

Презентация по лабораторной работе №11

Текстовый редактор emacs

Вакутайпа М.

17 Апреля 2024

Российский университет дружбы народов, Москва, Россия

Информация

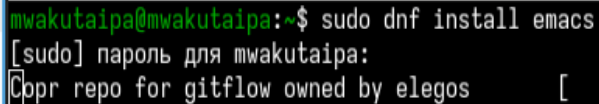
- Вакутайпа Милдред
- студентка НКА 02-23
- факультет физико-математических и естественных наук
- Российский университет дружбы народов
- 1032239009@rudn.ru
- <https://wakutaipa.github.io/ru/>

.. ..
.. ..

Цель работы

Получить практические навыки работы с редактором Emacs.

Для данной работы, мне надо была установить Emacs:

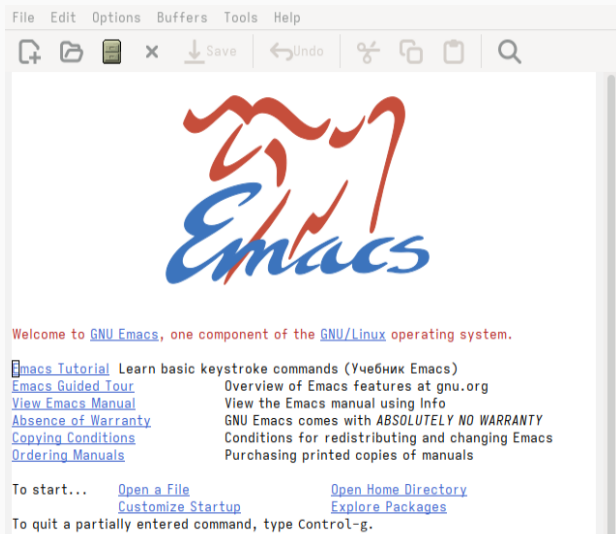
A terminal window with a black background and green text. The prompt is 'mwakutaipa@mwakutaipa:~\$'. The command entered is 'sudo dnf install emacs'. The output shows the password prompt '[sudo] пароль для mwakutaipa:', followed by a confirmation message 'Copr repo for gitflow owned by elegos' and a bracket '['.

```
mwakutaipa@mwakutaipa:~$ sudo dnf install emacs
[sudo] пароль для mwakutaipa:
Copr repo for gitflow owned by elegos      [
```

Рис. 1: Установка Emacs

Основные команды с редактором

Выполнив Emacs в командной строке, я открыла текстовый редактор:



Основные команды с редактором

С помощью комбинации Ctrl-x Ctrl-f, создала файл lab07.sh:

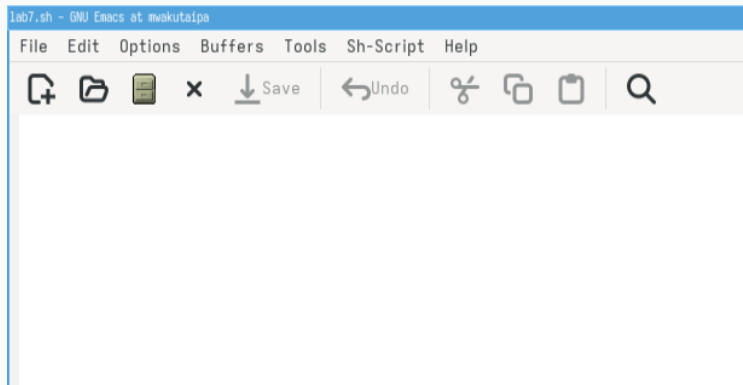
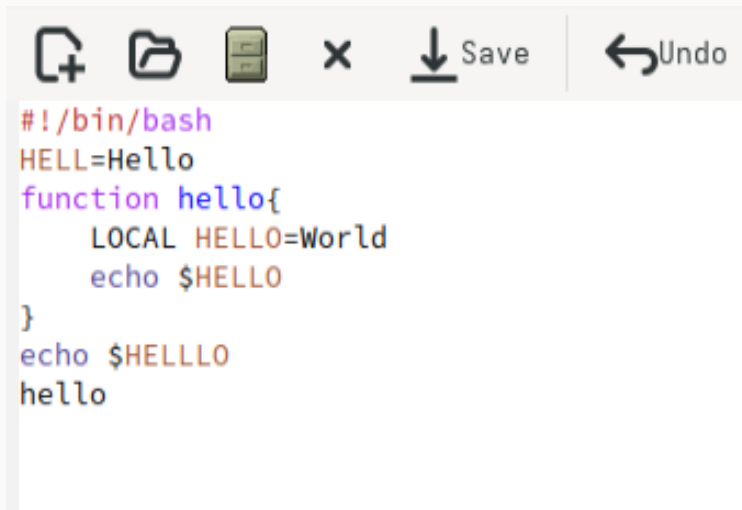


Рис. 3: Созданный файл

Основные команды с редактором

Я сохранила файл с помощью комбинации Ctrl-x Ctrl-s:



The image shows a code editor window. The toolbar at the top contains icons for opening a file, saving a file, closing a file, and a 'Save' button with a downward arrow. To the right of the toolbar is an 'Undo' button with a left-pointing arrow. The code area below the toolbar contains a shell script with the following content:

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

Вырезать целую строку (C-k):

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

hello
echo $HELLO
```

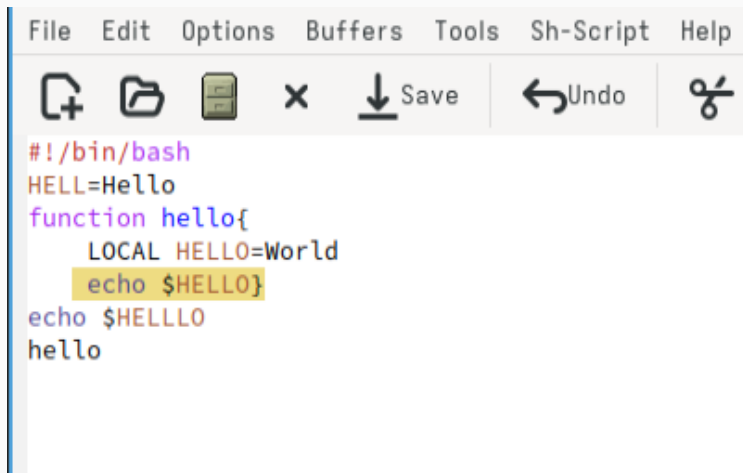
С помощью C-у вставила эту строку в конец файла:

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

hello
    echo $HELLO
echo $HELLLO
```

Основные команды с редактором

Выделила область текста (C-space):



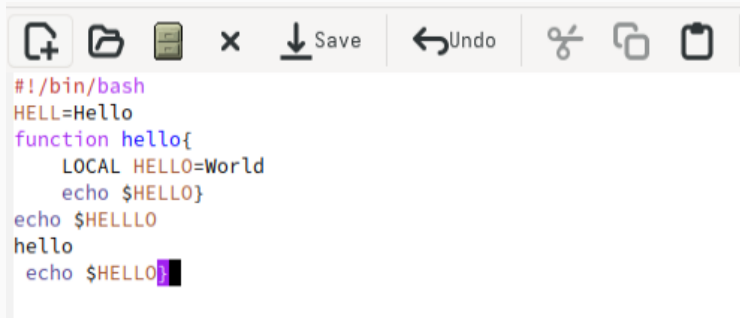
The image shows a screenshot of a text editor window. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with icons for opening a file, saving a file, closing a file, a close button (X), a save button (floppy disk), a 'Save' label, an 'Undo' button (curved arrow), and a cut button (scissors). The main text area contains a shell script with the following content:

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

The line `echo $HELLO}` is highlighted in yellow.

Рис. 7: Выделенный текст

Скопировала область в буфер обмена (M-w) и вставила ее в конец файла:

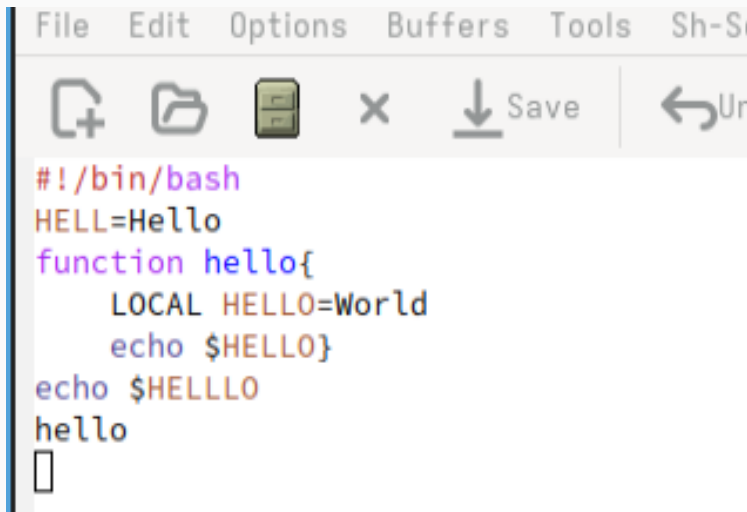


```
#!/bin/bash
HELL=Hello
function hello{
    LOCAL HELLO=World
    echo $HELLO}
echo $HELLLO
hello
echo $HELLO
```

Рис. 8: копирование и вставка

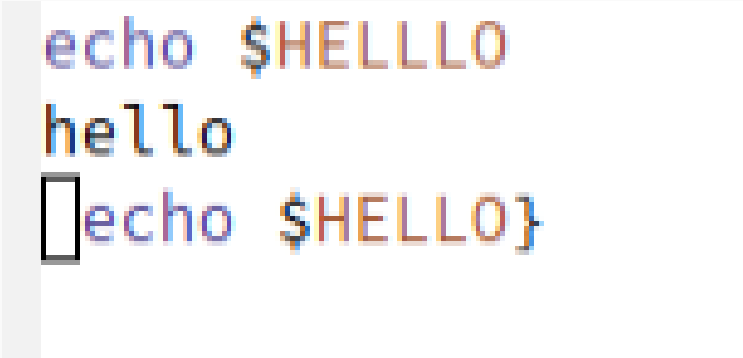
Основные команды с редактором

Выделила эту же область и на этот раз вырезала её (C-w):



```
File Edit Options Buffers Tools Sh-S
[Copy] [Paste] [Save] [Undo]
#!/bin/bash
HELL=Hello
function hello{
    LOCAL HELLO=World
    echo $HELLO}
echo $HELLLO
hello
█
```

С помощью C-/ отменила последнее действие:

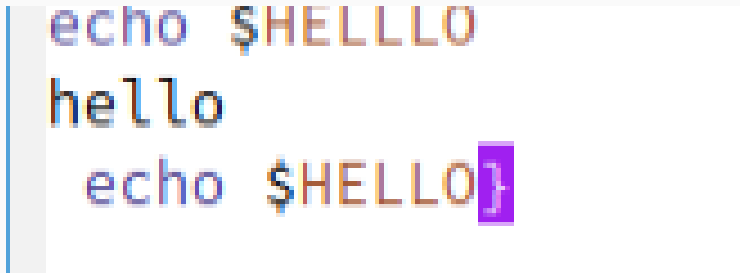


```
echo $HELLO  
hello  
[ echo $HELLO}
```

The image shows a terminal window with a light gray background. It displays three lines of text: 'echo \$HELLO', 'hello', and '[echo \$HELLO}'. The third line is partially enclosed by a black rectangular cursor box on its left side, indicating it is the current line being edited. The text is rendered in a monospaced font with a color scheme where 'echo' is blue, '\$HELLO' is orange, and the closing brace '}' is blue.

Рис. 10: отмена действие

С помощью C-a переместила курсор в начало строки:



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

Рис. 11: Перемещение курсор в начало строки

С помощью C-e переместила курсор в конец строки:

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

Переместила курсор в начало и конец буфера с помощью M-< и M-> соответственно:

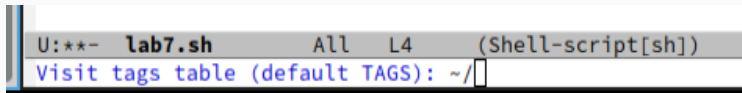
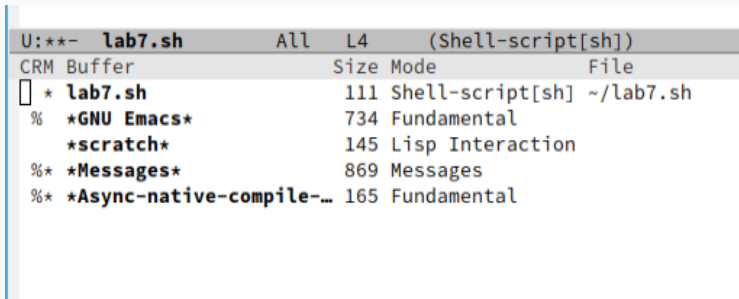


Рис. 13: Перемещение курсор в буфере

Выводила список активных буферов на экран с помощью C-x C-b:

A screenshot of the Emacs buffer list window. The title bar at the top reads 'U: **~ lab7.sh All L4 (Shell-script[sh])'. The table below lists active buffers with columns for 'Buffer', 'Size', 'Mode', and 'File'. The first buffer, 'lab7.sh', is selected and highlighted. Other buffers include '*GNU Emacs*', '*scratch*', '*Messages*', and '*Async-native-compile-...'.

U: **~ lab7.sh All L4 (Shell-script[sh])			
CRM Buffer	Size	Mode	File
<input checked="" type="checkbox"/> * lab7.sh	111	Shell-script[sh]	~/lab7.sh
% *GNU Emacs*	734	Fundamental	
scratch	145	Lisp Interaction	
%* *Messages*	869	Messages	
%* *Async-native-compile-...	165	Fundamental	

Рис. 14: Активные буферы

Основные кманды с редактором

С помощью C-x o переместилась во вновь открытое окно со списком открытых буферов и переключилась на другой буфер:

```
-:--- lab7.sh All L9 (Shell-script[bash])
CRM Buffer Size Mode File
[ ] lab7.sh 111 Shell-script[ba... ~/lab7.sh
% *GNU Emacs* 734 Fundamental
*scratch* 145 Lisp Interaction
%* *Messages* 552 Messages
%* *Async-native-compile-... 165 Fundamental
```

С помощью C-x 0 закрыла окно со списком открытых буферов:

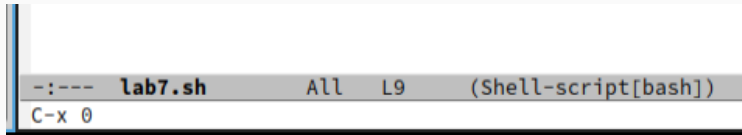
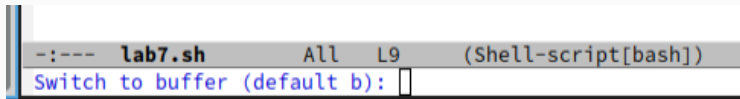


Рис. 16: Заккрытие окно

Без вывода списка буферов, я переключилась между буферами:



```
-:--- lab7.sh All L9 (Shell-script[bash])  
Switch to buffer (default b):
```

Рис. 17: Переключение между буферами

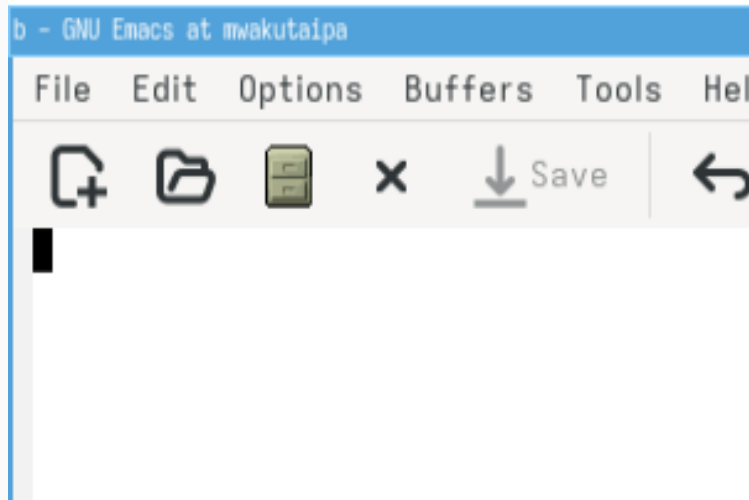
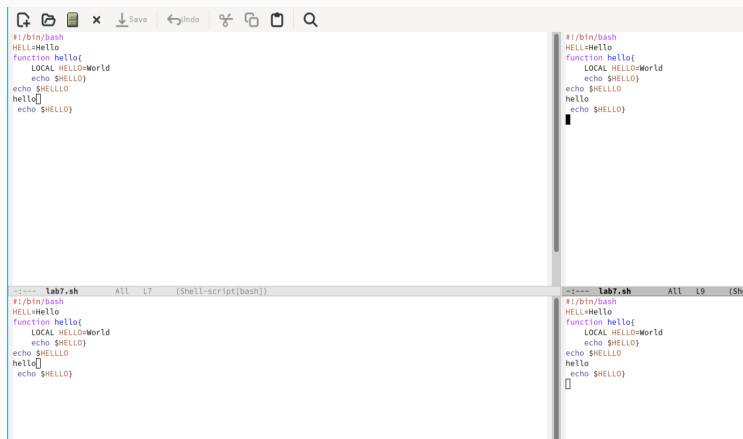


Рис. 18: Новый буфер

Основные кманды с редактором

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



В каждом из четырёх созданных окон открыла новый буфер:

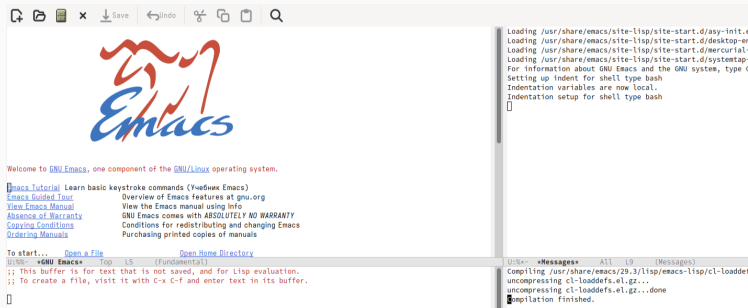


Рис. 20: Новые буферы

Переключилась в режим поиска (C-s) и искала Indent:

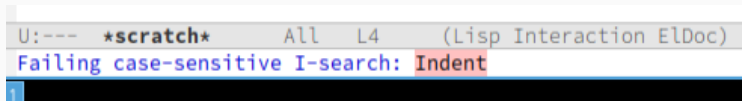


Рис. 21: Режим поиска

Переключалась между результатами поиска, нажимая C-s и вышла из режима поиска, нажав C-g:

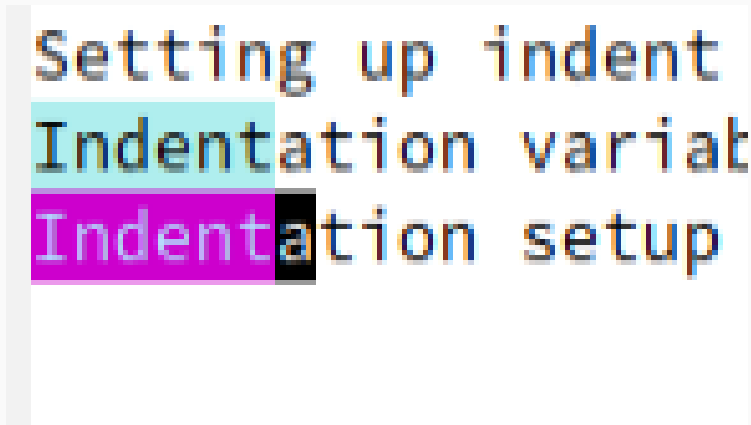


Рис. 22: Переключение между результатами

Перешла в режим поиска и замены (M-% или M-x query replace):

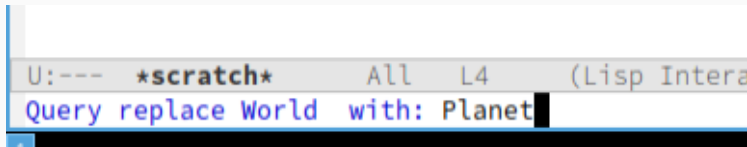


Рис. 23: Режим поиска

Нажав M-s o, я использовала другой режим поиска. Он отличается от предыдущего тем, что выводит результаты поиска в новом окне:

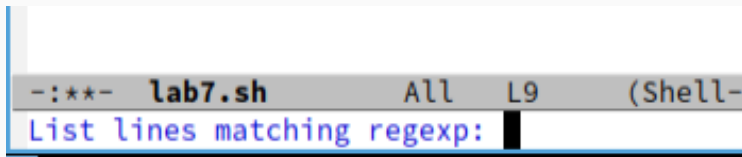
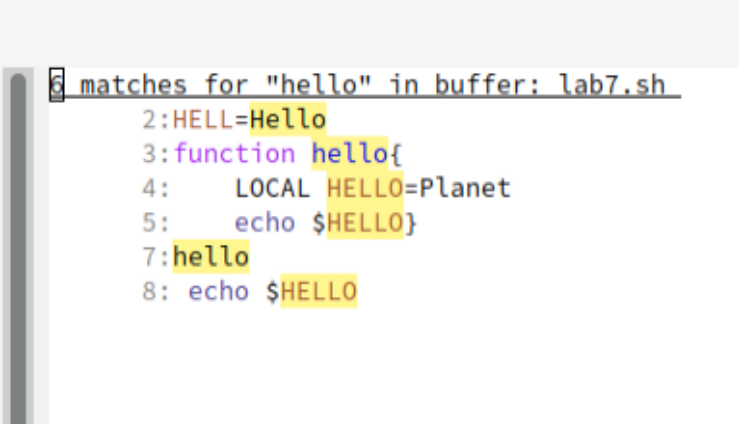


Рис. 24: другой режим поиска



```
6 matches for "hello" in buffer: lab7.sh
2:HELL=Hello
3:function hello{
4:    LOCAL HELLO=Planet
5:    echo $HELLO}
7:hello
8: echo $HELLO
```

Рис. 25: Результаты поиска

Выводы

При выполнении данной работы я получила практические навыки работы с Emacs.