

# **Отчет по лабораторной работе №9**

**Дисциплина: Операционные системы**

Шошина Евгения Александровна (НКАбд-03-22)

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
2.1	Последовательность выполнения работы . . . . .	6
2.1.1	Основные команды etacs . . . . .	6
<b>3</b>	<b>Теоретическое введение</b>	<b>9</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>10</b>
<b>5</b>	<b>Выводы</b>	<b>22</b>
<b>6</b>	<b>Контрольные вопросы</b>	<b>23</b>

# Список иллюстраций

figno1	Редактор Emacs . . . . .	10
figno2	Текст в редакторе . . . . .	11
figno3	Вырезание строчки . . . . .	12
figno4	Вставка строчки в конец файла . . . . .	12
figno5	Выделение области текста . . . . .	13
figno6	Вставка скопированной области в конец файла . . . . .	14
figno7	Вырезание области текста . . . . .	14
figno8	Отмена последнего действия . . . . .	15
figno9	Команды по перемещению курсора . . . . .	16
figno10	Список активных буферов . . . . .	17
figno11	Другой буфер . . . . .	17
figno12	Окна . . . . .	18
figno13	Окна . . . . .	19
figno14	Режим поиска . . . . .	20
figno15	Режим поиска . . . . .	20

## **Список таблиц**

# 1 Цель работы

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

## 2 Задание

### 2.1 Последовательность выполнения работы

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

#### 2.1.1 Основные команды emacs

1. Открыть emacs.
2. Создать файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).
3. Наберите текст, который дан.
4. Сохранить файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
5. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.
  - 5.1. Вырезать одной командой целую строку (C-k).
  - 5.2. Вставить эту строку в конец файла (C-y). 5.3. Выделить область текста (C-space).
  - 5.4. Скопировать область в буфер обмена (M-w).
  - 5.5. Вставить область в конец файла.
  - 5.6. Вновь выделить эту область и на этот раз вырезать её (C-w).
  - 5.7. Отмените последнее действие (C-/).

## 6. Научитесь использовать команды по перемещению курсора.

- 6.1. Переместите курсор в начало строки (C-a).
- 6.2. Переместите курсор в конец строки (C-e).
- 6.3. Переместите курсор в начало буфера (M-<).
- 6.4. Переместите курсор в конец буфера (M->).

## 7. Управление буферами.

- 7.1. Вывести список активных буферов на экран (C-x C-b).
- 7.2. Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер.
- 7.3. Закройте это окно (C-x 0).
- 7.4. Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран (C-x b).

## 8. Управление окнами.

- 8.1. Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2).
- 8.2. В каждом из четырёх созданных окон откройте новый буфер (файл) и введите несколько строк текста.

## 9. Режим поиска

- 9.1. Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте.
- 9.2. Переключайтесь между результатами поиска, нажимая C-s.
- 9.3. Выйдите из режима поиска, нажав C-g.
- 9.4. Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter, затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены.

- 9.5. Испробуйте другой режим поиска, нажав M-s o. Объясните, чем он отличается от обычного режима?

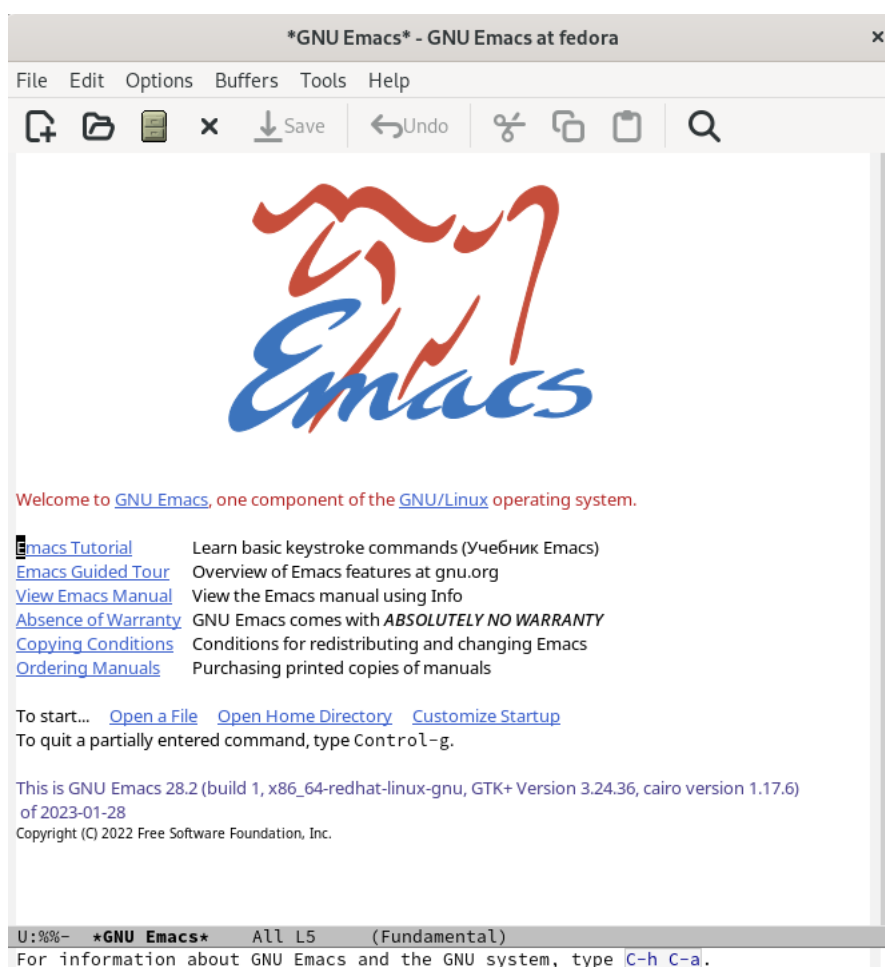


### 3 Теоретическое введение

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. [1]

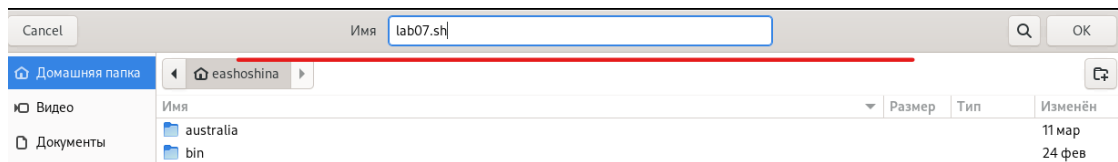
## 4 Выполнение лабораторной работы

1. Открыть emacs.



Редактор Emacs

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



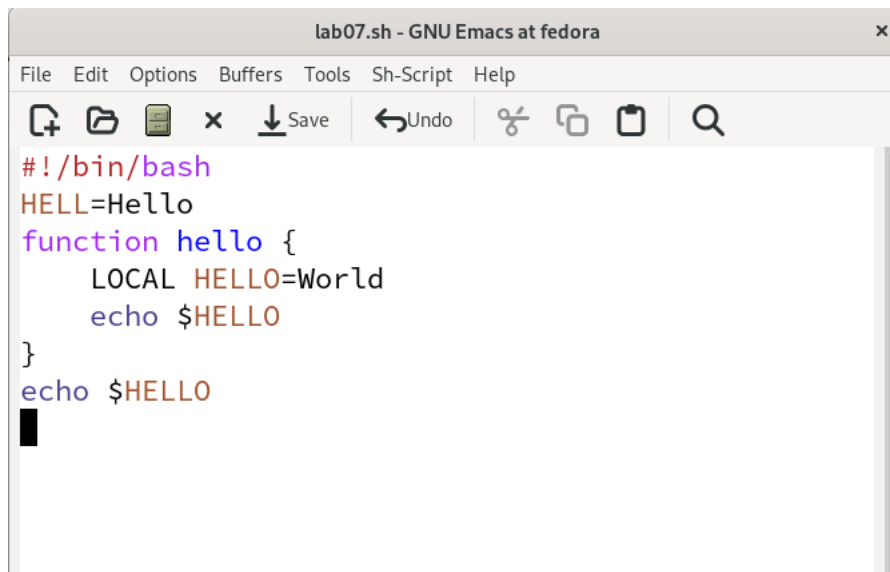
3. Наберите текст, который дан.
4. Сохранить файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).

A screenshot of the GNU Emacs editor window titled 'lab07.sh - GNU Emacs at fedora'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar has icons for file operations and editing. The text in the editor is as follows:

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

Текст в редакторе

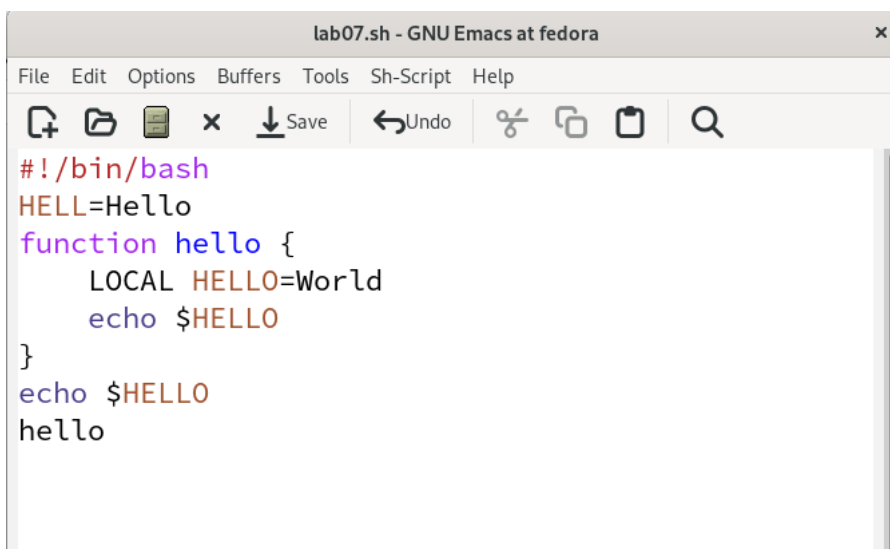
5. Прodelать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш. 5.1. Вырезать одной командой целую строку (C-k).



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

Вырезание строки

- 5.2. Вставить эту строку в конец файла (C-y).



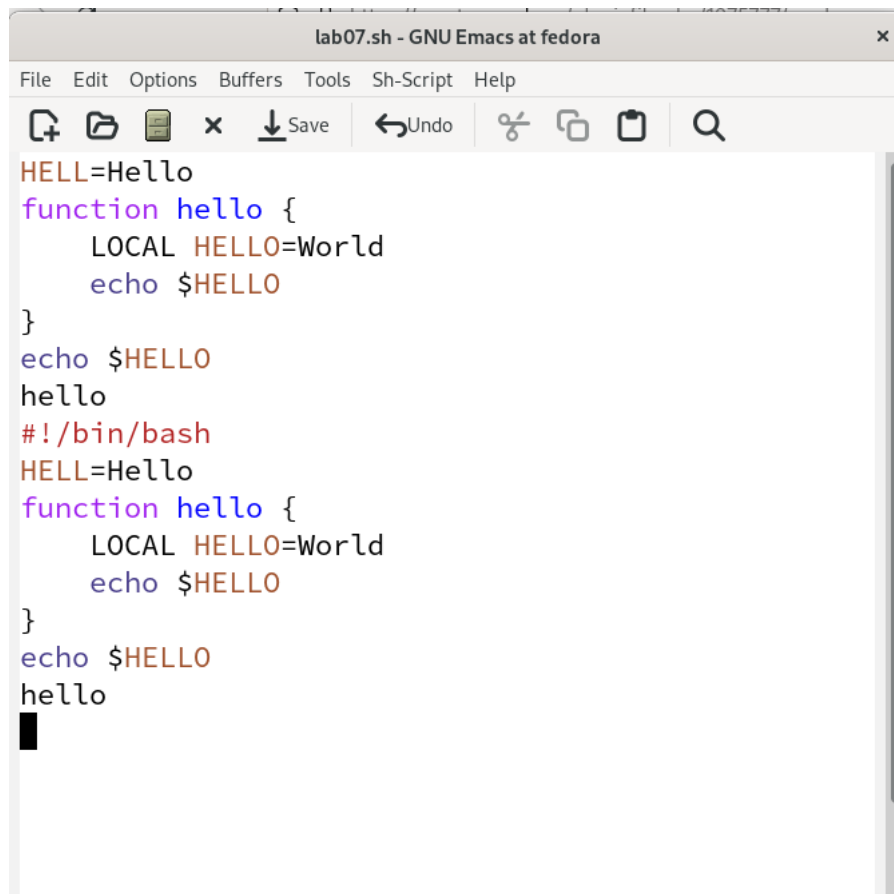
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Вставка строки в конец файла

- 5.3. Выделить область текста (C-space).

Выделение области текста

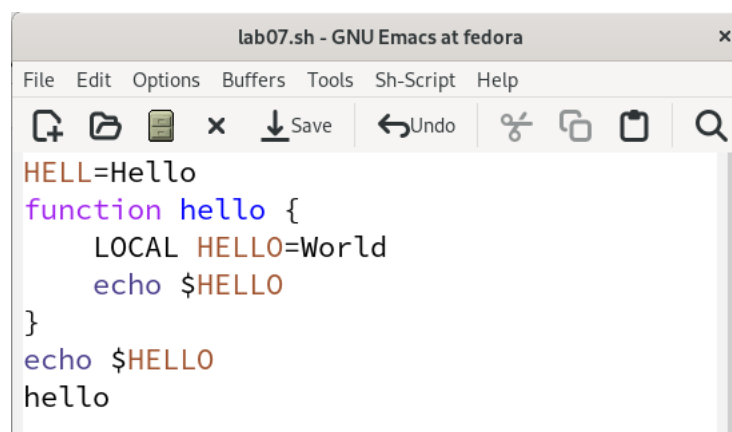
- 5.4. Скопировать область в буфер обмена (M-w).
- 5.5. Вставить область в конец файла.



```
lab07.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Cut, Copy, Paste, Find]
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
█
```

Вставка скопированной области в конец файла

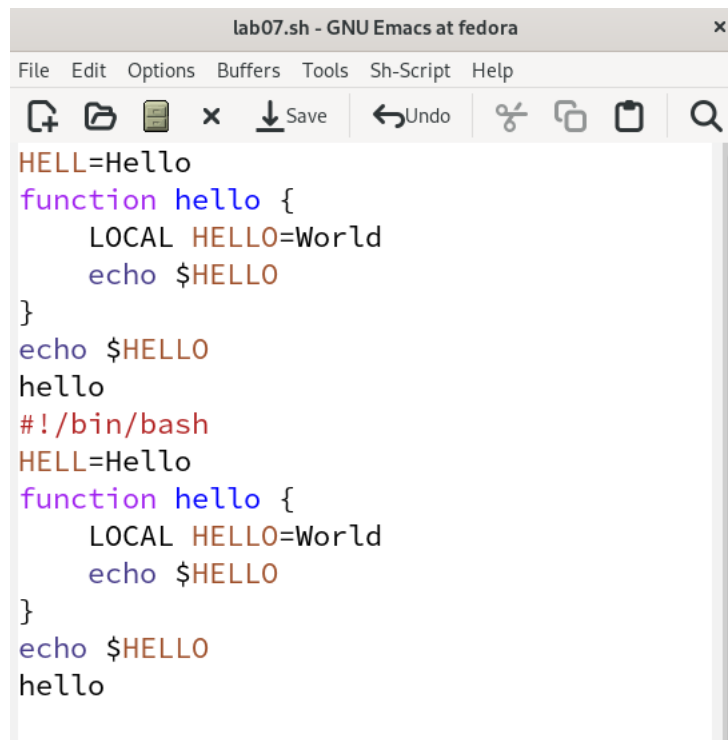
- 5.6. Вновь выделить эту область и на этот раз вырезать её (C-w).



```
lab07.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Cut, Copy, Paste, Find]
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Вырезание области текста

- 5.7. Отмените последнее действие (C-/).

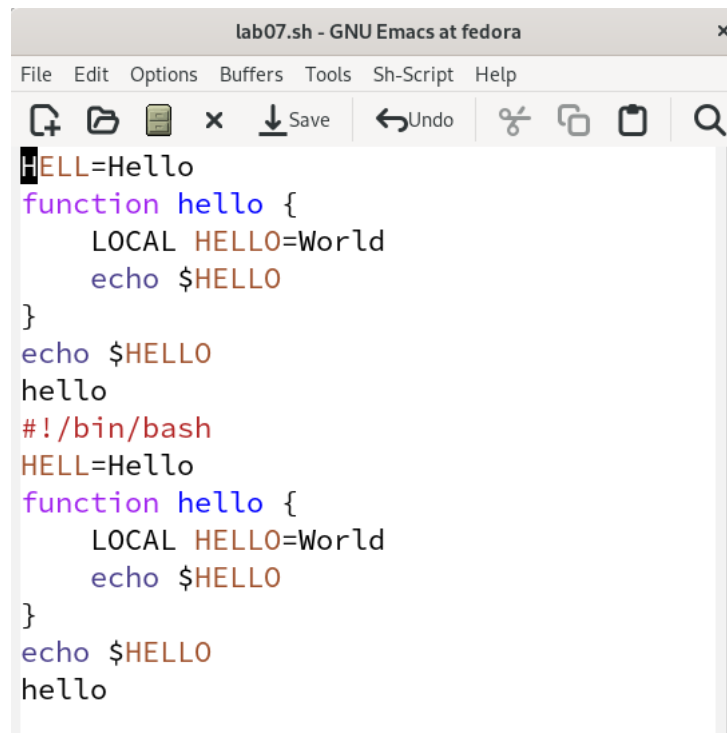


```
lab07.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, etc.]

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
```

Отмена последнего действия

6. Научитесь использовать команды по перемещению курсора. 6.1. Переместите курсор в начало строки (C-a).
- 6.2. Переместите курсор в конец строки (C-e).
  - 6.3. Переместите курсор в начало буфера (M-<). 6.4. Переместите курсор в конец буфера (M->).



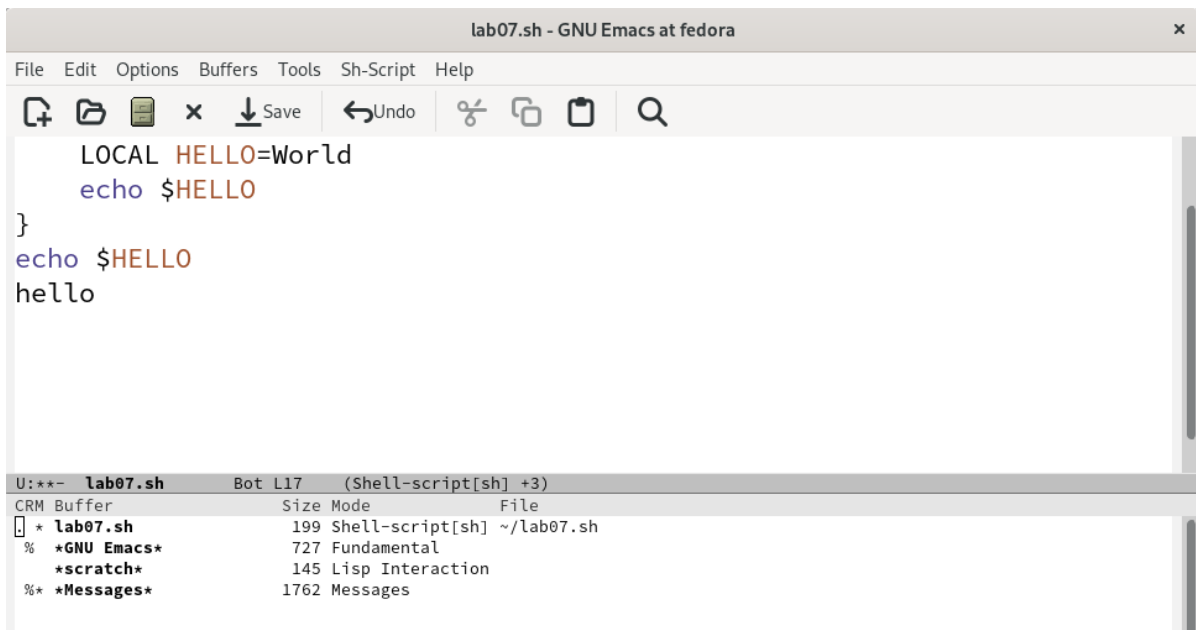
```
lab07.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, Copy, Paste, Find]
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Команды по перемещению курсора

## 7. Управление буферами.

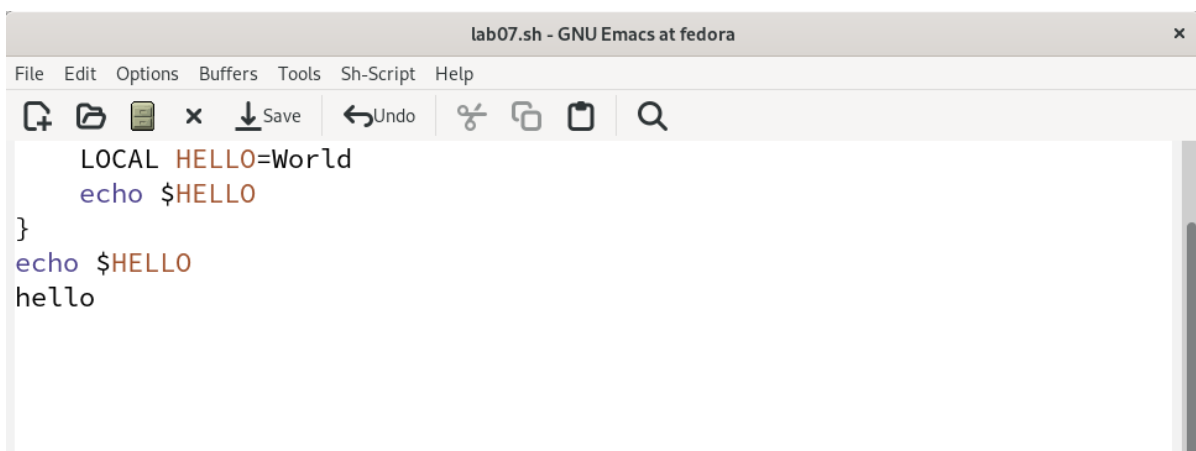
- 7.1. Вывести список активных буферов на экран (C-x C-b).





### Список активных буферов

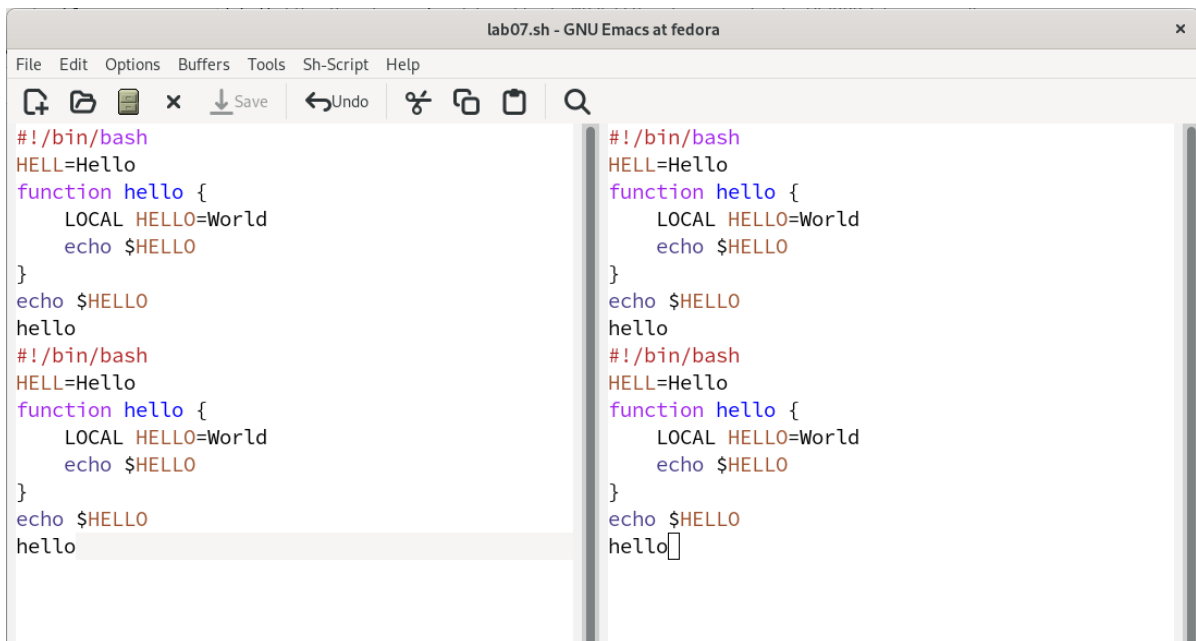
- 7.2. Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер.



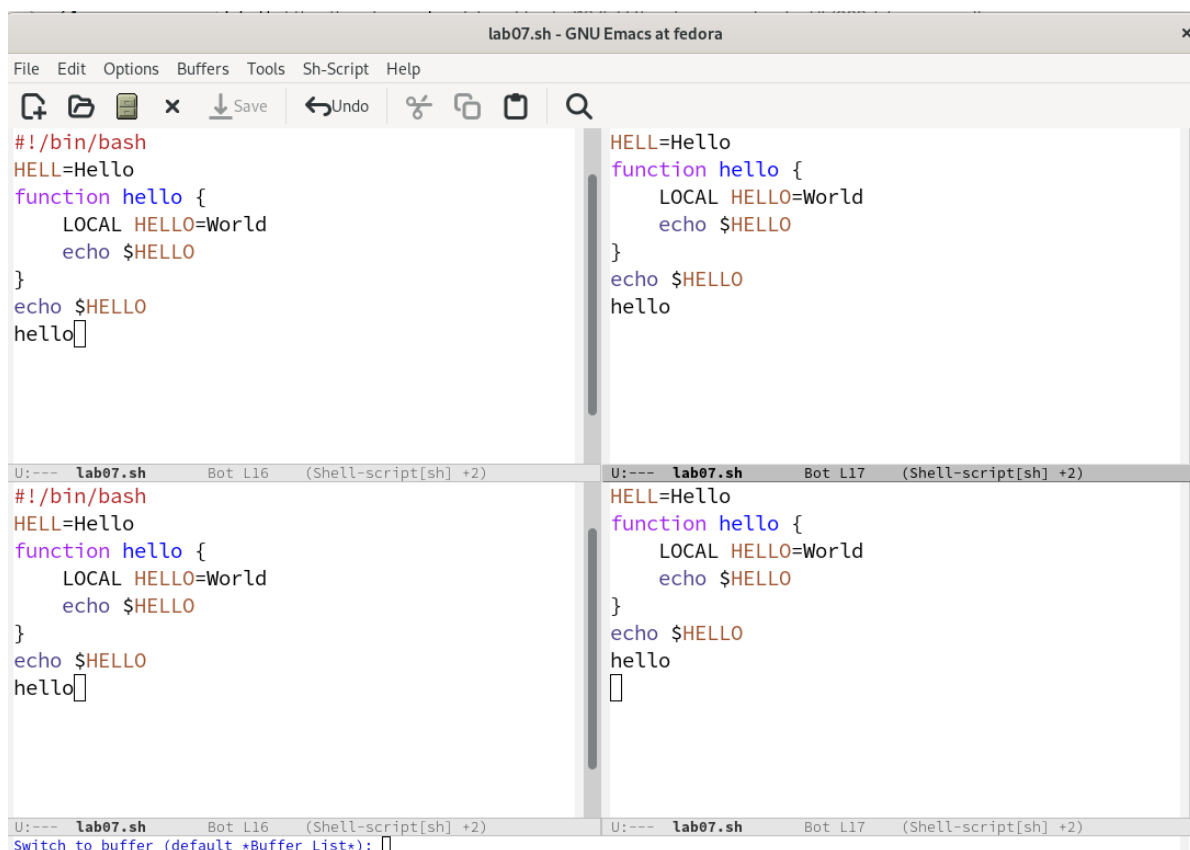
### Другой буфер

- 7.3. Закройте это окно (C-x 0).
  - 7.4. Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран (C-x b).
8. Управление окнами.

- 8.1. Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2).



2 окна

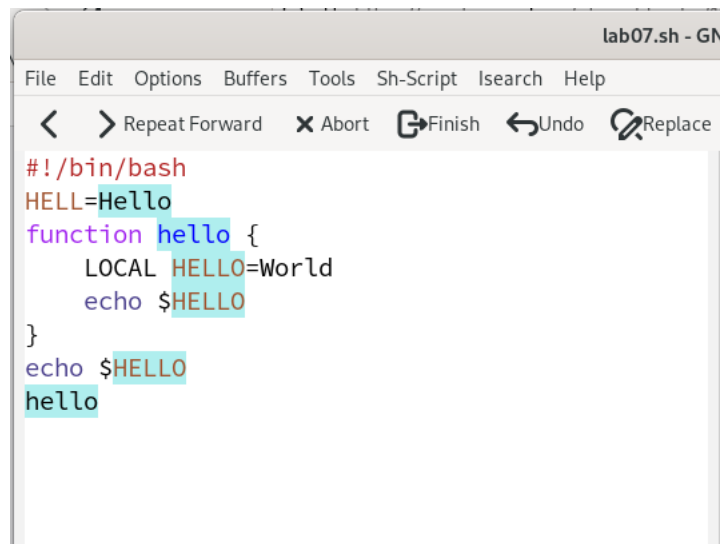


4 окна

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

## 9. Режим поиска

- 9.1. Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте.

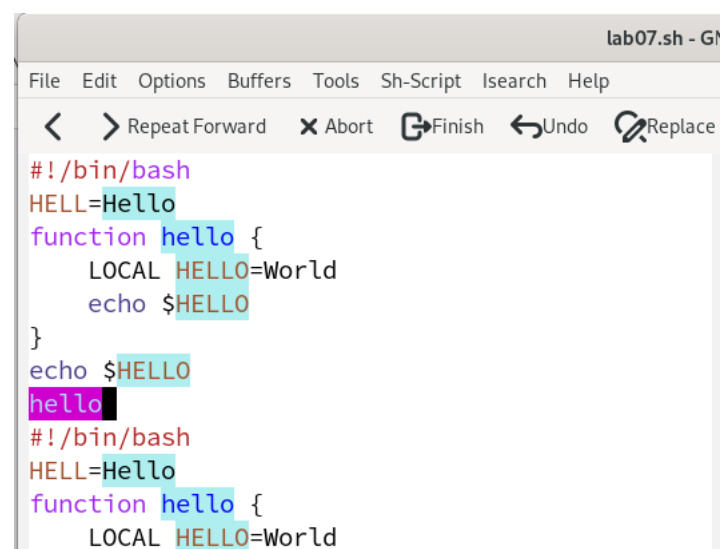
A screenshot of a terminal window titled 'lab07.sh - GN'. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', 'Isearch', and 'Help'. Below the menu bar is a toolbar with icons for 'Repeat Forward', 'Abort', 'Finish', 'Undo', and 'Replace'. The script content is as follows:

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

The words 'Hello' and 'HELLO' are highlighted in cyan, and 'hello' is highlighted in blue.

Режим поиска

- 9.2. Переключайтесь между результатами поиска, нажимая C-s.

A screenshot of a terminal window titled 'lab07.sh - GN' showing the same script as the previous image. The word 'hello' on the line 'hello' is highlighted in magenta. Below it, the first few lines of the script are repeated, showing the search results for the pattern 'hello'. The search results are as follows:

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

Режим поиска

- 9.3. Выйдите из режима поиска, нажав C-g.
- 9.4. Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter, затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены.

- 9.5. Испробуйте другой режим поиска, нажав M-s o. Объясните, чем он отличается от обычного режима?

**4.0.0.0.0.1** Отличие от обычного режима в том, что тут появляется отдельное окно с текстом из файла с выделенными словами, которые нужно было найти.

## **5 Выводы**

В процессе выполнения лабораторной работы я получила практические навыки работы в редакторе Emacs.

## 6 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?
- С 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.