в начало буфера (M-<)

Переместим курсор в конец буфера (M->)

7. Управление буферами

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

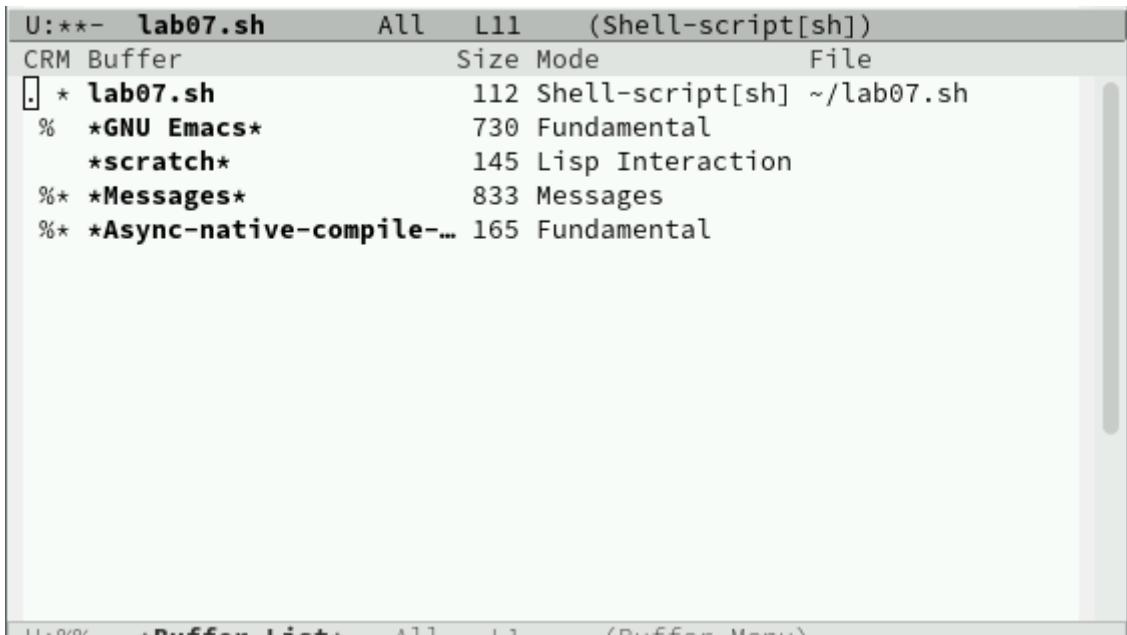


Рис. 3.10: Рис 1.7.1: активные буферы

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

Закроем это окно (C-x 0).



Рис. 3.11: Рис 1.7.2: другой буфер

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

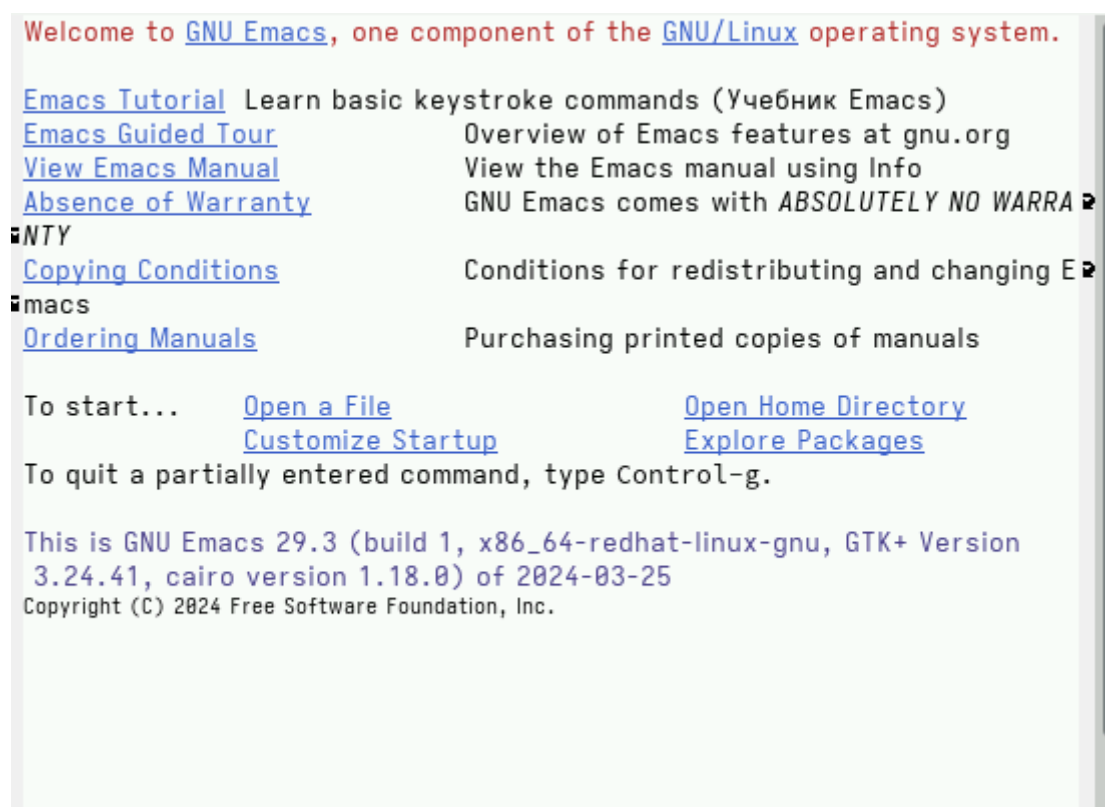


Рис. 3.12: Рис 1.7.3: переключение буфера

8. Управление окнами

Поделим фрейм на 4 части: разделите фрейм на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2).

CRM	Buffer	Size	CRM	Buffer	Size
%	*GNU Emacs*	→	%	*GNU Emacs*	→
*	lab07.sh	→	*	lab07.sh	→
	scratch	→		*scratch*	→
%*	*Messages*	→	%*	*Messages*	→
%*	*Async-native-compile-...	→	%*	*Async-native-compile-...	→
U:%%- *Buffer List* All L6			U:%%- *Buffer List* All L6		
CRM	Buffer	Size	CRM	Buffer	Size
%	*GNU Emacs*	→	%	*GNU Emacs*	→
*	lab07.sh	→	*	lab07.sh	→
	scratch	→		*scratch*	→
%*	*Messages*	→	%*	*Messages*	→
%*	*Async-native-compile-...	→	%*	*Async-native-compile-...	→

Рис. 3.13: Рис 1.8.1: разделение на части

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

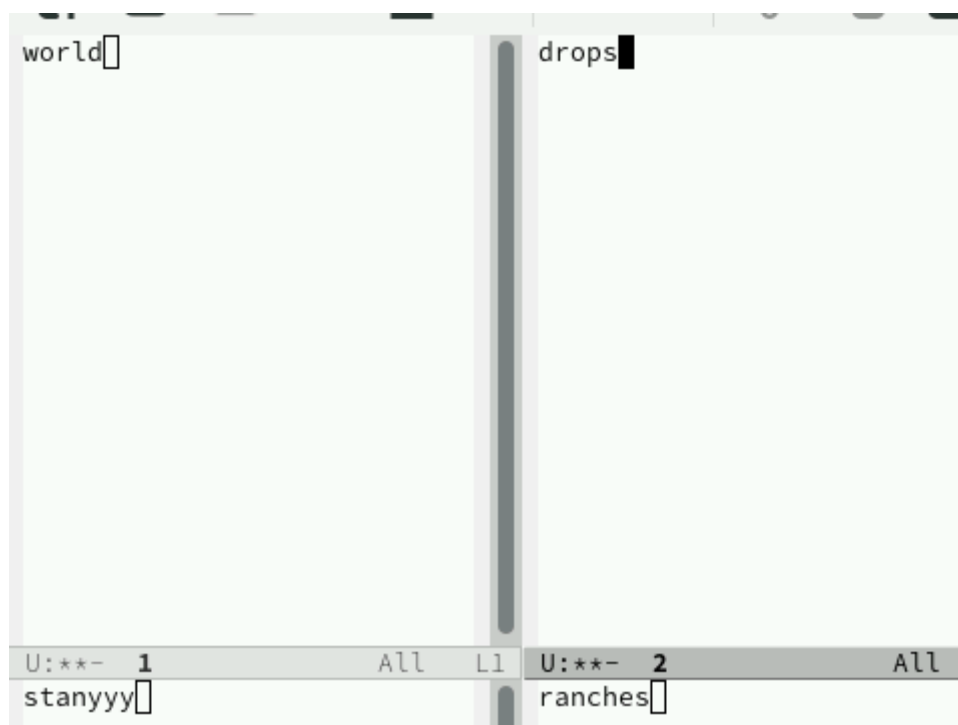


Рис. 3.14: Рис 1.8.2: демонстрация введенного текста

9. Режим поиска

Переключимся в режим поиска (C-s) и найдем несколько слов, присутствующих в тексте.

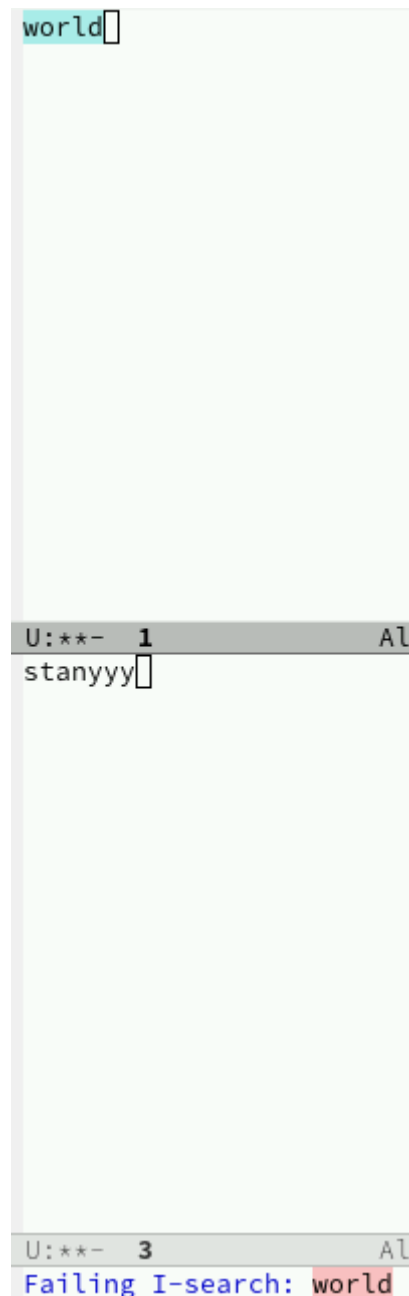


Рис. 3.15: Рис 1.9.1: переключение буфера

Переключимся между результатами поиска, нажимая C-s.

Выйдем из режима поиска, нажав C-g.

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.

Заменим drops на hello

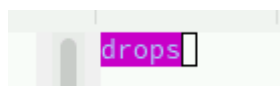


Рис. 3.16: Рис 1.9.2: выделенное слово после действий команд

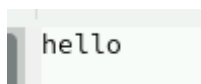


Рис. 3.17: Рис 1.9.3: замена

Попробуем другой режим поиска, нажав M-s o

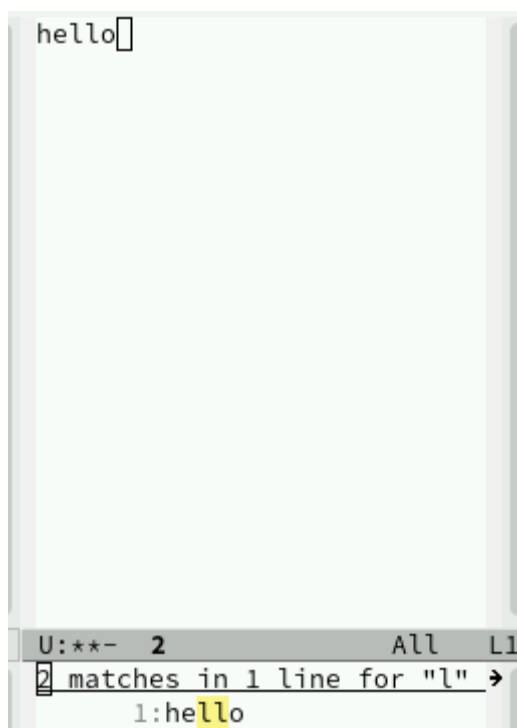


Рис. 3.18: Рис 1.9.4: демонстрация действия другого режима поиска

4 Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — это мощный текстовый редактор, который также может быть использован как среда разработки благодаря своей расширяемости. Он имеет множество функций, таких как расширяемая система управления буферами, встроенный интерпретатор языка Emacs Lisp и многое другое.

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

Сложность Emacs для новичка может возникнуть из-за его нетрадиционного подхода к управлению текстом. Он использует множество комбинаций клавиш, что может потребовать времени для изучения. Также его уникальная философия и обилие функций могут быть перегрузкой для новичка.

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

В терминологии Emacs буфер представляет собой область памяти, в которой хранится текст. Окно — это видимая часть буфера, которая отображается на экране. Окно может отображать только один буфер за раз, но один буфер может быть открыт в нескольких окнах.

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

В Emacs можно открыть больше 10 буферов в одном окне. Ограничений на количество буферов в одном окне нет.

5. Какие буферы создаются по умолчанию при запуске emacs?

При запуске Emacs по умолчанию создаются несколько буферов, таких как `scratch` (для экспериментов и временного кодирования), `Messages` (для вывода сообщений о работе Emacs), `Minibuffer` (для ввода команд) и другие, в зависимости от конфигурации.

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

Для ввода комбинаций C-c | и C-c C-| нажмите соответственно `Ctrl + c`, затем нажмите вертикальную черту `|`. Аналогично для комбинации C-c C-|, но вместо вертикальной черты используйте `Shift +`

7. Как поделить текущее окно на две части?

Чтобы поделить текущее окно на две части, можно использовать команду `split-window-horizontally` (обычно это `Ctrl + x 2`). Это создаст два окна, расположенных горизонтально друг над другом.

8. В каком файле хранятся настройки редактора emacs?

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

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

Клавиша `backspace` в Emacs выполняет функцию удаления символа перед курсором. Её можно переназначить с помощью конфигурации Emacs Lisp.

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

`gedit`. Потому что он похож на `notepad++`, которым я зачастую пользуюсь на windows

5 Выводы

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

Список литературы