

Отчёт по лабораторной работе №9

Сячинова Ксения Ивановна, НПМбд-02-21

Российский Университет Дружбы Народов

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

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

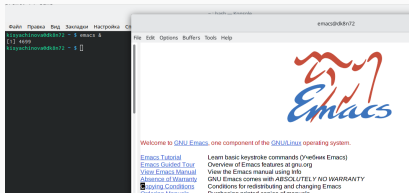


Figure 1: Запуск Emacs

2. Создаём файл lab07.sh с помощью комбинаций “ctrl-x”,”ctrl-f”.(рис. 2)

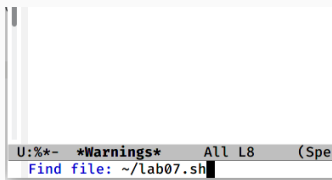


Figure 2: Создание файла

3. Напишем необходимый текст. (рис. 3)

A screenshot of a GNU Emacs editor window. The title bar at the top reads "U:%%- *GNU Emacs* Top". 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 script is written in a monospaced font with syntax highlighting: the shebang is red, the function name is purple, the function body is black, and the echo command is blue. The cursor is positioned at the end of the last line, "hello".

Figure 3: Текст

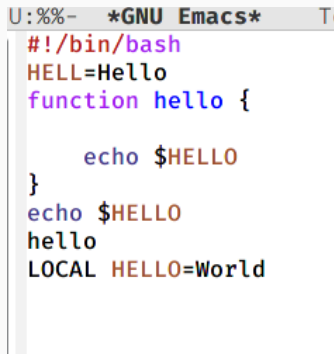
4. Сохраняем файл с помощью комбинаций клавиш “ctrl-x”, “ctrl-s”
5. Далее проделаем с файлом определённые действия
 - 5.1. Вырезать одной командой целую строку (C-k).(рис. 4)



```
Ordering Manuals  PU
U:%%- *GNU Emacs*
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
```

Figure 4: Вырезаем строку

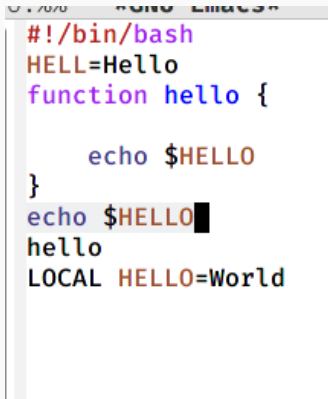
- 5.2. Вставить эту строку в конец файла (C-y).(рис. 5)



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

Figure 5: Вставляем строку в конец файла

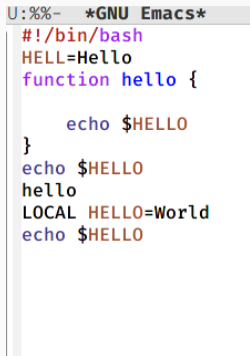
- 5.3. Выделить область текста (C-space).(рис. 6)

A screenshot of a terminal window with a light gray title bar. The terminal displays a Bash script with syntax highlighting: red for the shebang, blue for keywords, and brown for variable names. A gray selection bar highlights the text 'echo \$HELLO' on the line following the function definition. The script content is as follows:

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

Figure 6: Выделение текста

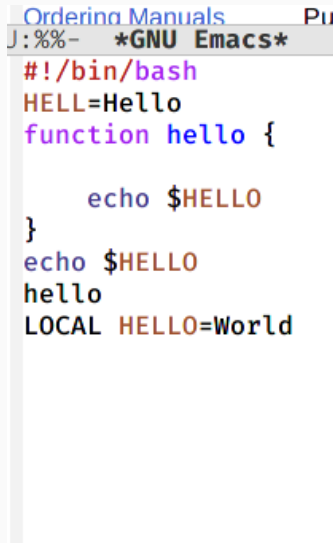
- 5.4. Скопировать область в буфер обмена (M-w).
- 5.5. Вставить область в конец файла(рис. 7)



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

Figure 7: Вставление текста

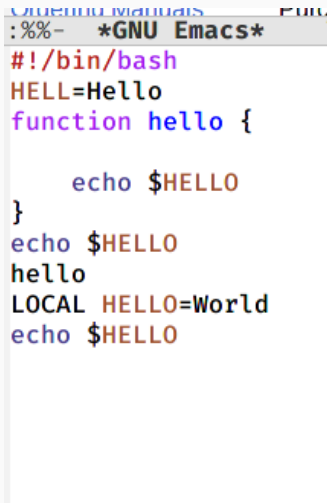
- 5.6. Вновь выделить эту область и на этот раз вырезать её (C-w).(рис. 8)



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

Figure 8: Вырезание текств

- 5.7. Отмените последнее действие (C-/).(рис. 9)



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

Figure 9: Отмена действия

6. Действия с курсором.

- 6.1. Перемещаем курсор в начало строки (C-a), (рис. 10)



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

The image shows a terminal window with a shell script. The script defines a function 'hello' that echoes the value of the environment variable 'HELLO'. It then calls the function and sets a local variable 'HELLO' to 'World'. The cursor is positioned at the beginning of the last line, 'echo \$HELLO'.

Figure 10: Перемещение курсора в начало строки

- 6.2. Перемещаем курсор в конец строки (C-e), (рис. 11)

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

Figure 11: Перемещение курсора в конец строки

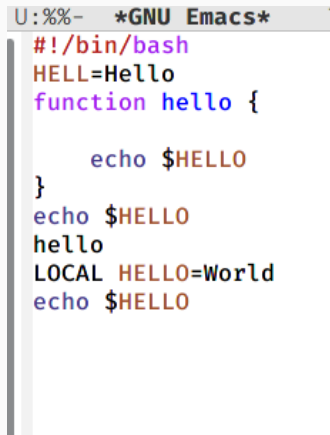
- 6.3. Перемещаем курсор в начало буфера (M-<), (рис. 12)



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

Figure 12: Перемещение курсора в начало буфера

- 6.4. Перемещаем курсор в конец буфера (M->), (рис. 13)

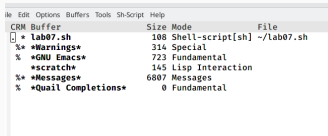


```
U:%%- *GNU Emacs*
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
echo $HELLO
```

Figure 13: Перемещение курсора в конец буфера

7. Управление буферами.

- 7.1. Выведем список активных буферов на экран (C-x C-b).(рис. 14)

A screenshot of the Emacs buffer list window, which is titled 'CRM Buffer'. The window shows a table of active buffers. The columns are 'Buffer', 'Size', 'Mode', and 'File'. The buffers listed are: 'lab07.sh' (Size 108, Mode Shell-script[sh], File ~/lab07.sh), '*Warnings*' (Size 314, Mode Special), '*GNU Emacs*' (Size 723, Mode Fundamental), '*scratch*' (Size 145, Mode Lisp Interaction), '*Messages*' (Size 6807, Mode Messages), and '*Quail Completions*' (Size 0, Mode Fundamental).

CRM Buffer	Size	Mode	File
[] * lab07.sh	108	Shell-script[sh]	~/lab07.sh
%* *Warnings*	314	Special	
% *GNU Emacs*	723	Fundamental	
% *scratch*	145	Lisp Interaction	
%* *Messages*	6807	Messages	
% *Quail Completions*	0	Fundamental	

Figure 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”).(рис. 15)

```

:***- lab07.sh      All L13      (Shell-script[bash]) Пр ма
CRM Buffer          Size Mode      File
% lab07.sh          110 Shell-script[... ~/lab07.s
% *GNU Emacs*       723 Fundamental
% *scratch*          145 Lisp Interaction
%* *Messages*        4045 Messages
%* *Warnings*         314 Special
% *Quail Completions* 0 Fundamental

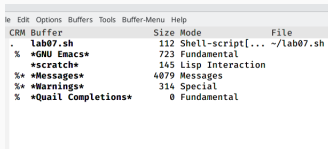
```

Figure 15: Переключение на другой буфер

- 7.3. Закроем это окно (C-x 0), (рис. 16)

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

- 7.4. Теперь вновь переключимся между буферами, но уже без вывода их списка на экран (C-x b), (рис. 17)



The image shows a screenshot of the Emacs buffer menu, which is a window titled "le" with a menu bar containing "Edit", "Options", "Buffers", "Tools", "Buffer-Menu", and "Help". The menu displays a list of buffers with columns for "CRM", "Buffer", "Size", "Mode", and "File". The buffers listed are: ". lab07.sh" (112, Shell-script, ~/lab07.sh), "% *GNU Emacs*" (723, Fundamental), "% *scratch*" (145, Lisp Interaction), "%* *Messages*" (4079, Messages), "%* *Warnings*" (314, Special), and "% *Quail Completions*" (0, Fundamental).

CRM	Buffer	Size	Mode	File
.	lab07.sh	112	Shell-script	~/lab07.sh
%	*GNU Emacs*	723	Fundamental	
%	*scratch*	145	Lisp Interaction	
%*	*Messages*	4079	Messages	
%*	*Warnings*	314	Special	
%	*Quail Completions*	0	Fundamental	

Figure 17: Переключение между буферами

8. Управление окнами

- 8.1. Поделим фрейм на 4 части: разделим 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)



Figure 18: 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. 19)

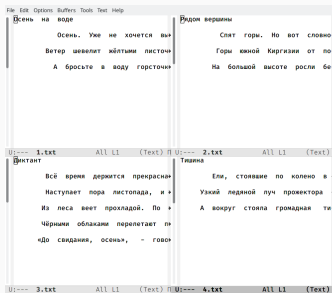


Figure 19: Новые файлы

9. Режим поиска

- 9.1. Перключимся в режим поиска (C-s) и найдём несколько слов, присутствующих в тексте. (рис. 20)

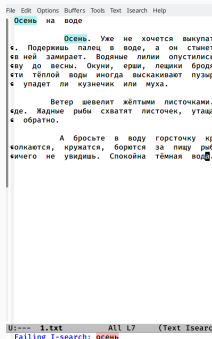


Figure 20: Поиск слов

- 9.2. Переключимся между результатами поиска, нажимая C-s. (рис. 21)

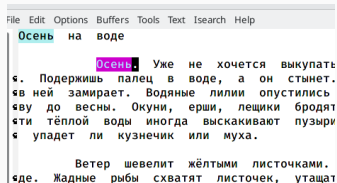


Figure 21: Переключение между результатами

- 9.3. Выходим из режима поиска, нажав C-g.
- 9.4. Переходим в режим поиска и замены (M-%), вводим текст, который следует найти и заменить, нажимаем Enter, затем вводим текст для замены. После того как будут подсвечены результаты поиска, нажимаем “!” для подтверждения замены. (рис. 22), (рис. 23)

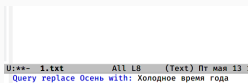


Figure 22: Замена слова

Выводы

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