

# **Отчёт по лабораторной работе №9**

**Текстовый редактор emacs**

Гоша

# Содержание

0.1	Цель работы . . . . .	3
<b>1</b>	<b>Выполнение лабораторной работы</b>	<b>4</b>
1.1	Ознакомление с etacs . . . . .	4
1.2	Перемещение по документу . . . . .	5
1.3	Управление буферами . . . . .	5
1.4	Управление окнами . . . . .	7
1.5	Режим поиска . . . . .	8
1.6	Ответы на контрольные вопросы . . . . .	10
1.7	Заключение . . . . .	11

# Список иллюстраций

1.1	Создание файла . . . . .	4
1.2	Изменение файла . . . . .	5
1.3	Активные буферы . . . . .	6
1.4	Переключение буфера . . . . .	6
1.5	Разделение окон . . . . .	7
1.6	Редактирование буферов . . . . .	8
1.7	Поиск слова . . . . .	9
1.8	Выход из режима поиска . . . . .	9
1.9	Замена текста . . . . .	9
1.10	Другой режим поиска . . . . .	10

## 0.1 Цель работы

Цель работы — познакомиться с интерфейсом текстового редактора emacs.

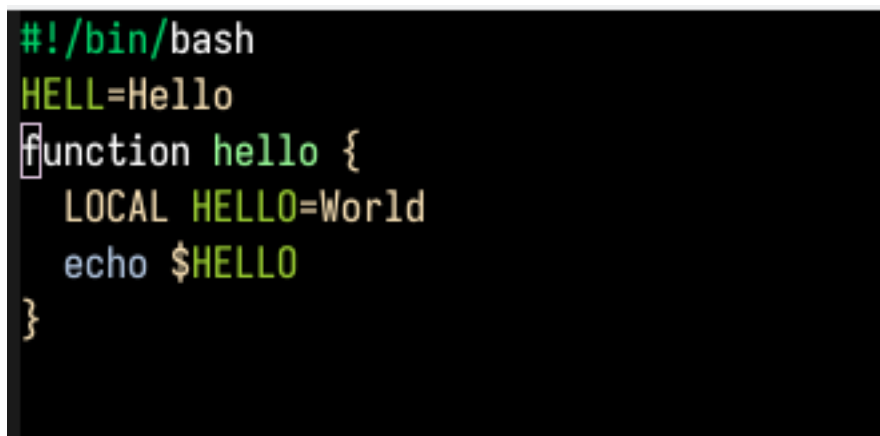
# 1 Выполнение лабораторной работы

## 1.1 Ознакомление с emacs

В редакторе emacs создали новый файл lab09.sh и ввели следующий код:

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

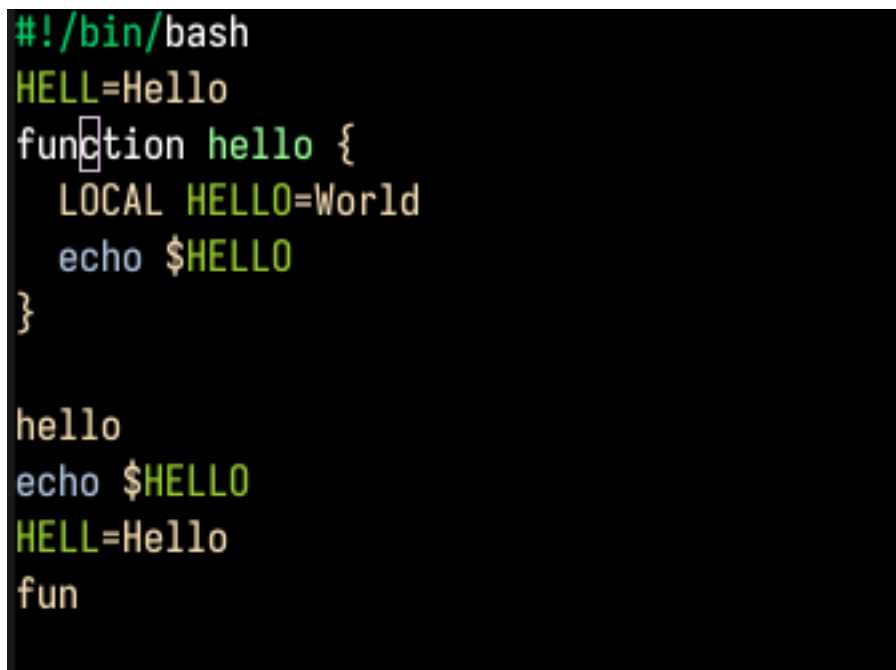
Сохранили файл:

A screenshot of a terminal window with a black background and green and yellow text. The text shows the script content: a shebang line, a variable assignment, a function definition with a local variable and an echo command, and two final echo and function call lines.

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

Рис. 1.1: Создание файла

Затем по порядку выполнили следующие махинации над текстом: 1. Вырезали строку при помощи комбинации клавиш C-k 2. Вставили эту строку в конце файла (C-y) 3. Выделили область текста (C-space) 4. Скопировали область в буфер обмена (M-w) 5. Вставили область в конец файла 6. Вырезали ту же область (C-w) 7. Отменили последнее действие (C-\)



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

hello
echo $HELLO
HELL=Hello
fun
```

Рис. 1.2: Изменение файла

## 1.2 Перемещение по документу

Далее научились перемещать курсор в начало строки (C-a), в конец строки (C-e), в начало буфера (M-<) и конец буфера (M->).

## 1.3 Управление буферами

Вывели список активных буферов на экран (C-x C-b):

CRM	Buffer	Size	Mode	File
.	* lab07.sh	109	Shell-script[sh]	~/lab07.sh
%*	*Custom Themes*	1612	Themes	
%	*GNU Emacs*	723	Fundamental	
	*scratch*	145	Lisp Interaction	
%*	*Messages*	50875	Messages	
%*	*Compile-Log*	248	elisp-compile	
%*	*Async-native-compil...	12807	Fundamental	
%*	*Warnings*	1035	Special	
%	*Disabled Command*	771	Help	
%	*Quail Completions*	0	Fundamental	

Рис. 1.3: Активные буферы

Переместились в открытое окно. Следом закрыли это окно (C-x 0). Вновь переключились между буферами, но уже без вывода их списка на экран (C-x b).

```

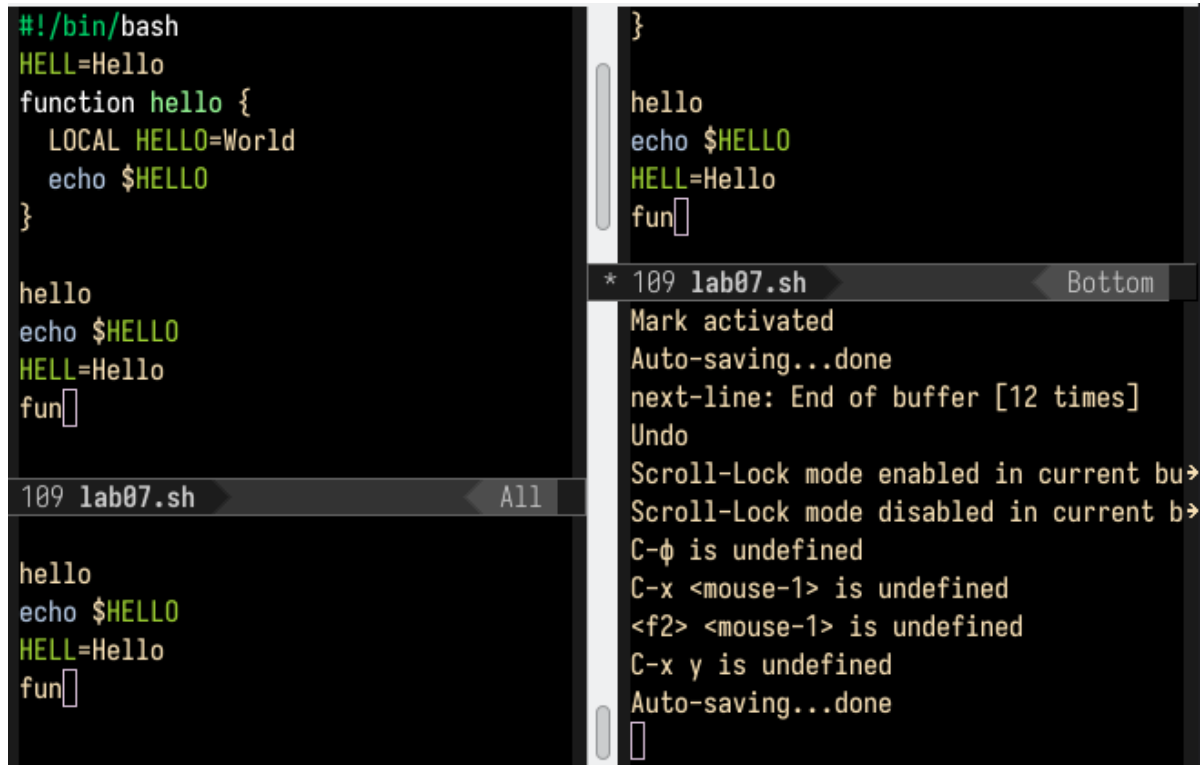
Mark activated
Auto-saving...done
next-line: End of buffer [12 times]
Undo
Scroll-Lock mode enabled in current buffer
Scroll-Lock mode disabled in current buffer
C-φ is undefined
C-x <mouse-1> is undefined

```

Рис. 1.4: Переключение буфера

## 1.4 Управление окнами

При помощи сочетаний клавиш C-x 3 и C-x 2 поделили фрейм сначала по вертикали, а затем дважды по горизонтали:



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

hello
echo $HELLO
HELL=Hello
fun

hello
echo $HELLO
HELL=Hello
fun

* 109 lab07.sh Bottom
Mark activated
Auto-saving...done
next-line: End of buffer [12 times]
Undo
Scroll-Lock mode enabled in current bu
Scroll-Lock mode disabled in current b
C-φ is undefined
C-x <mouse-1> is undefined
<f2> <mouse-1> is undefined
C-x y is undefined
Auto-saving...done
```

Рис. 1.5: Разделение окон

В каждом окне открыли новый буфер и ввели там несколько строк:

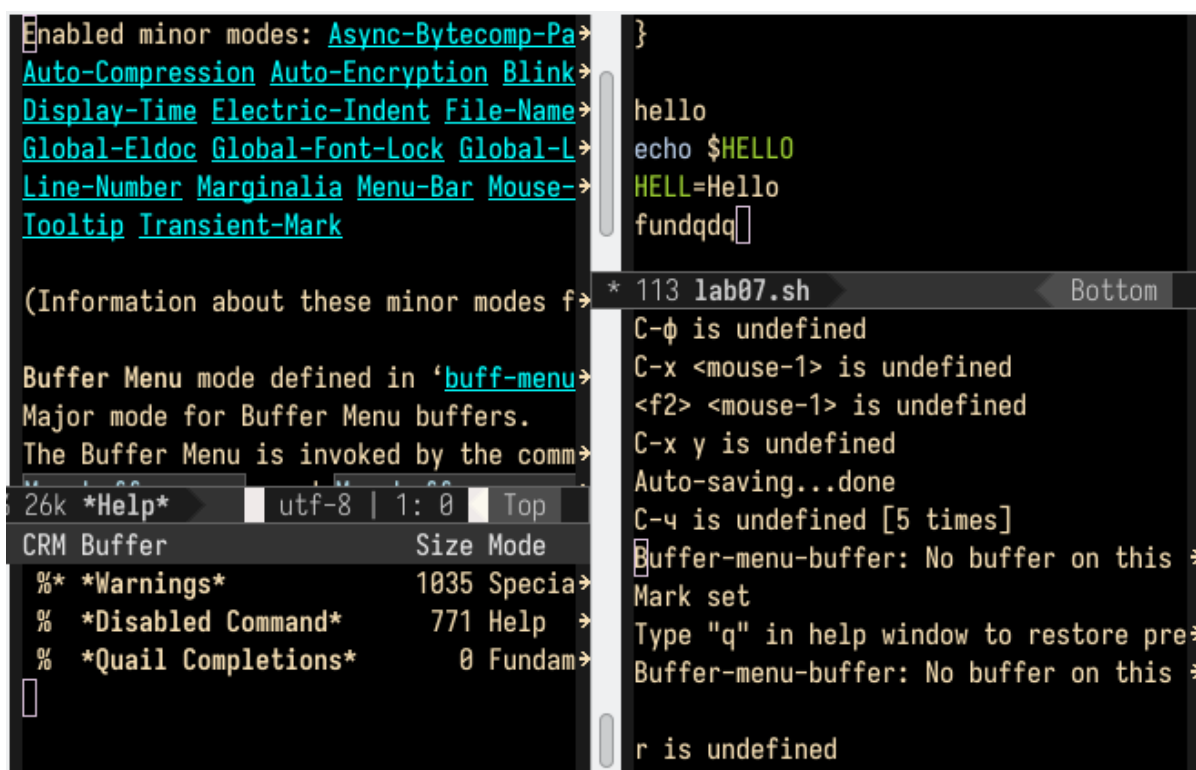


Рис. 1.6: Редактирование буферов

## 1.5 Режим поиска

В режиме поиска (C-s) нашли слово *World*:



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

hello
echo $HELLO
HELL=Hello
fun
```

109 lab07.sh | Shell-script | unix | 11: 3 | All

Failing case-sensitive I-search: World

Рис. 1.7: Поиск слова

Вышли из режима поиска (C-g):

Quit

Рис. 1.8: Выход из режима поиска

Выполнили замену текста (M-%):

Replaced 0 occurrences

Рис. 1.9: Замена текста

Выполнили поиск уже при помощи сочетания клавиш M-s:



Рис. 1.10: Другой режим поиска

От первого режима отличается тем, что выполняет поиск по словам целиком и при нахождении совпадения переводит каретку в конец первого совпадения. Первый режим текста находит любое совпадение.

## 1.6 Ответы на контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

*Ответ:* emacs — экранный редактор текста.

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

*Ответ:* совершенно не поддающиеся какой бы то ни было логике сочетания клавиш поставят любого пользователя в тупик. Удачи.

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

*Ответ:* буфер — хранилище текста, окно — хранилище буферов.

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

*Ответ:* оказалось, что нет.

5. Какие буферы создаются по умолчанию при запуске emacs?

*Ответ:* 'GNU Emacs' и \*Warnings\*.

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

*Ответ:* `ctrl-c-shift-\` и `ctrl-c ctrl-shift-\`.

7. Как поделить текущее окно на две части?

*Ответ:* по горизонтали (C-x 2) и вертикали (C-x 3).

8. В каком файле хранятся настройки редактора emacs?

*Ответ:* `~/ .emacs`.

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

*Ответ:* перемещение каретки по буферу. Переназначить нельзя.

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

*Ответ:* vi оказался удобнее ввиду своей легковесности.

## 1.7 Заключение

В ходе выполнения лабораторной работы был изучен интерфейс текстового редактора emacs. Цель работы была достигнута.