

Лабораторная работа №9


Легиньких Галина - группа НФИбд-02-21

11.05.2022

Текстовый редактор emacs

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

1. Открыла emacs.(рис. 1)

A terminal window with a dark background. The prompt 'galeginfkikh@dk6n64 ~ \$' is shown in green. The command 'emacs' is entered in white. The prompt 'galeginfkikh@dk6n64 ~ \$' is shown again in green, followed by the letter 'e' in white, with a cursor at the end.

```
galeginfkikh@dk6n64 ~ $ emacs  
galeginfkikh@dk6n64 ~ $ e
```

Figure 1: emacs

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

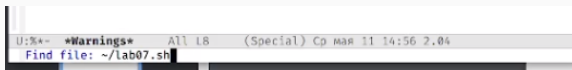
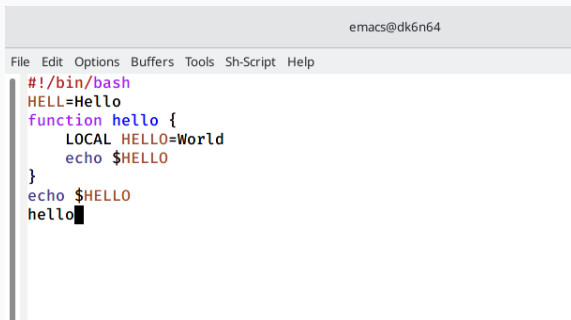


Figure 2: Создание файла

3. Набрала текст.(рис. 3)

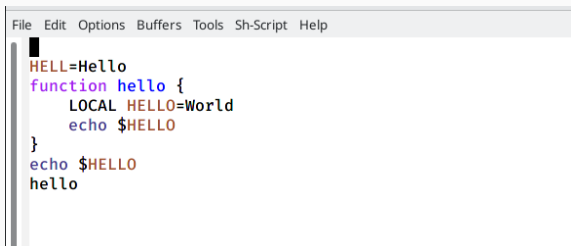
The image shows a screenshot of an Emacs editor window. The title bar at the top reads "emacs@dk6n64". Below the title bar is a menu bar with the following items: "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The main editing area contains a shell script with the following text:

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

The text is color-coded: the shebang line is purple, the variable assignment is brown, the function keyword is purple, the function name is blue, the local variable assignment is brown, the echo command is blue, and the variable names are brown. The cursor is positioned at the end of the last line, "hello".

Figure 3: Текст

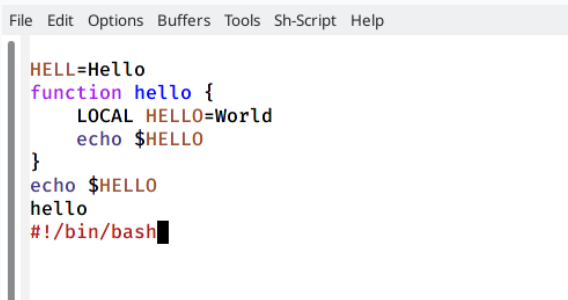
4. Сохранила файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
5. Проделала с текстом стандартные процедуры редактирования:
 - 5.1. Вырезала одной командой целую строку (C-k).(рис. 4)



```
File Edit Options Buffers Tools Sh-Script Help
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Figure 4: Вырезание строки

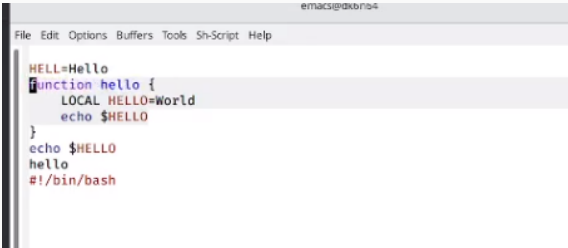
5.2. Вставила эту строку в конец файла (С-у).(рис. 5)

A screenshot of a text editor window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The editor displays a shell script with the following content: 'HELL=Hello', 'function hello {', ' LOCAL HELLO=World', ' echo \$HELLO', '}', 'echo \$HELLO', 'hello', and '#!/bin/bash'. The text is color-coded: 'HELL' is brown, 'Hello' is black, 'function' is purple, 'hello' is blue, 'LOCAL' is brown, 'HELLO' is black, 'World' is black, 'echo' is blue, '\$HELLO' is brown, and '#!/bin/bash' is red. A black cursor is positioned at the end of the last line.

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

Figure 5: Строка в конце файла

5.3. Выделила область текста (C-space).(рис. 6)

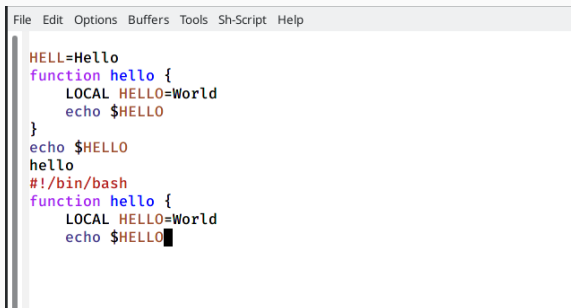


```
emacs@qkbn4
File Edit Options Buffers Tools Sh-Script Help
HELL=Hello
function hello {
  LOCAL HELLO=World
  echo $HELLO
}
echo $HELLO
hello
#!/bin/bash
```

Figure 6: Выделение текста

5.4. Скопировала область в буфер обмена (M-w).

5.5. Вставила область в конец файла.(рис. 7)



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

Figure 7: Область в конце файла

5.6. Вновь выделила эту область и на этот раз вырезала её (C-w).

5.7. Отменила последнее действие (C-/).

6. Научилась использовать команды по перемещению курсора:

6.1. Переместила курсор в начало строки (C-a).(рис. 8)

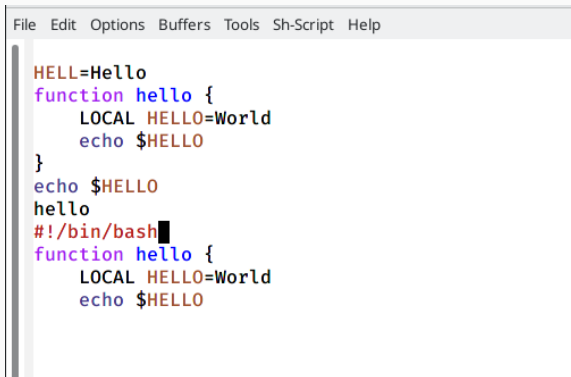
A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The terminal displays a script with the following lines: 'HELL=Hello', 'function hello {', ' LOCAL HELLO=World', ' echo \$HELLO', '}', 'echo \$HELLO', 'hello', and '#!/bin/bash'. The script is repeated once more. A black cursor is positioned at the beginning of the second 'function hello {' line. The text is color-coded: 'HELL' is brown, 'Hello' is black, 'function' is purple, 'hello' is black, 'LOCAL' is brown, 'HELLO=World' is black, 'echo' is purple, '\$HELLO' is black, and '#!/bin/bash' is red.

```
File Edit Options Buffers Tools Sh-Script Help

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

Figure 8: Курсор в начале строки

6.2. Переместила курсор в конец строки (C-e).(рис. 9)



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

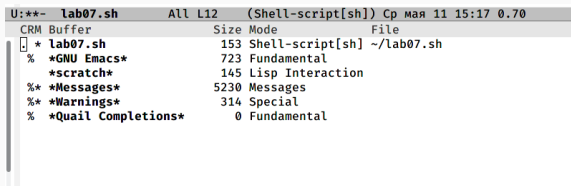
Figure 9: Курсор в конце строки

6.3. Переместила курсор в начало буфера (M-<).

6.4. Переместила курсор в конец буфера (M->).

7. Научилась управлять буфером:

7.1. Вывела список активных буферов на экран (C-x C-b).(рис. 10)

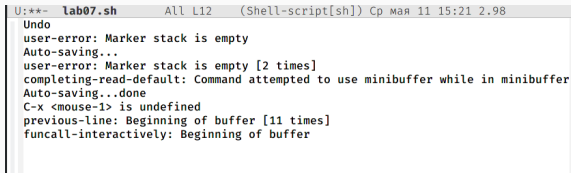
The image shows a screenshot of the Emacs buffer list command output. At the top, a status bar reads 'U:**~ lab07.sh All L12 (Shell-script[sh]) Ср мая 11 15:17 0.70'. Below this is a table with columns: CRM, Buffer, Size, Mode, and File. The table lists several buffers: 'lab07.sh' (153 bytes, Shell-script[sh], ~/lab07.sh), '*GNU Emacs*' (723 bytes, Fundamental), '*scratch*' (145 bytes, Lisp Interaction), '*Messages*' (5230 bytes, Messages), '*Warnings*' (314 bytes, Special), and '*Quail Completions*' (0 bytes, Fundamental).

U:**~ lab07.sh All L12 (Shell-script[sh]) Ср мая 11 15:17 0.70				
CRM	Buffer	Size	Mode	File
	* lab07.sh	153	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	723	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	5230	Messages	
%*	*Warnings*	314	Special	
%	*Quail Completions*	0	Fundamental	

Figure 10: Буфер

7.2. Переместилась во вновь открытое окно (C-x) со списком открытых буферов и переключилась на другой буфер.

7.3. Закрала это окно (C-x 0).(рис. 11)



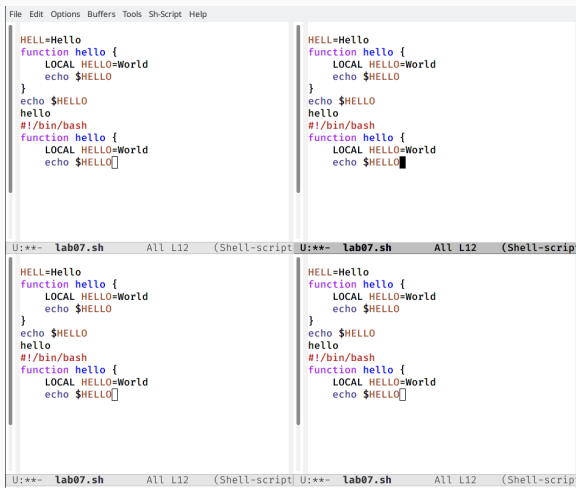
```
U:*** lab07.sh All L12 (Shell-script[sh]) Ср мая 11 15:21 2.98
Undo
user-error: Marker stack is empty
Auto-saving...
user-error: Marker stack is empty [2 times]
completing-read-default: Command attempted to use minibuffer while in minibuffer
Auto-saving...done
C-x <mouse-1> is undefined
previous-line: Beginning of buffer [11 times]
funcall-interactively: Beginning of buffer
```

Figure 11: Закрала окно

7.4. Теперь вновь переключилась между буферами, но уже без вывода их списка на экран (С-х b).

8. Управление окнами:

8.1. Поделила фрейм на 4 части: разделила фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2).(рис. 12)



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

8.2. В каждом из четырёх созданных окон открыла новый буфер (файл) и ввела несколько строк текста.(рис. 13)

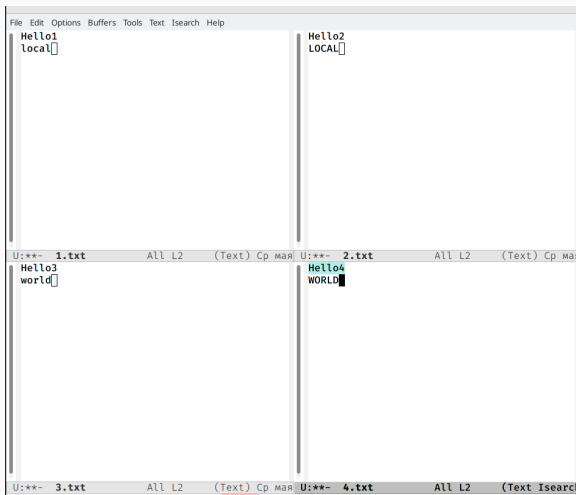


Figure 13: Текст

9. Режим поиска:

9.1. Переключилась в режим поиска (C-s) и нашла несколько слов, присутствующих в тексте.(рис. 14)(рис. 15)

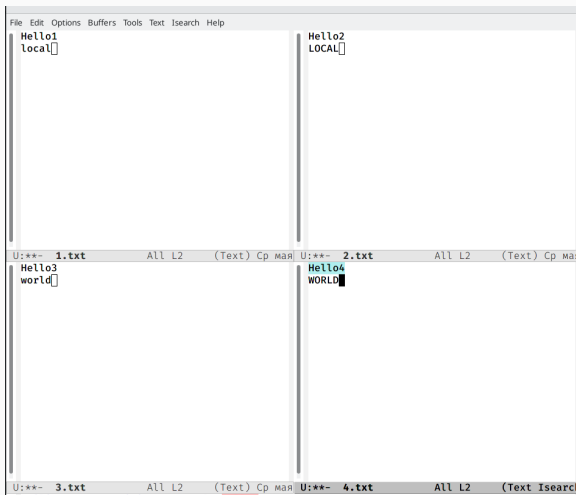


Figure 14: Поиск

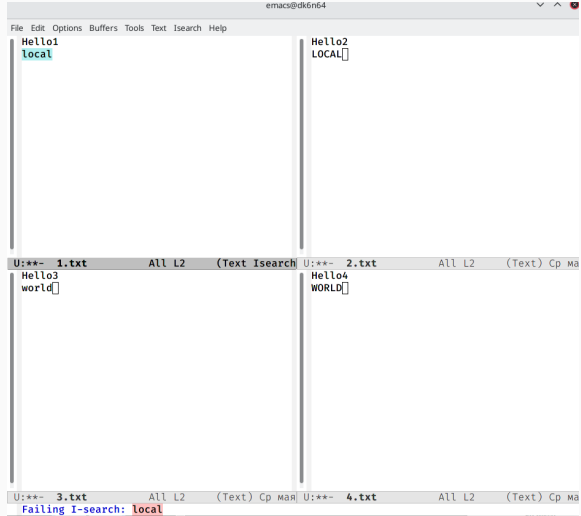


Figure 15: Режим поиска

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

9.3. Вышла из режима поиска, нажав C-g.

9.4. Перешла в режим поиска и замены (M-%), ввела текст, который следует найти и заменила, нажала Enter , затем ввела текст для замены. После того как были подсвечены результаты поиска, нажала ! для подтверждения замены.

9.5. Испробовала другой режим поиска, нажав M-s o.

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