

# Презентация по лабораторной работе №9

---

Паращенко Антонина

28 апреля 2022

РУДН, Москва, Россия

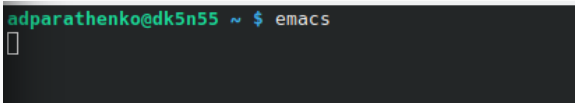
## Цель работы

---

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

## Ход лабораторной работы

---

A terminal window with a dark background. The prompt 'adparathenko@dk5n55 ~ \$' is shown in green, followed by the command 'emacs' in white. A white cursor is positioned at the start of a new line below the command.

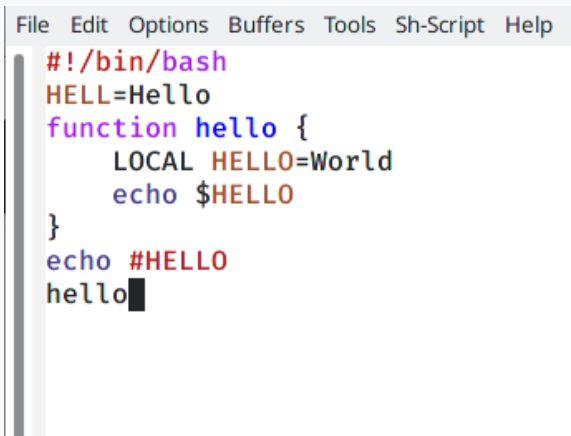
```
adparathenko@dk5n55 ~ $ emacs
```

Figure 1: emacs

A screenshot of the Emacs text editor interface. The central part of the image shows a grey rectangular buffer containing the text 'lab07.sh' in a bold, black, monospaced font. The buffer is set against a white background, which represents the Emacs window. The text 'lab07.sh' is centered within the buffer.

Figure 2: emacs

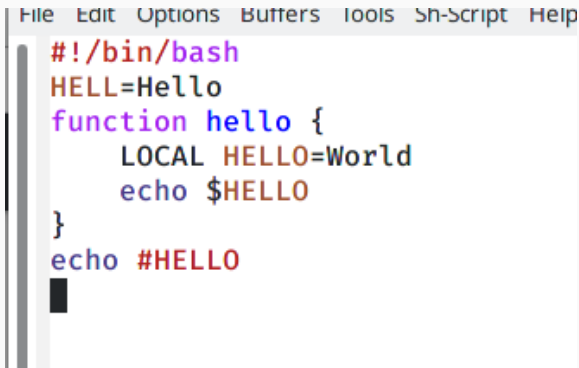
## Набираем данный текст

A screenshot of the Emacs text editor interface. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The text area shows a Bash script with syntax highlighting: '#!/bin/bash' in purple, 'HELL=Hello' in brown, 'function hello {' in purple, 'LOCAL HELLO=World' in brown, 'echo \$HELLO' in brown, '}' in purple, 'echo #HELLO' in brown, and 'hello' in black. A black cursor is positioned at the end of the word 'hello' on the last line.

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

Figure 3: emacs

Сохраняем файл с помощью комбинации Ctrl-x Ctrl-s




```
File Edit Options Buffers Tools Sn-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo #HELLO
█
```

Figure 4: Вырезать одной командой целую строку (C-k)



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

Figure 5: Вставить эту строку в конец файла (C-y)



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

Figure 6: Выделить область текста (C-space)

## Проделать с текстом стандартные процедуры редактирования

```
the Edit Options buffers tools scripts help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo #HELLO
helloHELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
```

U: \*\* lab07.sh All L8 (S

cause of the error in your initial  
the '--debug-init' option to view  
□

U: % \*\*Warnings\* Bot L8 (S

M-x wHELL=Hello  
function hello {  
 LOCAL HELLO=World  
 echo \$HELLO□

# Проделать с текстом стандартные процедуры редактирования

```
the Edit Options buffers tools scripts help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo #HELLO
helloHELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
```

U: \*\* lab07.sh All L8 (S

cause of the error in your initial  
the '--debug-init' option to view  
□

U: % \*\*Warnings\* Bot L8 (S

M-x wHELL=Hello  
function hello {  
 LOCAL HELLO=World  
 echo \$HELLO□

# Прodelать с текстом стандартные процедуры редактирования

File Edit Options Buffers Tools Sh-Script Help

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

U:\*\*\* lab07.sh All L8

cause of the error in your init  
the '--debug-init' option to vi

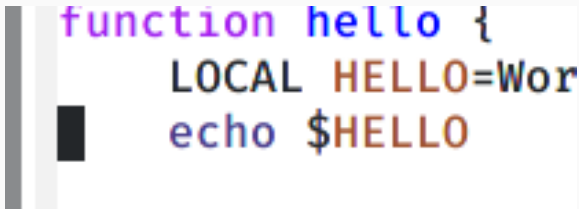
U:%\*- \*Warnings\* Bot L8

```
M-x wHELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
```



```
File Edit Options Buffers Tools Sn-Script He
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo #HELLO
helloHELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
```

Figure 10: Отмените последнее действие (C-/)



```
function hello {  
    LOCAL HELLO=Wor  
    echo $HELLO
```

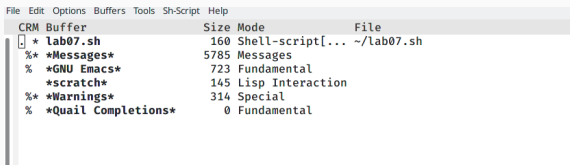
Figure 11: Переместите курсор в начало строки (C-a))

```
function hello {  
    LOCAL HELLO=Wor  
    echo $HELLO
```

Figure 12: Переместите курсор в конец строки (C-e)



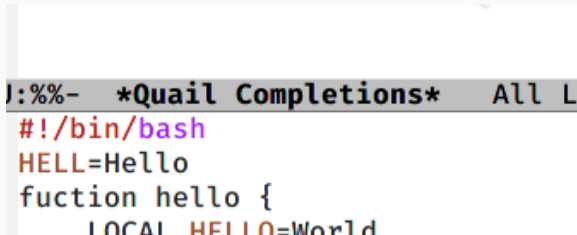
# Совершаем управление буферами



The screenshot shows the Emacs buffer list window, which is titled "CRM Buffer". The window displays a list of active buffers with their sizes, modes, and file names. The buffers listed are: \*lab07.sh (160 characters, Shell-script mode, file ~/lab07.sh), \*Messages\* (5785 characters, Messages mode), \*GNU Emacs\* (723 characters, Fundamental mode), \*scratch\* (145 characters, Lisp Interaction mode), \*Warnings\* (314 characters, Special mode), and \*Quail Completions\* (0 characters, Fundamental mode).

CRM Buffer	Size	Mode	File
* lab07.sh	160	Shell-script[...]	~/lab07.sh
% *Messages*	5785	Messages	
% *GNU Emacs*	723	Fundamental	
% *scratch*	145	Lisp Interaction	
% *Warnings*	314	Special	
% *Quail Completions*	0	Fundamental	

Figure 13: Вывести список активных буферов на экран (C-x C-b)

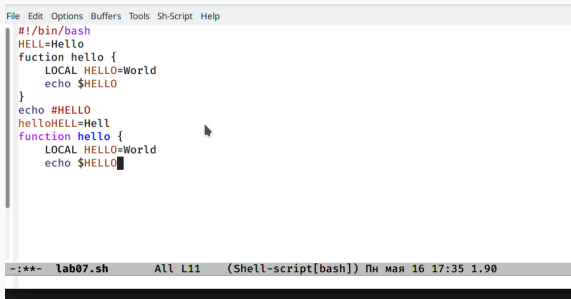


A screenshot of a terminal window showing the Quail Completions menu. The menu is displayed as a list of options, each preceded by a tab character. The options are: `#!/bin/bash` (highlighted in purple), `HELL=Hello` (highlighted in red), `fuction hello {` (highlighted in blue), and `LOCAL HELLO=World` (highlighted in red). The prompt `!:%%-` is visible on the left, and `*Quail Completions*` and `All L` are visible on the right.

```
!:%%-  *Quail Completions*  All L
#!/bin/bash
HELL=Hello
fuction hello {
LOCAL HELLO=World
```

Figure 14: Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер

# Совершаем управление буферами



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo #HELLO
helloHELL=Hell
function hello {
    LOCAL HELLO=World
    echo $HELLO
```

-:\*\*~ lab07.sh All L11 (Shell-script[bash]) Пн мая 16 17:35 1.90

Figure 15: Закройте это окно (C-x 0)

# Совершаем управление буферами

```
File Edit Options Buffers Tools Help
delete-backward-char: Text is read-only [6 times]
user-error: File /afs/.dk.sci.pfu.edu.ru/home/a/d/adparathenko/TAGS does not exist
Auto-saving...
user-error: Marker stack is empty [3 times]
delete-backward-char: Text is read-only
user-error: Marker stack is empty
completing-read-default: Command attempted to use minibuffer while in minibuffer [11 times]
user-error: Marker stack is empty [2 times]
completing-read-default: Command attempted to use minibuffer while in minibuffer [2 times]
user-error: Marker stack is empty [3 times]
completing-read-default: Command attempted to use minibuffer while in minibuffer [5 times]
user-error: Marker stack is empty [7 times]
completing-read-default: Command attempted to use minibuffer while in minibuffer [3 times]
Auto-saving...done
funcall-interactively: Beginning of buffer [18 times]
command-execute: Buffer is read-only: #<buffer *Messages*>
Mark set
user-error: Marker stack is empty [3 times]
U:~ *Messages* 78% L85 (Messages) Пн мая 16 17:36 1.62
C-x
```

Figure 16: Переключайтесь между буферами, но уже без вывода их списка на экран (C-x b)

# Управление окнами

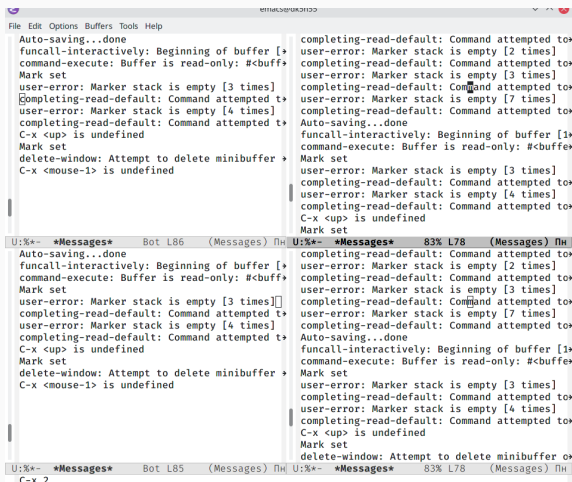


Figure 17: разделите фрейм на два окна по вертикали (C-x 3), на две части по горизонтали (C-x 2)



The image displays four terminal windows arranged in a 2x2 grid. Each window shows a shell script with the following content:

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

The top-left window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The bottom-left window has a title bar that reads '-:\*\*\*- lab07.sh All L12 (Shell-script)'. The bottom-right window has a title bar that reads '-:\*\*\*- lab07.sh All L8 (Shell-script)'. The top-right window does not have a visible title bar.

Figure 18: введите несколько строк текста



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

~:\*\*\* lab07.sh All L12 (Shell-script

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

~:\*\*\* lab07.sh All L8

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

~:\*\*\* lab07.sh All L5 (Shell-script

```
I-search: echo
```

~:\*\*\* lab07.sh All L8

Figure 19: Переключитесь в режим поиска (C-s) и найдите несколько слов,

```
HELLOHELLO=hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hell
```

\*\*\*- lab07.sh All

Wrapped I-search: echo

Figure 20: Переключайтесь между результатами поиска, нажимая C-s



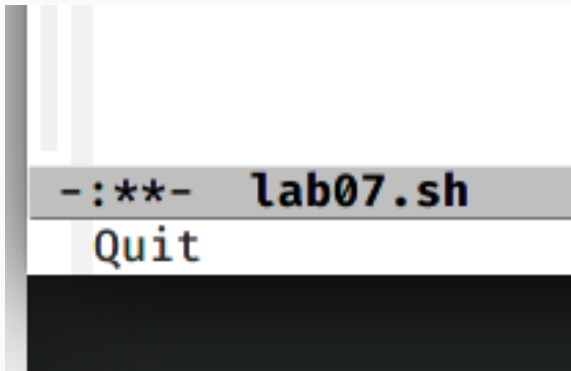


Figure 21: Выйдите из режима поиска, нажав C-g

## Вывод

---

Познакомилась с операционной системой Linux. Получила практические навыки работы с редактором Emacs.