

Презентация по лабораторной работе №11

Текстовый редактор emacs

Вакутайпа М.

17 Апреля 2024

Российский университет дружбы народов, Москва, Россия

Информация

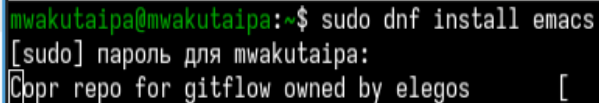
- Вакутайпа Милдред
- студентка НКА 02-23
- факультет физико-математических и естественных наук
- Российский университет дружбы народов
- 1032239009@rudn.ru
- <https://wakutaipa.github.io/ru/>

.. ..
.. ..

Цель работы

Получить практические навыки работы с редактором Emacs.

Для данной работы, мне надо была установить Emacs:

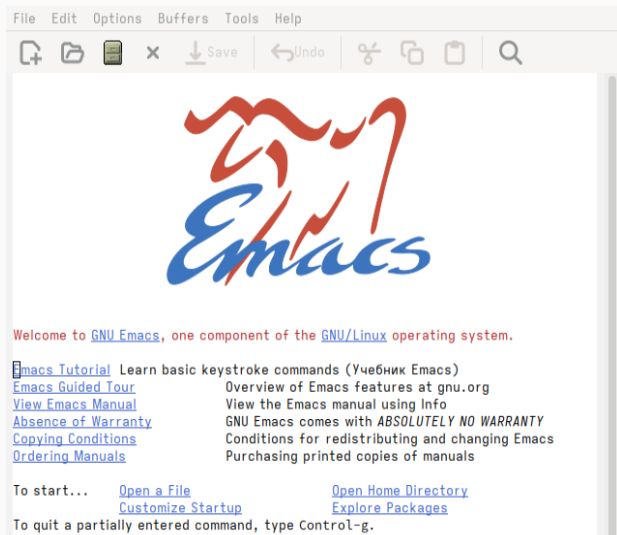
A terminal window with a black background and green text. The prompt is 'mwakutaipa@mwakutaipa:~\$'. The command 'sudo dnf install emacs' has been entered. The output shows '[sudo] пароль для mwakutaipa:' followed by a blank line, and then 'Copr repo for gitflow owned by elegos' followed by a '[' character.

```
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:

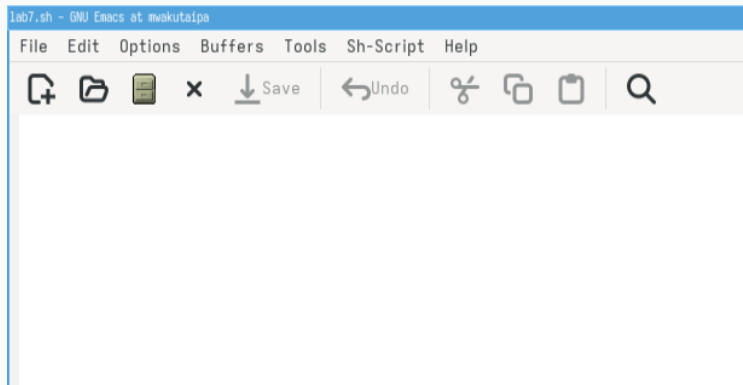
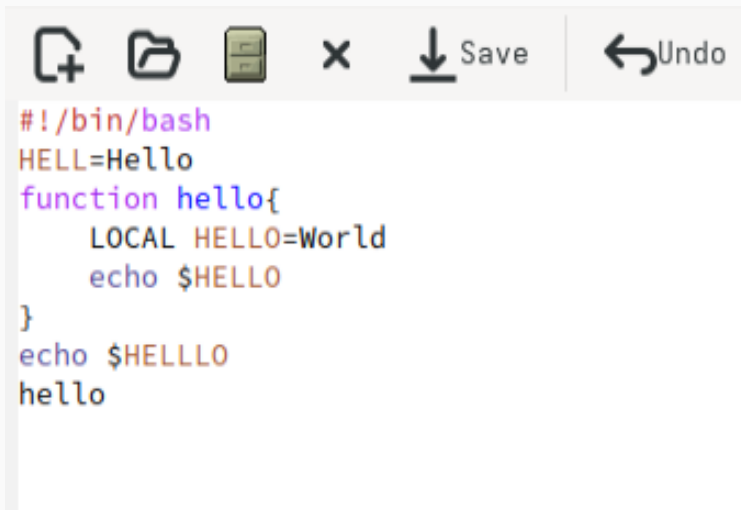


Рис. 3: Созданный файл

Основные команды с редактором

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



The image shows a code editor window. At the top is a toolbar with icons for opening a file, saving a file, closing a file, and a 'Save' button with a downward arrow. To the right of the 'Save' button is an 'Undo' button with a left-pointing arrow. Below the toolbar, the editor 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}

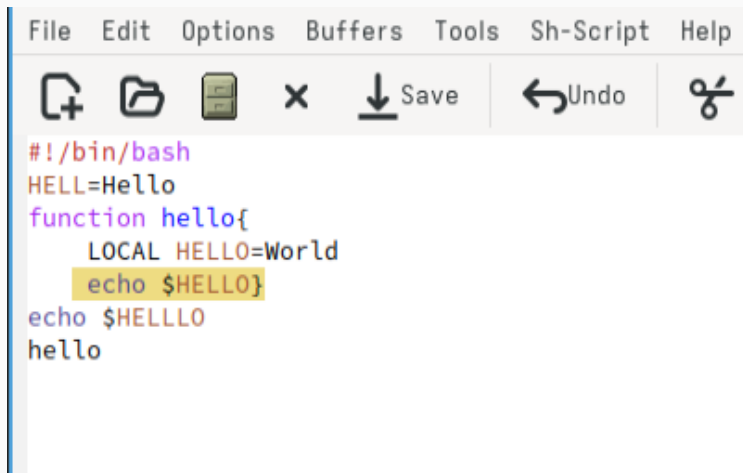
hello
echo $HELLO
```

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

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

hello
    echo $HELLO
echo $HELLLO
```

Выделила область текста (C-space):



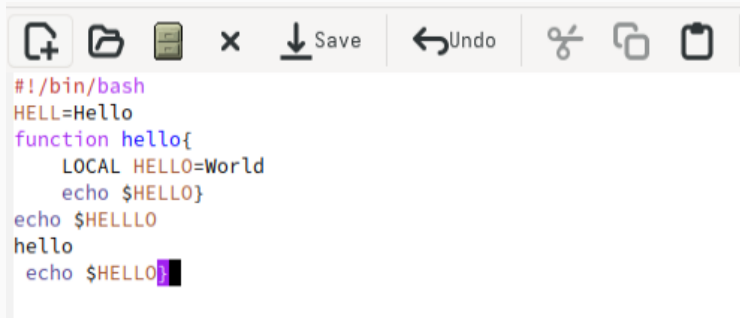
The image shows a screenshot of a text editor window. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with icons for opening a file, saving, closing, and undo, along with text labels 'Save', 'Undo', and a scissors icon for cut. 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:

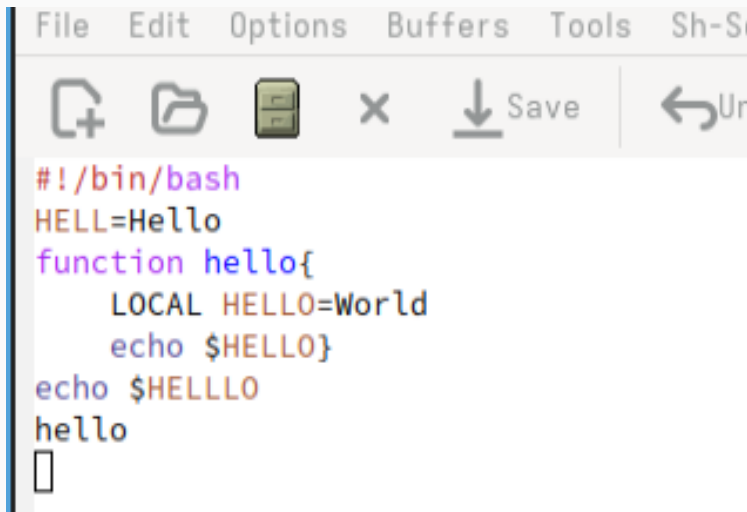


```
#!/bin/bash
HELL=Hello
function hello{
    LOCAL HELLO=World
    echo $HELLO}
echo $HELLLO
hello
echo $HELLO
```

Рис. 8: копирование и вставка

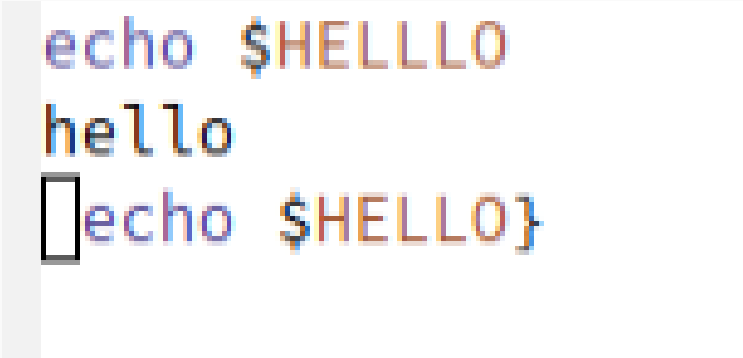
Основные команды с редактором

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:

A screenshot of a terminal window with a light gray background. It shows three lines of text: 'echo \$HELLO' on the first line, 'hello' on the second line, and 'echo \$HELLO}' on the third line. A black rectangular cursor is positioned at the start of the third line, before the 'e' in 'echo'.

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

Рис. 10: отмена действие

Основные команды с редактором

С помощью C-а можно переместить курсор в начало строки. С помощью C-е 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:

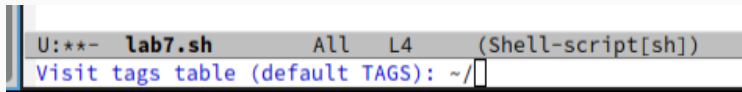
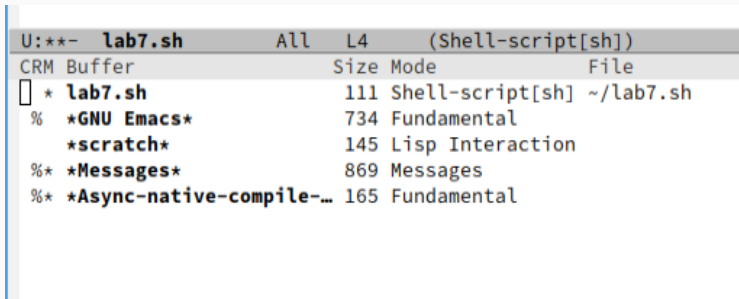


Рис. 12: Перемещение курсор в буфере

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 the active buffers. The first row is highlighted in grey and serves as a header: 'CRM Buffer', 'Size', 'Mode', and 'File'. The subsequent rows list buffers: '* lab7.sh' (111 bytes, Shell-script[sh], ~/lab7.sh), '% *GNU Emacs*' (734 bytes, Fundamental), '*scratch*' (145 bytes, Lisp Interaction), '%* *Messages*' (869 bytes, Messages), and '%* *Async-native-compile-...' (165 bytes, Fundamental).

U: **~ lab7.sh All L4 (Shell-script[sh])			
CRM Buffer	Size	Mode	File
<input 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. 13: Активные буферы

Основные команды 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:

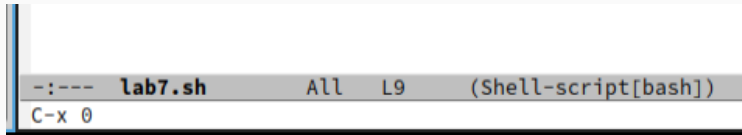
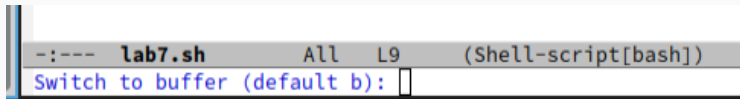


Рис. 15: Заккрытие окно

Без вывода списка буферов, я переключилась между буферами:



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

Рис. 16: Переключение между буферами

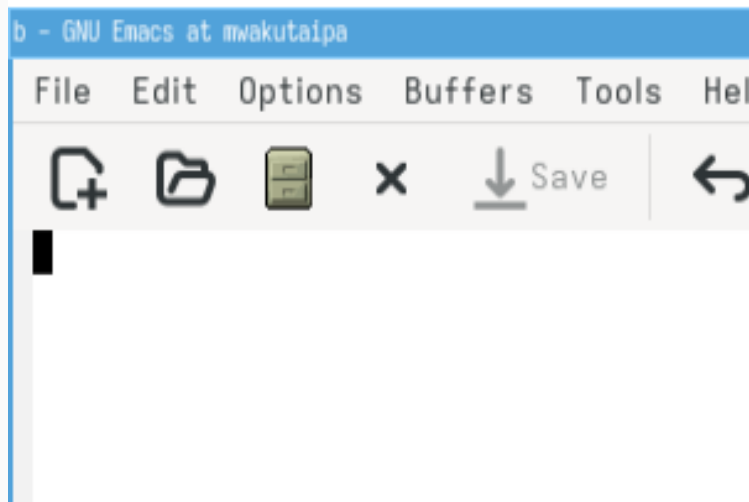


Рис. 17: Новый буфер

Основные команды 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):

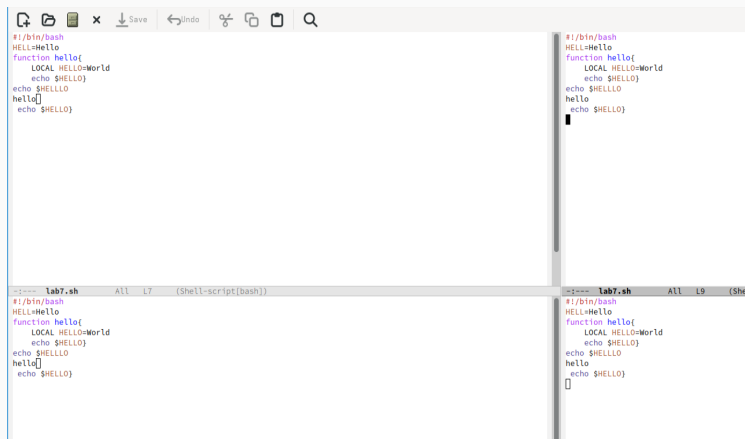


Рис. 18: Фрейм раздѣленный на 4 окна

Основные команды с редактором

В каждом из четырёх созданных окон открыла новый буфер:

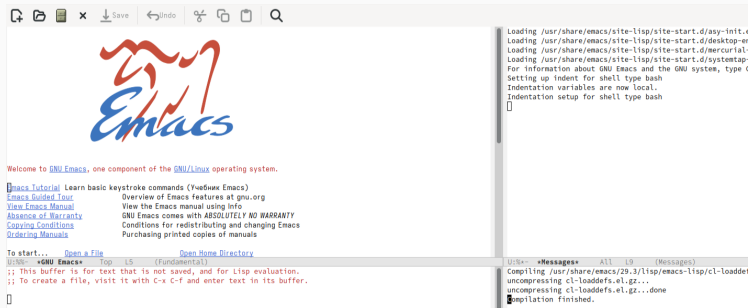


Рис. 19: Новые буферы

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

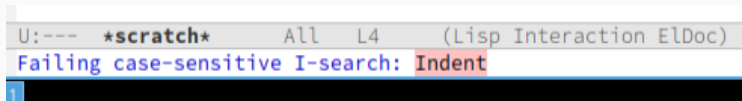


Рис. 20: Режим поиска

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

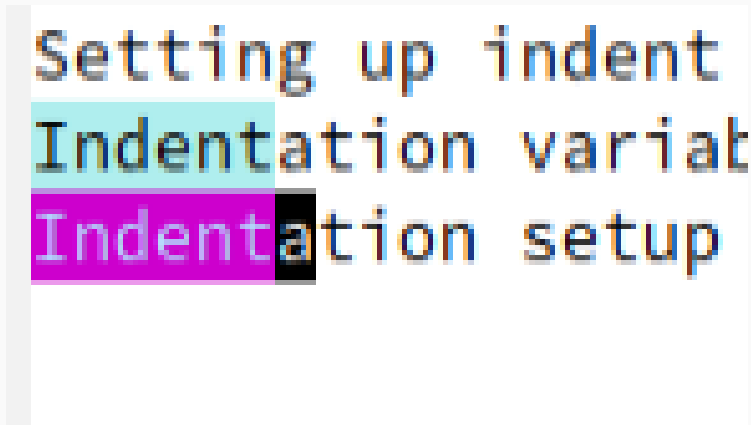


Рис. 21: Переключение между результатами

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

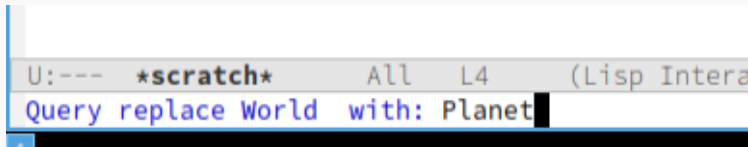


Рис. 22: Режим поиска

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

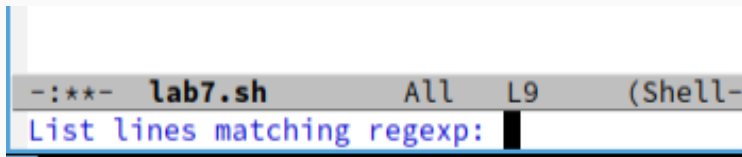
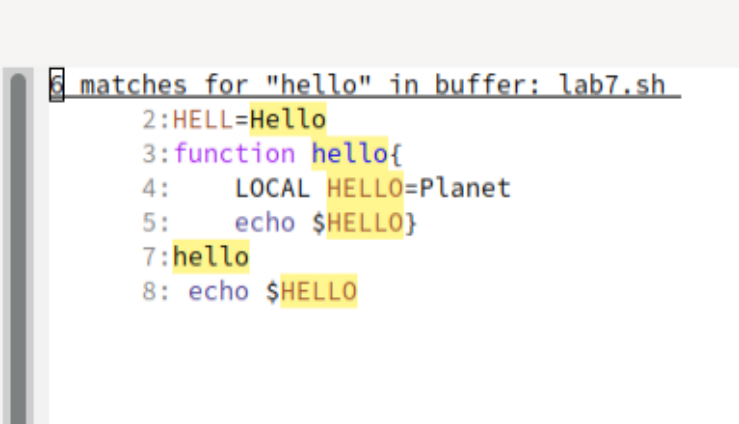


Рис. 23: другой режим поиска



```
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. 24: Результаты поиска

Выводы

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