

Презентация по лабораторной работе №9

Операционные системы

Сячинова Ксения Ивановна, НММбд-03-22

10 марта 2023

Российский университет дружбы народов, Москва, Россия

Цель работы

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

Задание

1. Ознакомиться с теоретическим материалом.
2. Ознакомиться с редактором emacs.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы.

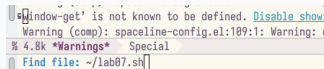
Выполнение лабораторной работы

1. Для работы в “Emacs” необходимо установить его. Открываем его с помощью команды “emacs &”



Рис. 1: Запуск emacs

2. Создаём файл lab07.sh и с помощью комбинаций “ctrl-x”, “ctrl-f” открываем его.

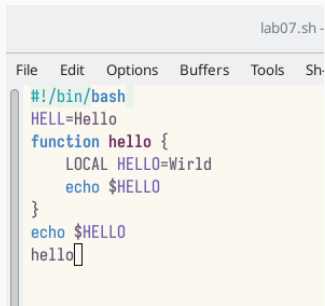


The screenshot shows a terminal window with a light blue background. The first line displays a warning: "window-get' is not known to be defined. [Disable show](#)". The second line shows another warning: "Warning (comp): spaceline-config.el:109:1: Warning:". The third line shows a summary of warnings: "% 4.8k *Warnings* Special". The fourth line shows the command "Find file: ~/lab07.sh" with a cursor at the end.

```
Warning (comp): spaceline-config.el:109:1: Warning:  
% 4.8k *Warnings* Special  
Find file: ~/lab07.sh
```

Рис. 2: Создание файла

3. Напишем необходимый текст.

A screenshot of a terminal window with a light gray title bar that says "lab07.sh -". Below the title bar is a menu bar with the items "File", "Edit", "Options", "Buffers", "Tools", and "Sh-". The main area of the terminal has a yellow background and contains the following text:

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

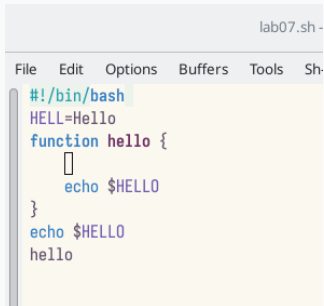
 The text is color-coded: the shebang is blue, the variable assignment is purple, the function keyword is blue, the function name is purple, the opening brace is blue, the local variable assignment is purple, the echo command is blue, the closing brace is blue, the second echo command is blue, and the function call is black. A cursor is visible at the end of the "hello" line.

Рис. 3: Текст

4. Сохраняем файл с помощью комбинаций клавиш “ctrl-x”, “ctrl-s”

5. Выполним ряд действий:

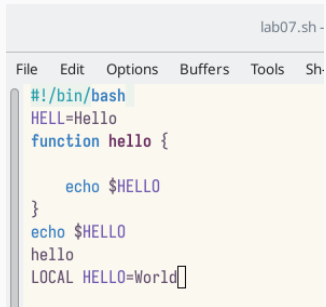
- 5.1. Вырезать одной командой целую строку (C-k).



```
lab07.sh -
File Edit Options Buffers Tools Sh-
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
```

Рис. 4: Вырезаем строку

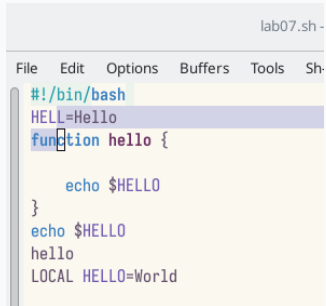
- 5.2. Вставить эту строку в конец файла (С-у).



```
lab07.sh -  
File Edit Options Buffers Tools Sh-  
#!/bin/bash  
HELL=Hello  
function hello {  
    echo $HELLO  
}  
echo $HELLO  
hello  
LOCAL HELLO=World
```

Рис. 5: Вставляем строку в конец

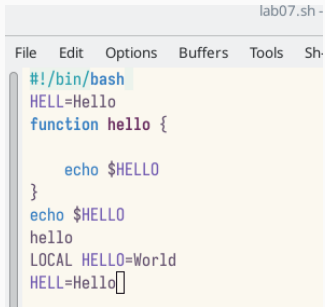
- 5.3. Выделить область текста (C-space)



```
lab07.sh -
File Edit Options Buffers Tools Sh-
#!/bin/bash
HELLO=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
```

Рис. 6: Вставляем строку в конец

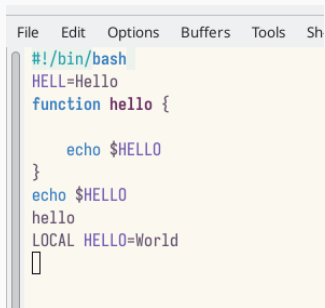
- 5.4. Скопировать область в буфер обмена (M-w).
- 5.5. Вставить область в конец файла.



```
lab07.sh -
File Edit Options Buffers Tools Sh-
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
HELL=Hello
```

Рис. 7: Вставка области

- 5.6. Вновь выделить эту область и на этот раз вырезать её (C-w).



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

Рис. 8: Вырезаем область

- 5.7. Отмените последнее действие (C-/).



```
File Edit Options Buffers To
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
HELL=Hello
```

Рис. 9: Отмена действия

6. Действия с курсором

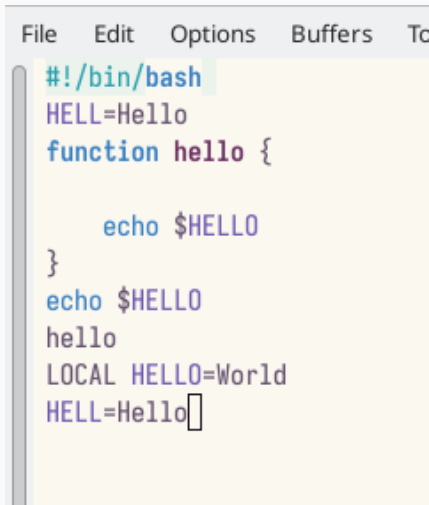
- 6.1. Переместите курсор в начало строки (C-a).



```
File Edit Options Buffers To
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
HELL=Hello
```

Рис. 10: В начало строки

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



```
File Edit Options Buffers To
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
HELL=Hello
```

Рис. 11: В конец строки

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



The image shows a screenshot of a text editor window. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', and 'To'. The text area has a yellow background and contains a shell script. The first line, `#!/bin/bash`, is highlighted in light blue, and a black cursor is positioned at the very beginning of this line. The rest of the script is as follows:

```
#!/bin/bash
HELL=Hello
function hello {

    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
HELL=Hello
```

Рис. 12: В начало буфера

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

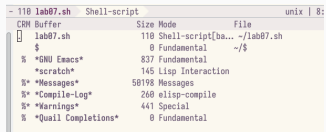


```
File Edit Options Buffers To
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
HELL=Hello
```

Рис. 13: В конец буфера

7. Управление буферами.

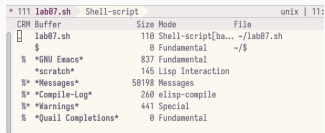
- 7.1. Вывести список активных буферов на экран (C-x C-b).

A screenshot of the Emacs buffer list window, titled "110 lab07.sh Shell-script unix | 8:". The window displays a table of active buffers. The first buffer is "lab07.sh" with size 110, mode "Shell-script", and file path "~/lab07.sh". The second buffer is "\$" with size 0, mode "Fundamental", and file path "~/\$. Below these are several wildcard buffers: "*GNU Emacs*" (size 837, mode "Fundamental"), "*scratch*" (size 145, mode "Lisp Interaction"), "*Messages*" (size 50198, mode "Messages"), "*Compile-Log*" (size 260, mode "elisp-compile"), "*Warnings*" (size 441, mode "Special"), and "*Quail Completions*" (size 0, mode "Fundamental").

CRN Buffer	Size	Mode	File
lab07.sh	110	Shell-script	~/lab07.sh
\$	0	Fundamental	~/
% *GNU Emacs*	837	Fundamental	
% *scratch*	145	Lisp Interaction	
% *Messages*	50198	Messages	
% *Compile-Log*	260	elisp-compile	
% *Warnings*	441	Special	
% *Quail Completions*	0	Fundamental	

Рис. 14: Список активных буферов

- 7.2. Переместим во вновь открытое окно (С-х о) со списком открытых буферов и переключимся на другой буфер (для этого нажмём “enter”)



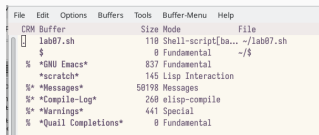
The screenshot shows the Emacs buffer list window. The title bar reads '* 111 lab07.sh Shell-script unix | 11:'. The window contains a table with four columns: 'CRN Buffer', 'Size', 'Mode', and 'File'. The first row is highlighted with a blue background. The table lists several buffers, including 'lab07.sh', '\$', '*GNU Emacs*', '*scratch*', '*Messages*', '*Compile-Log*', '*Warnings*', and '*Quail Completions*'. The 'lab07.sh' buffer is the first one in the list and is highlighted.

CRN Buffer	Size	Mode	File
lab07.sh	110	Shell-script[ba...	~/lab07.sh
\$	0	Fundamental	~/
% *GNU Emacs*	837	Fundamental	
% *scratch*	145	Lisp Interaction	
%* *Messages*	50198	Messages	
%* *Compile-Log*	260	elisp-compile	
%* *Warnings*	441	Special	
% *Quail Completions*	0	Fundamental	

Рис. 15: Переключение на другой буфер

- 7.3. Закройте это окно (C-x 0).

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



File	Edit	Options	Buffers	Tools	Buffer-Menu	Help
CRM Buffer			Size	Mode	File	
			lab07.sh	110	Shell-script[ba...	~/lab07.sh
			\$	0	Fundamental	~/
		%	*GNU Emacs*	837	Fundamental	
		%*	*scratch*	145	Lisp Interaction	
		%*	*Messages*	50198	Messages	
		%*	*Compile-log*	260	elisp-compile	
		%*	*Warnings*	441	Special	
		%	*Quail Completions*	0	Fundamental	

Рис. 16: Переключение между буферами

8. Управление окнами.

- 8.1. Поделите 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)

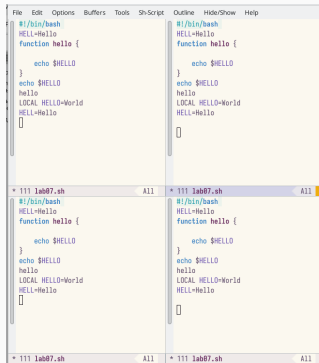


Рис. 17: Разделение окна на 4 части

- 8.2. В каждом из четырёх созданных окон откроем новый буфер (файл) и введём несколько строк текста. Для этого я заранее создала 4 файла с разным текстом.

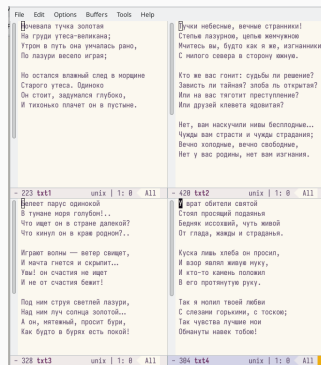
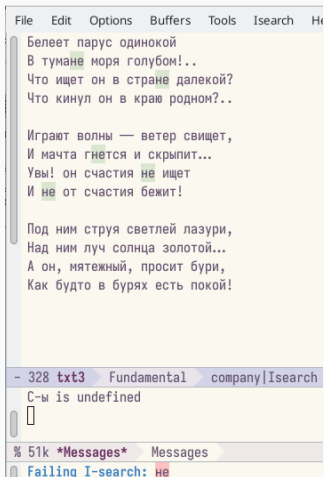


Рис. 18: Новые файлы

9. Режим поиска

- 9.1. Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте.



The screenshot shows the Emacs editor interface. The menu bar at the top includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Isearch', and 'H'. The main text area contains a poem in Russian. The word 'не' is highlighted in green in several lines. At the bottom, the Isearch status bar shows the search term 'не' in red. The command history at the bottom left shows 'C-ы is undefined'.

```
File Edit Options Buffers Tools Isearch H
Белеет парус одинокой
В тумане моря голубом!..
Что ищет он в стране далекой?
Что кинул он в краю родном!..

Играют волны — ветер свищет,
И мачта гнется и скрипит...
Увы! он счастья не ищет
И не от счастья бежит!

Под ним струя светлей лазури,
Над ним луч солнца золотой...
А он, мятежный, просит бури,
Как будто в бурях есть покой!
```

- 328 txt3 Fundamental company|Isearch

C-ы is undefined

% 51k *Messages* Messages

Failing I-search: не

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

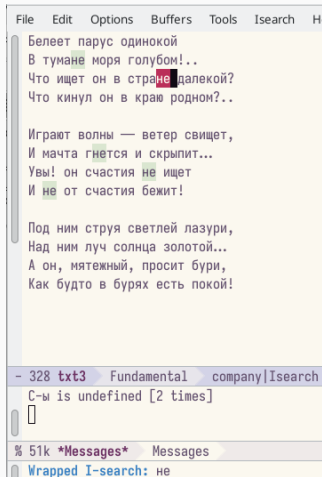
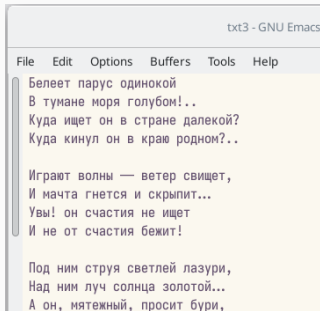


Рис. 20: Переход

- 9.3. Выйдите из режима поиска, нажав C-g.
- 9.4. Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter , затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены.



Рис. 21: Замена слов



- 9.5. Попробуем режим поиска (M-s o). Данный поиск отличается тем, что тут считывается строка поиска, которая трактуется как регулярное выражение, и не осуществляется поиск точно совпадения в тексте буфера. Регулярное выражение - это образец, который обозначает набор строк, возможно, и неограниченный набор

Выводы

В ходе выполнения данной лабораторной работы я познакомилась с операционной системой Linux и получила практические навыки по работе с редактором Emacs.