

# **Отчет по лабораторной работе №9**

**Операционные системы**

Бембо лумингу жозе

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>19</b>
<b>6</b>	<b>Ответы на контрольные вопросы</b>	<b>20</b>

## Список иллюстраций

4.1	Открытие программы . . . . .	9
4.2	Создание файла . . . . .	9
4.3	Редактирование файла . . . . .	10
4.4	Сохранение изменений в файле . . . . .	10
4.5	Вырезание строки . . . . .	10
4.6	Вставка строки в конце файла . . . . .	10
4.7	Вырезанная область . . . . .	11
4.8	Вставка в конец файла . . . . .	11
4.9	Отмена последнего действия . . . . .	12
4.10	Курсор в начале строки . . . . .	12
4.11	Курсор в конце строки . . . . .	12
4.12	Начало буфера . . . . .	13
4.13	Конец буфера . . . . .	13
4.14	Список активных буферов . . . . .	13
4.15	Другое окно буфера . . . . .	14
4.16	Закрытие буфера . . . . .	14
4.17	Открытие другого буфера . . . . .	15
4.18	Четыре окна . . . . .	15
4.19	Открытие буферов в четырех окнах . . . . .	16
4.20	Режим поиска . . . . .	16
4.21	Перемещение по найденным выражениям . . . . .	17
4.22	Выход из режима поиска . . . . .	17
4.23	Замена слова . . . . .	17
4.24	Слово заменено . . . . .	18
4.25	Режим поиска . . . . .	18

## **Список таблиц**

# 1 Цель работы

Цель данной лабораторной работы - познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

## 2 Задание

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

## 3 Теоретическое введение

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. По популярности он соперничает с редактором vi и его клонами. В зависимости от ситуации, Emacs может быть:

- текстовым редактором;
- программой для чтения почты и новостей Usenet;
- интегрированной средой разработки (IDE);
- операционной системой;

Всё это разнообразие достигается благодаря архитектуре Emacs, которая позволяет расширять возможности редактора при помощи языка Emacs Lisp. На языке C написаны лишь самые базовые и низкоуровневые части Emacs, включая полнофункциональный интерпретатор языка Lisp. Таким образом, Emacs имеет встроенный язык программирования, который может использоваться для настройки, расширения и изменения поведения редактора. В действительности, большая часть того редактора, с которым пользователи Emacs работают в наши дни, написана на языке Lisp.

Первая версия редактора Emacs была написана в 70-х годах 20-го столетия Richard Stallman (Ричардом Столманом) как набор макросов для редактора TECO. В дальнейшем, уже будучи основателем Фонда Свободного программного обеспечения Free Software Foundation и проекта GNU, Stallman разработал GNU Emacs в развитие оригинального Emacs и до сих пор сопровождает эту программу. Emacs является одним из старейших редакторов. Он использовался тысячами

программистов на протяжении последних 20 с лишним лет, для него создано много дополнительных пакетов расширений. Эти дополнения позволяют делать с помощью Emacs такие вещи, которые Stallman , вероятно, даже не считал возможными в начале своей работы над редактором.



## 4 Выполнение лабораторной работы

Открываю Emacs через терминал (рис. fig. 4.1).

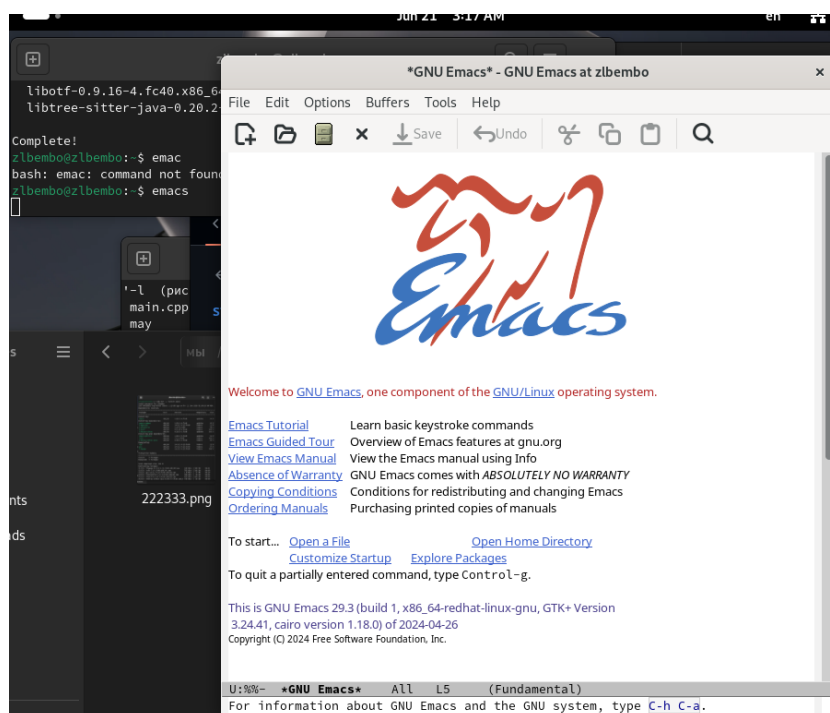


Рис. 4.1: Открытие программы

Создаю файл ab07.sh с помощью комбинации Ctrl-x Ctrl-f (рис. fig. 4.2).

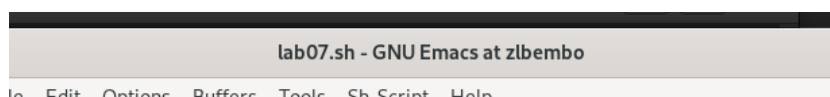


Рис. 4.2: Создание файла

Прописываю в файле текст программы (рис. fig. 4.3).

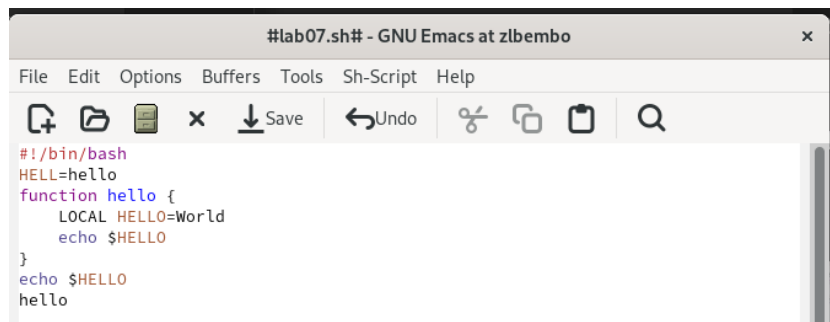


Рис. 4.3: Редактирование файла

Сохраняю файл с помощью комбинации C-x C-s (рис. fig. 4.4).

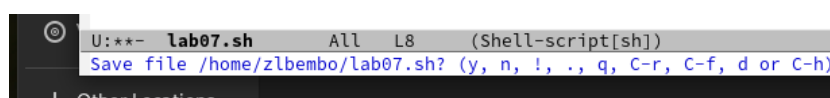


Рис. 4.4: Сохранение изменений в файле

Вырезаю одной командой целую строку (C-k) (рис. fig. 4.5).

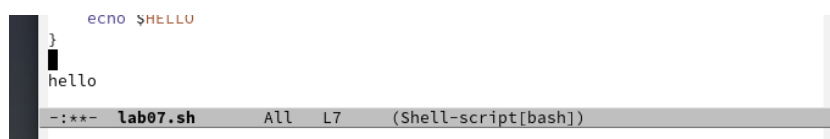


Рис. 4.5: Вырезание строки

Вставляю эту строку в конец файла (C-y) (рис. fig. 4.6).

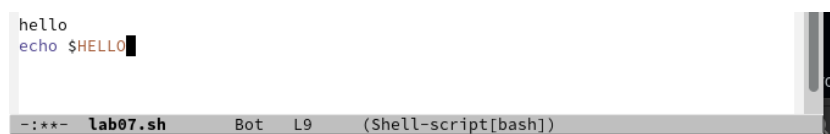


Рис. 4.6: Вставка строки в конце файла

Выделяю область текста (C-space), копирую область в буфер обмена (M-w), вырезаю эту область с помощью C-w (рис. fig. 4.7).

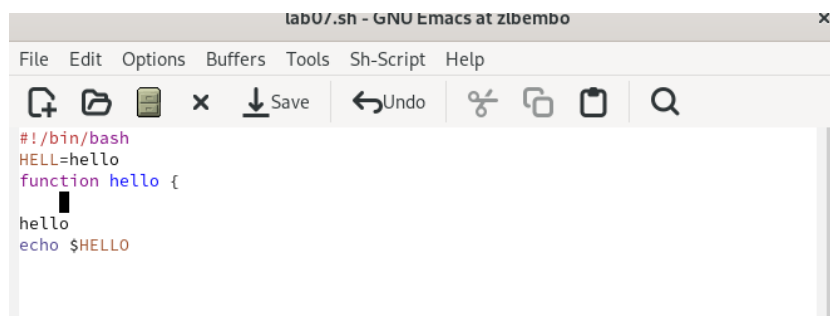


Рис. 4.7: Вырезанная область

Вставляю область в конец файла (C-y)(рис. fig. 4.8).

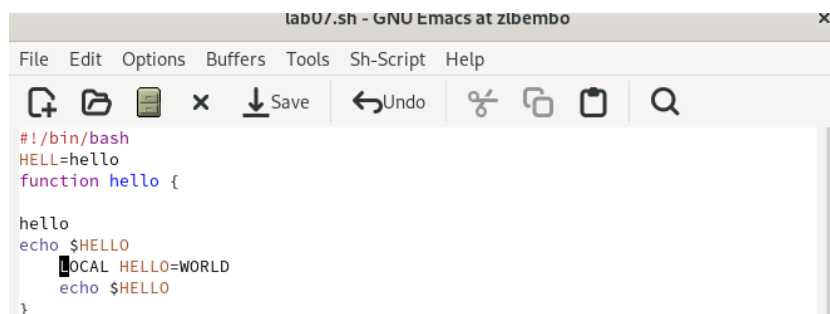


Рис. 4.8: Вставка в конец файла

Отменяю последнее действие C-/ (рис. fig. 4.9).

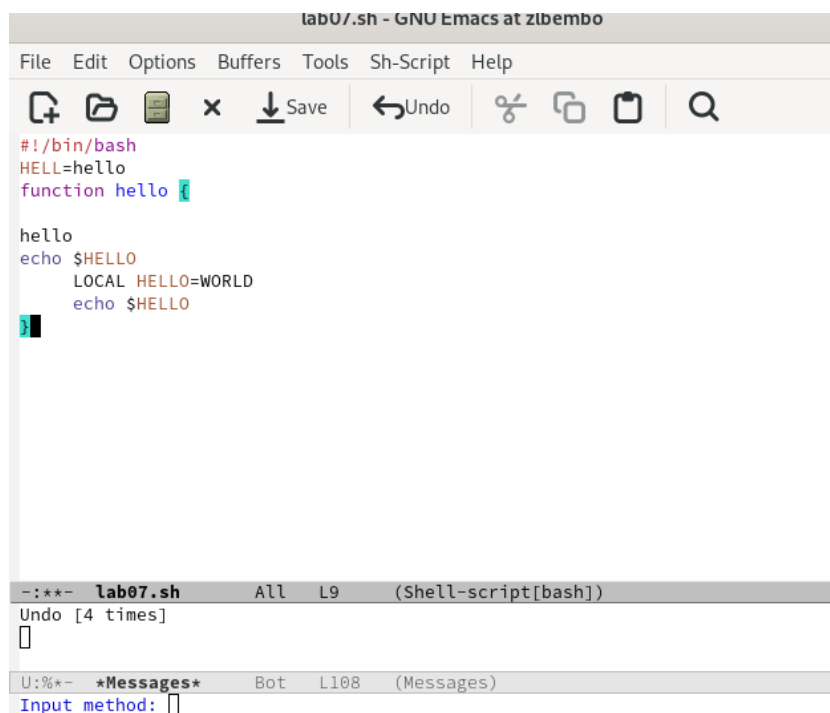


Рис. 4.9: Отмена последнего действия

Перевожу курсор в начало строки C-a (рис. fig. 4.10).

```

hello
echo $HELLO
  LOCAL_HELLO=WORLD
  echo $HELLO
}

```

Рис. 4.10: Курсор в начале строки

Перемещаю курсор в конец строки C-e (рис. fig. 4.11).

```

hello
echo $HELLO
  LOCAL_HELLO=WORLD
  echo $HELLO
}

```

Рис. 4.11: Курсор в конце строки

Перемещаю курсор в начало файла M-< (рис. fig. 4.12).

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

Рис. 4.12: Начало буфер

Перемещаю курсор в конец файла М->(рис. fig. 4.13).

```
#!/bin/b.ash
HELL=hello
funCtion hello █

hello
echo $HELLO
LOCAL HELLO=WORLD
echo $HELLO
█
```

Рис. 4.13: Конец буфера

Открываю список активных буферов в другом окне С-х С-b (рис. fig. 4.14).

-:*** lab07.sh		All	L8	(Shell-script[bash])	
CRM	Buffer		Size	Mode	File
█	* lab07.sh		105	Shell-script[ba...	~/lab07.sh
%	*GNU Emacs*		718	Fundamental	
	*scratch*		145	Lisp Interaction	
%*	*Messages*		2559	Messages	
%*	*Async-native-compile...		165	Fundamental	

Рис. 4.14: Список активных буферов

Переключаюсь на другой буфер С-х о (рис. fig. 4.15).

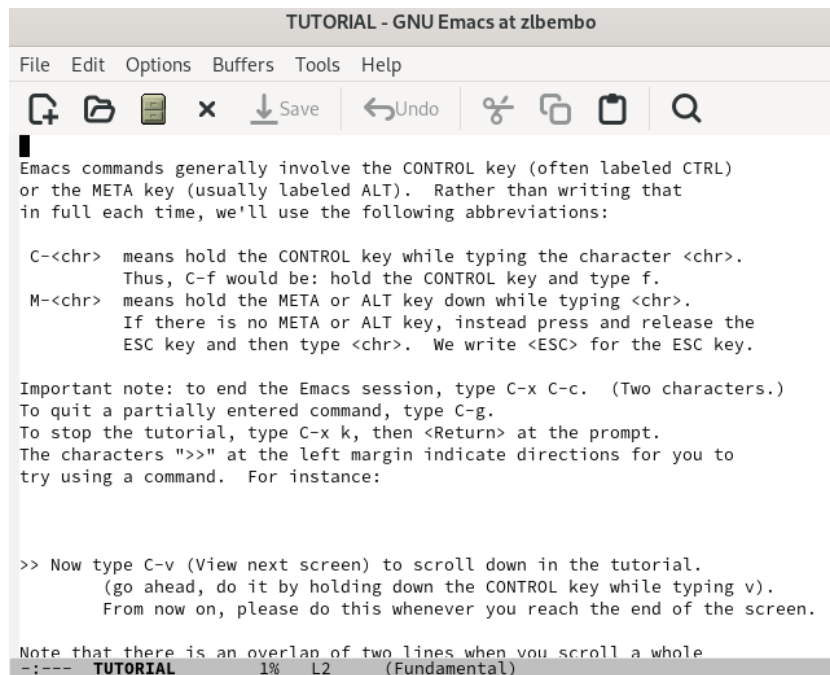


Рис. 4.15: Другое окно буфера

Закрываю окно другого буфера C-x 0 (рис. fig. 4.16).

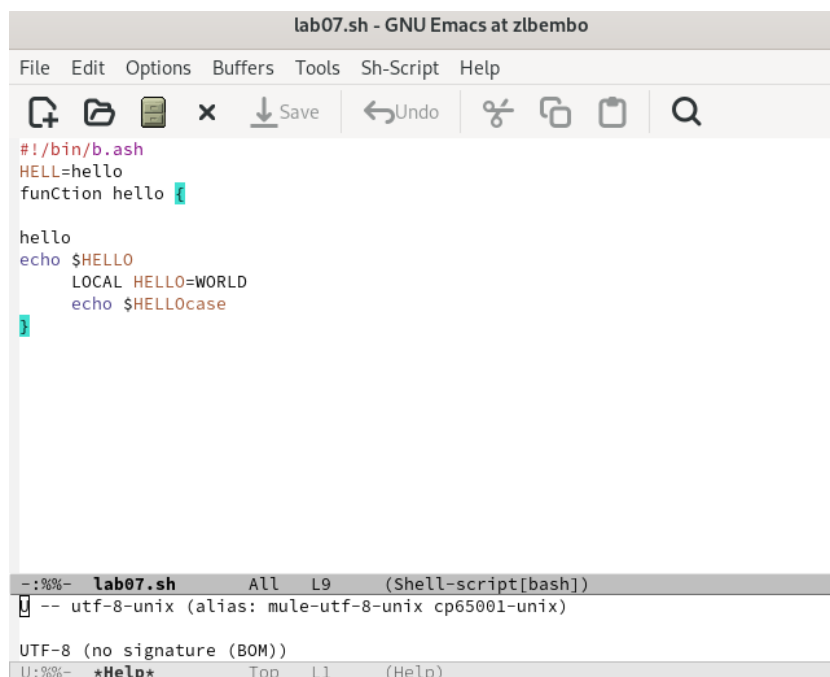


Рис. 4.16: Заккрытие буфера

Открываю другой буфер без вывода их списка на экран с помощью C-x b(рис.

fig. 4.17).

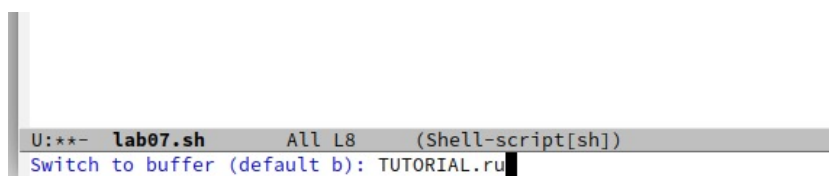


Рис. 4.17: Открытие другого буфера

Делю фрейм на 4 части: сначала на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2) (рис. fig. 4.18).

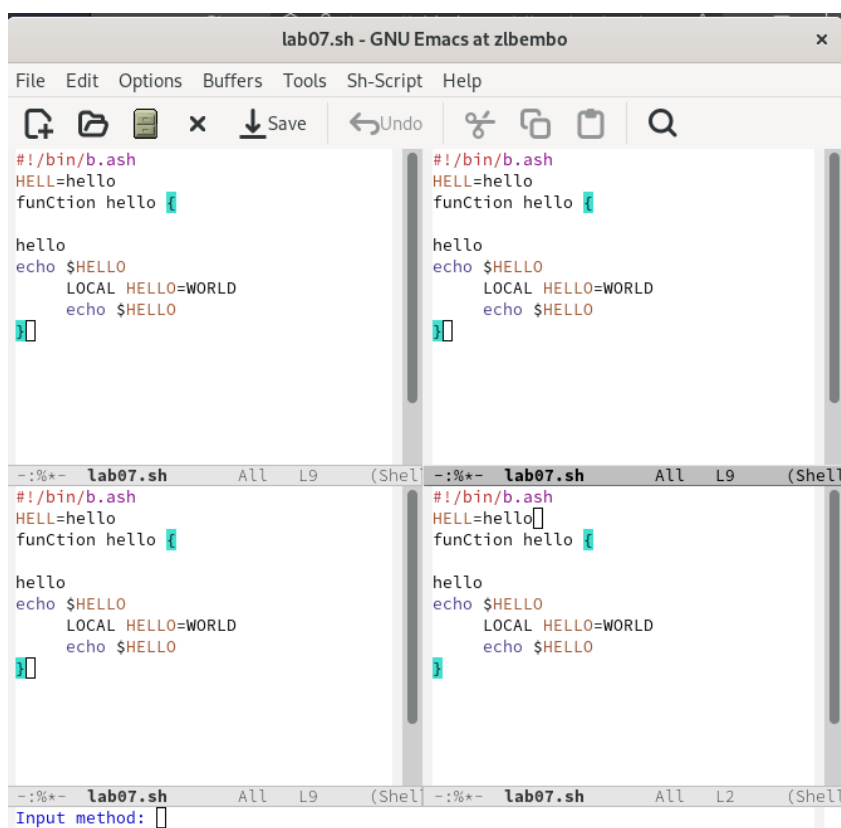


Рис. 4.18: Четыре окна

В каждом из четырех созданных окон с помощью С-х b открыла разные буферы и (рис. fig. 4.19).

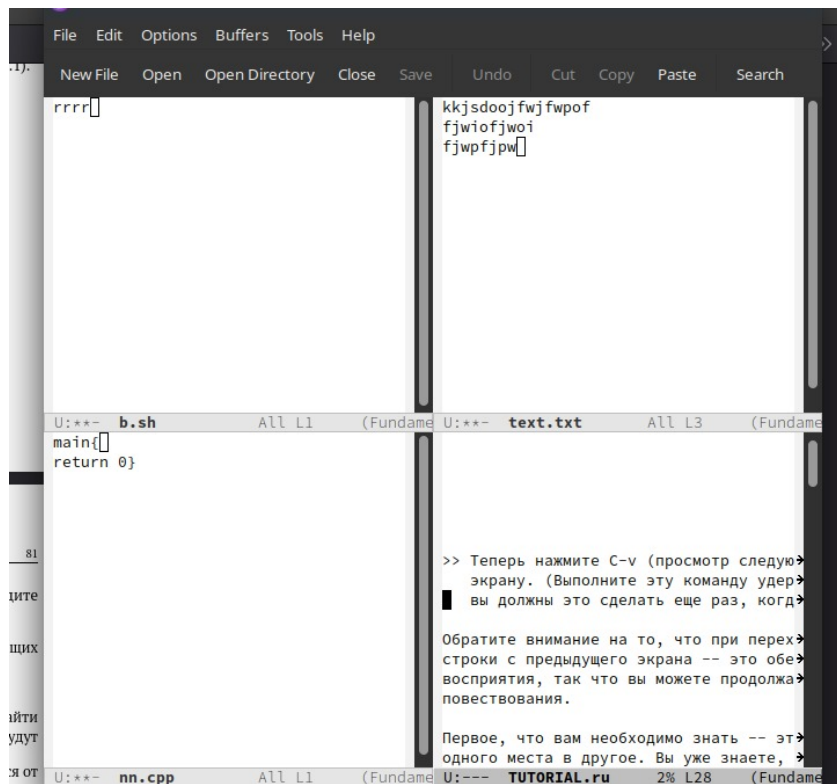


Рис. 4.19: Открытие буферов в четырех окнах

Перехожу в режим поиска с помощью C-s, ищу слова в тексте, они подсвечиваются (рис. fig. 4.20).

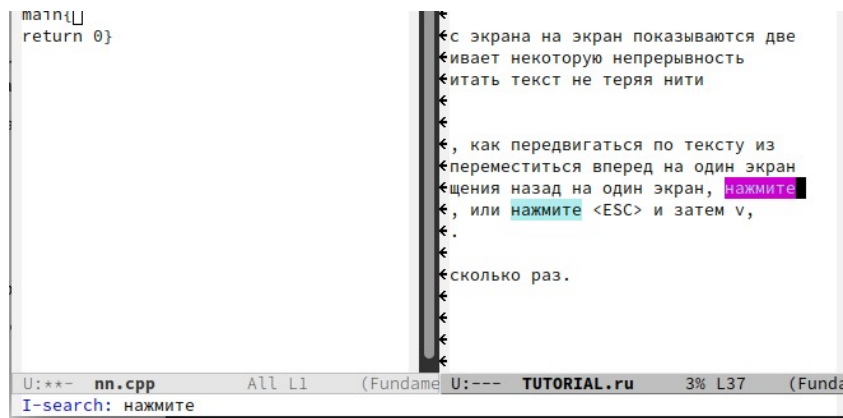


Рис. 4.20: Режим поиска

С помощью той же комбинации C-s я могу перемещаться по результатам поиска (рис. fig. 4.21).



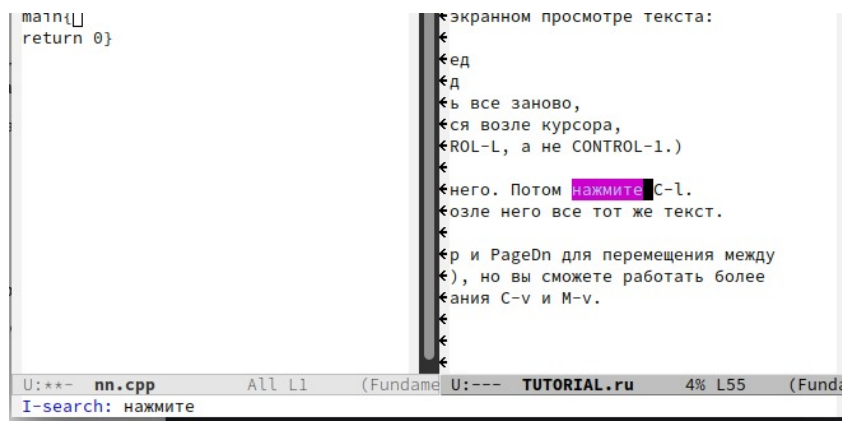


Рис. 4.21: Перемещение по найденным выражениям

С помощью C-g выхожу из режима поиска, снимается выделение (рис. fig. 4.22).

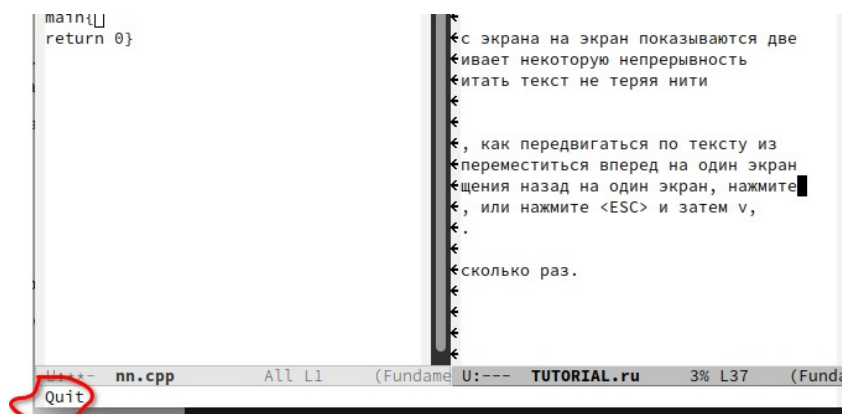


Рис. 4.22: Выход из режима поиска

Перехожу в режим поиска и замены с помощью M-%, ввожу какое слово хочу заменить, затем ввожу на какое хочу заменить (рис. fig. 4.23).

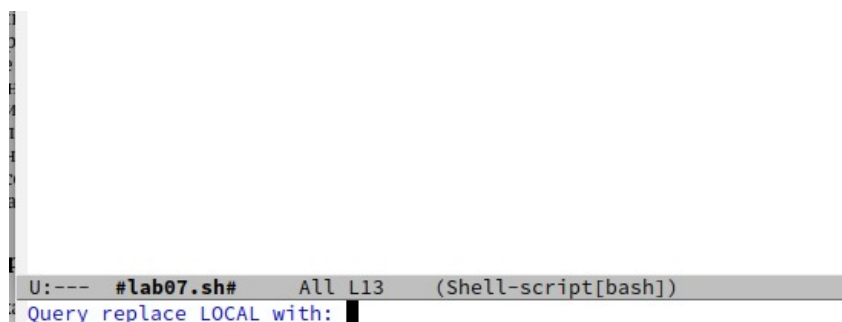


Рис. 4.23: Замена слова

Видим, что слова были заменены успешно (рис. fig. 4.24).

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

Рис. 4.24: Слово заменено

С помощью M+s перехожу в другой режим поиска. Он отличается от предыдущего тем, что выводит результат в отдельном окне от окна буфера (рис. fig. 4.25).

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

---

U:\*\*\* #lab07.sh# All L13 (Shell-script[bash])  
0 matches for "local" in buffer: #lab07.sh#  
7: local HELLO= World  
8: local HELLO= World

---

U:\*\*\* \*Occur\* All L1 (Occur)  
Searched 1 buffer; 2 matches for "local"

Рис. 4.25: Режим поиска

## 5 Выводы

В ходе данной лабораторной работы я познакомилась с операционной системой Linux, получила практические навыки работы с редактором Emacs.

## 6 Ответы на контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. Написан на языке высокого уровня Lisp.

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

Большое разнообразие сложных комбинаций клавиш, которые необходимы для редактирования файла и в принципе для работа с Emacs.

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

Буфер - это объект в виде текста. Окно - это прямоугольная область, в которой отображен буфер.

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

Да, можно.

5. Какие буферы создаются по умолчанию при запуске emacs?

Emacs использует буферы с именами, начинающимися с пробела, для внутренних целей. Отчасти он обращается с буферами с такими именами особым образом — например, по умолчанию в них не записывается информация для отмены изменений.

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

Ctrl + c, а потом | и Ctrl + c Ctrl + |

7. Как поделить текущее окно на две части?

C помощью команды Ctrl + x 3 (по вертикали) и Ctrl + x 2 (по горизонтали).

8. В каком файле хранятся настройки редактора emacs?

Настройки emacs хранятся в файле .emacs, который хранится в домашней директории пользователя. Кроме этого файла есть ещё папка .emacs.

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

Выполняет функцию стереть, думаю можно переназначить.

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

Для меня удобнее был редактор Emacs, так как у него есть командная оболочка. А vi открывается в терминале, и выглядит своеобразно.