

# Презентация по лабораторной работе №9

---

Шмырин Михаил Сергеевич (группа: НПМбд-02-21)

Российский Университет Дружбы Народов

## Цель работы

---

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

## Выполнение лабораторной работы

---

# Выполнение лабораторной работы

1. Открываю emacs.(рис. 1) (рис. 2)

```
midhs@msshmihrin:~$ emacs &  
[1] 4489  
midhs@msshmihrin:~$ emacs &  
[2] 4667  
[1]   Завершён   emacs
```

Figure 1: Emacs

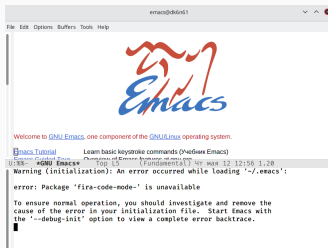


Figure 2: Emacs

2. Создаю файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (рис. 3)

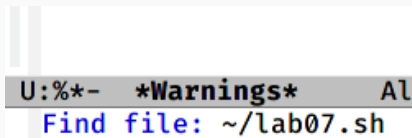


Figure 3: Создание файла lab07

3. Набираю текст: (рис. 4)



The image shows a screenshot of a GNU Emacs editor window. The title bar at the top reads "U:%%- \*GNU Emacs\* Top L8". The main text area contains a Bash script with the following content: 

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

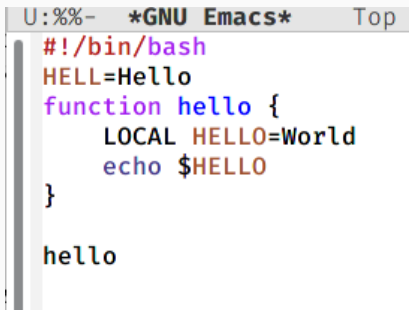
 The text is color-coded: the shebang is purple, variable assignments are brown, function definitions are purple, and the function body is brown. The bottom status bar shows "U:%%- lab07.sh All L8".

Figure 4: Ввод текста

4. Сохраняю файл с помощью комбинации Ctrl-x Ctrl-s.

5. Проделываю с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.

5.1. Вырезаю одной командой целую строку (C-k).(рис. 5)



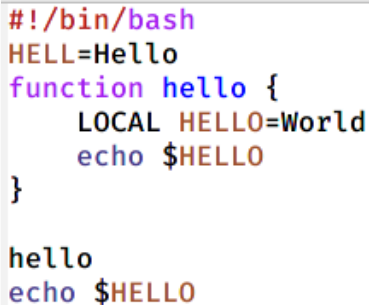
```
U:%%- *GNU Emacs* Top
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
```

Figure 5: Вырезаем целую строку



## Вставляем строку в конец файла

5.2. Вставляю эту строку в конец файла (С-у).(рис. 6)

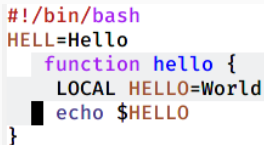


```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}

hello
echo $HELLO
```

Figure 6: Вставляем строку в конец файла

5.3. Выделяю область текста (C-space).(рис. 7)



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

Figure 7: Выделяем область текста

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

## Вставляем область в конец файла

5.5. Вставляю область в конец файла. (рис. 8)



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

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

Figure 8: Вставляем область в конец файла

5.6. Вновь выделяю эту область и на этот раз вырезаю её (C-w).9)

```
#!/bin/bash
HELL=Hello

    echo $HELLO
}

hello
echo $HELLO
function hello {
    LOCAL HELLO=World
    █
```

Figure 9: Вырезаем область

### 5.7. Отменяю последнее действие (C-/).(рис. 10)



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

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

Figure 10: Отмена последнего действия

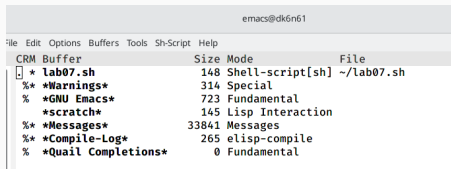
6.1. Перемещаю курсор в начало строки (C-a).

6.2. Перемещаю курсор в конец строки (C-e).

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

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

## 7.1. Вывожу список активных буферов на экран (C-x C-b) (рис. 11)

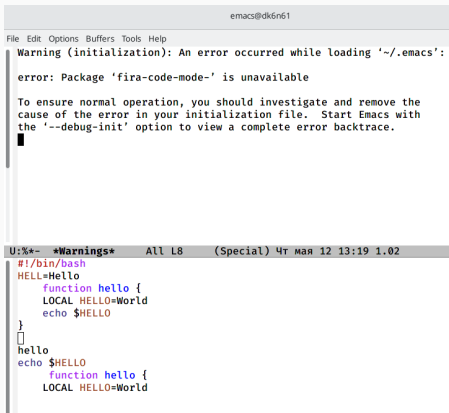
A screenshot of the Emacs buffer list window. The title bar shows 'emacs@dk6n61'. The menu bar includes 'File Edit Options Buffers Tools Sh-Script Help'. The buffer list table has columns 'CRM', 'Buffer', 'Size', 'Mode', and 'File'. The buffers listed are: '\* lab07.sh' (148, Shell-script[sh], ~/lab07.sh), '%\* \*Warnings\*' (314, Special), '% \*GNU Emacs\*' (723, Fundamental), '\*scratch\*' (145, Lisp Interaction), '%\* \*Messages\*' (33841, Messages), '%\* \*Compile-Log\*' (265, elisp-compile), and '% \*Quail Completions\*' (0, Fundamental).

CRM	Buffer	Size	Mode	File
<input checked="" type="checkbox"/>	* lab07.sh	148	Shell-script[sh]	~/lab07.sh
<input type="checkbox"/>	%* *Warnings*	314	Special	
<input type="checkbox"/>	% *GNU Emacs*	723	Fundamental	
<input type="checkbox"/>	*scratch*	145	Lisp Interaction	
<input type="checkbox"/>	%* *Messages*	33841	Messages	
<input type="checkbox"/>	%* *Compile-Log*	265	elisp-compile	
<input type="checkbox"/>	% *Quail Completions*	0	Fundamental	

Figure 11: Вывод списка активных буферов

# Переключение на другой буфер

7.2. Переместитесь во вновь открытое окно (С-х) о со списком открытых буферови переключитесь на другой буфер.(рис. 12)



The screenshot shows the Emacs editor interface. At the top, the title bar reads 'emacs@dk6n61'. Below it is the menu bar with 'File Edit Options Buffers Tools Help'. The main text area displays a warning message: 'Warning (initialization): An error occurred while loading '~/.emacs': error: Package 'fira-code-mode-' is unavailable'. Below the warning, there is a paragraph of text: 'To ensure normal operation, you should investigate and remove the cause of the error in your initialization file. Start Emacs with the '--debug-init' option to view a complete error backtrace.' followed by a cursor. At the bottom, a status bar shows 'U:%\*- \*Warnings\* All L8 (Special) Чт мая 12 13:19 1.02'. Below the status bar, the buffer content is shown, starting with '#!/bin/bash' and followed by a function definition and its execution.

```
emacs@dk6n61
File Edit Options Buffers Tools Help
Warning (initialization): An error occurred while loading '~/.emacs':
error: Package 'fira-code-mode-' is unavailable

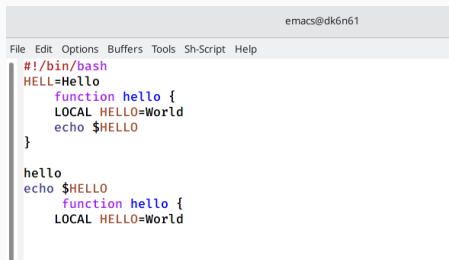
To ensure normal operation, you should investigate and remove the
cause of the error in your initialization file. Start Emacs with
the '--debug-init' option to view a complete error backtrace.
█

U:%*- *Warnings* All L8 (Special) Чт мая 12 13:19 1.02
#!/bin/bash
HELL=Hello
function hello {
  LOCAL HELLO=World
  echo $HELLO
}
hello
echo $HELLO
function hello {
  LOCAL HELLO=World
```

Figure 12: Переключение на другой буфер



## 7.3. Закрываю это окно (C-x 0).(рис. 13)

A screenshot of an Emacs editor window. The title bar at the top reads 'emacs@dk6n61'. Below it is a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The main text area contains a shell script with the following content:

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

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

Figure 13: Закрываю окно со списком буферов

7.4. Теперь вновь переключаюсь между буферами, но уже без вывода их списка на экран (C-x b).(рис. 14)

A screenshot of a terminal window. The prompt 'Switch to buffer (default \*Warnings\*):' is displayed in blue text. A black cursor is positioned at the end of the prompt. The background of the terminal is light gray, and the text is in a monospaced font.

```
Switch to buffer (default *Warnings*):
```

Figure 14: Переключение на другой буфер

8.1. Делю фрейм на 4 части: разделите фрейм на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2)(рис. 15)

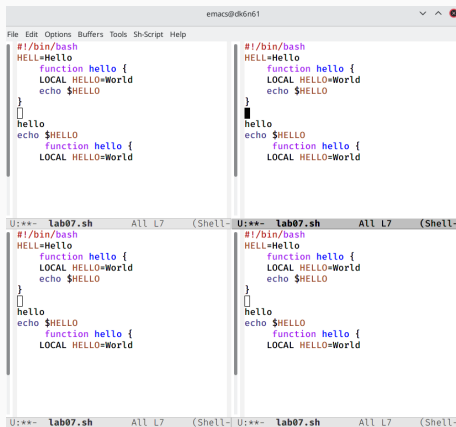


Figure 15: Деление фрейма на 4 части

8.2. В каждом из четырёх созданных окон открываю новый буфер (файл) и ввожу текст.(рис. 16)

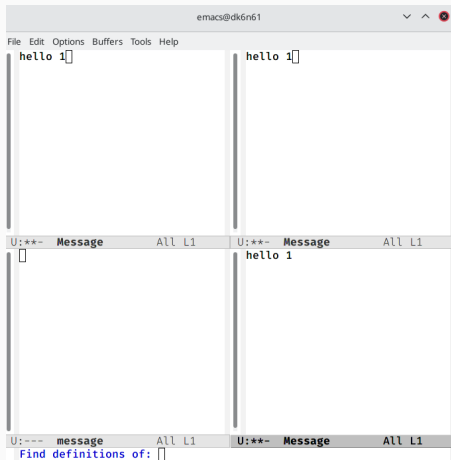


Figure 16: Ввод текста

9.1. Переключаюсь в режим поиска (C-s) и нахожу несколько слов, присутствующих в тексте. (рис. 17)



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

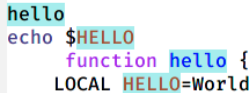
hello
echo $HELLO
function hello {
    LOCAL HELLO=World
```

Figure 17: Режим поиска

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

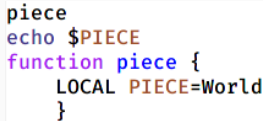
9.3. Выхожу из режима поиска, нажав C-g.

9.4. Перехожу в режим поиска и замены (M-%), ввожу текст, который следует найти и заменить, нажимаю Enter, затем ввожу текст для замены. После того как будут подсвечены результаты поиска(рис. 18), нажимаю ! для подтверждения замены.(рис. 19)



A screenshot of a code editor with a light gray background. The text is as follows:  
hello  
echo \$HELLO  
function hello {  
LOCAL HELLO=World  
The word 'hello' on the first line is highlighted in light blue. The word 'HELLO' in the second line is highlighted in light orange. The word 'hello' in the third line is highlighted in light blue. The word 'HELLO' in the fourth line is highlighted in light orange.

Figure 18: Результат поиска



A screenshot of a code editor with a light gray background. The text is as follows:  
piece  
echo \$PIECE  
function piece {  
LOCAL PIECE=World  
}  
The word 'piece' on the first line is highlighted in light blue. The word 'PIECE' in the second line is highlighted in light orange. The word 'piece' in the third line is highlighted in light blue. The word 'PIECE' in the fourth line is highlighted in light orange.

Figure 19: Замена

## Выводы

---

Я познакомился с операционной системой Linux, получил практические навыки работы с редактором Emacs.