

# Лабораторная работа №11

Операционные системы

---

Луангсуваннавонг Сайпхачан

23 апреля 2025

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Луангсуваннавонг Сайпхачан
- Студент из группы НКАбд-01-24
- Российский университет дружбы народов
- <https://sayprachanh-lsvnv.github.io>

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

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

## Выполнение лабораторной работы

---

Я запускаю emacs через терминал (рис. 1)

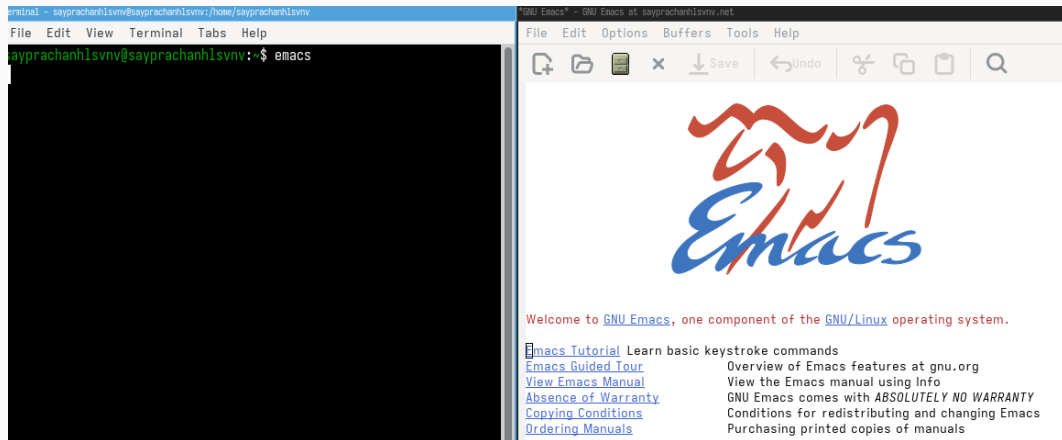


Рис. 1: Окно Emacs

С помощью Ctrl-x Ctrl-f (C-x C-f) создаю файл lab07.sh (рис. 2)

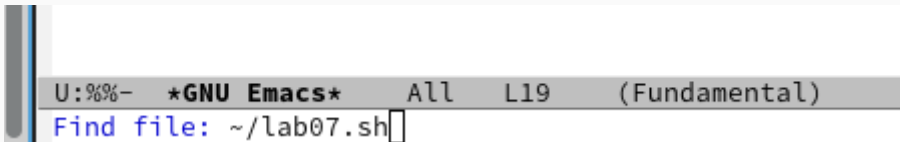


Рис. 2: Создание нового файла



Добавляю текстовый код в файл lab07.sh (рис. 3)

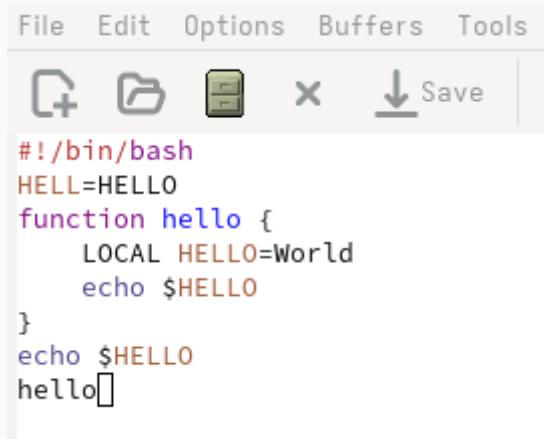
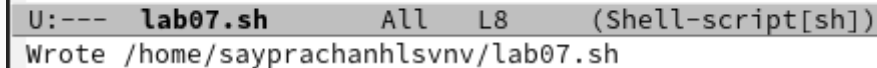


Рис. 3: Добавление текстового кода

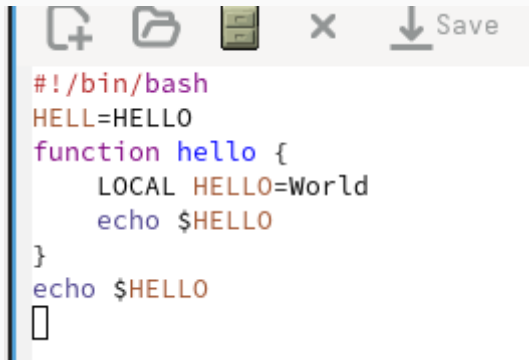
Сохраняю файл комбинацией Ctrl-x Ctrl-s (C-x C-s) (рис. 4)



```
U:--- lab07.sh All L8 (Shell-script[sh])  
Wrote /home/sayprachanhlsnv/lab07.sh
```

Рис. 4: Сохранение файла

Используя Ctrl-k (C-k), вырезаю целую строку (рис. 5)



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

Рис. 5: Вырезание целой линии

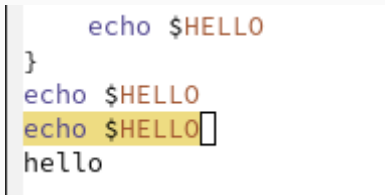
Вырезаю другую строку, затем вставляю её в конец файла комбинацией Ctrl-y (C-y) (рис. 6)



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

Рис. 6: Вставка строки

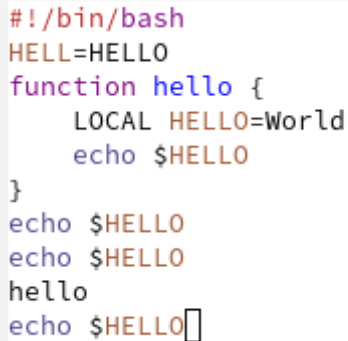
Выделяю область текста с помощью Ctrl-space (C-space), затем копирую её комбинацией Alt-w или End-w (M-w) (рис. 7)



The screenshot shows a text editor with the following content: `echo $HELLO`, `}`, `echo $HELLO`, `echo $HELLO`, and `hello`. The line `echo $HELLO` is highlighted in yellow, indicating it has been selected. A cursor is visible at the end of this line.

Рис. 7: Выделение текста и его копирование

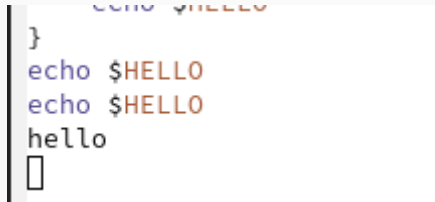
Затем вставляю скопированный текст в конец файла с помощью Ctrl-y (C-y) (рис. 8)



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

Рис. 8: Вставка строки

Снова выделяю текст и вырезаю его комбинацией Ctrl-w (C-w) (рис. 9)



```
}  
echo $HELLO  
echo $HELLO  
hello  
|
```

Рис. 9: Вырезание целой линии

Для отмены последнего действия использую Ctrl-/ (C-/) (рис. 10 и рис. 11)



Рис. 10: Отмена последнего действия



```
}  
echo $HELLO  
echo $HELLO  
hello  
echo $HELLO
```

Рис. 11: Текстовый код

Перемещаю курсор в начало строки с помощью Ctrl-a (C-a) (рис. 12)

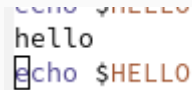
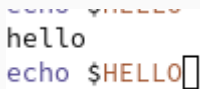
A screenshot of an Emacs buffer with a light gray background. It shows three lines of text: the first line is partially visible as 'echo \$HELLO' in blue and orange; the second line is 'hello' in black; and the third line is 'echo \$HELLO' in blue and orange. A black cursor box is positioned at the very beginning of the third line, before the 'e' of 'echo'.

Рис. 12: Курсор в начале строки

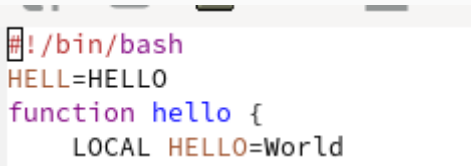
Перемещаю курсор в конец строки комбинацией Ctrl-e (C-e) (рис. 13)



```
hello  
echo $HELLO
```

Рис. 13: Курсор в конце строки

Используя Alt-< или Esc-< (M-<), перемещаю курсор в начало буфера (рис. 14)

A screenshot of an Emacs editor window showing a shell script. The text is color-coded: the shebang line is red, the variable assignment is orange, and the function definition is blue. A black cursor box is positioned at the very beginning of the first line, before the hash symbol.

```
#!/bin/bash  
HELL=HELLO  
function hello {  
  LOCAL HELLO=World
```

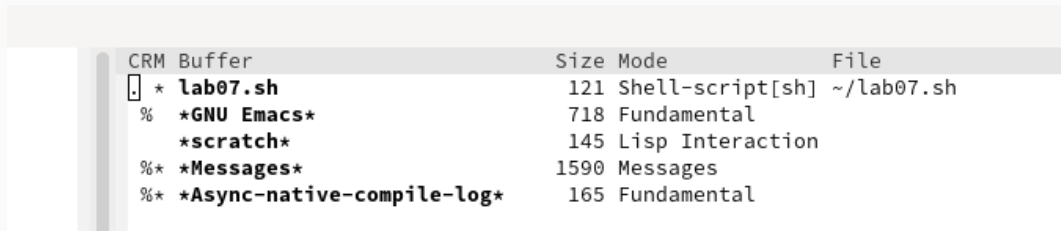
Рис. 14: Курсор в начале файла

С помощью Alt-> или Esc-> (M->) перемещаю курсор в конец буфера (рис. 15)

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

Рис. 15: Курсор в конце файла

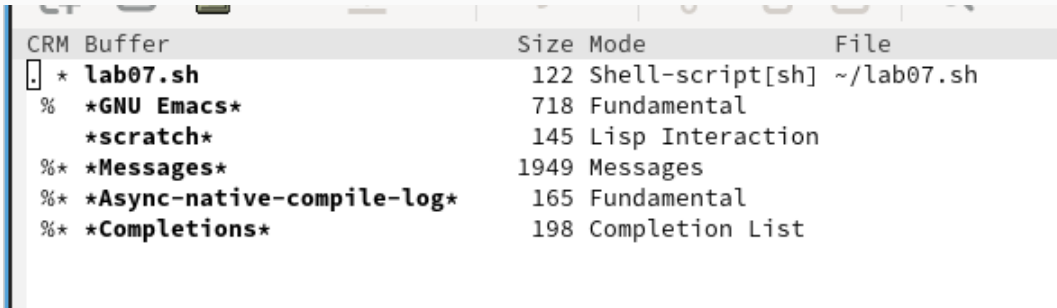
Комбинацией Ctrl-x Ctrl-b (C-x C-b) отображаю список активных буферов, затем переключаюсь между ними с помощью Ctrl-x и буквы o (Ctrl-x o) (рис. 16)

A screenshot of the Emacs buffer list, showing a table with columns for CRM, Buffer, Size, Mode, and File. The buffers listed are lab07.sh, GNU Emacs, scratch, Messages, and Async-native-compile-log. The first buffer, lab07.sh, is highlighted with a cursor.

CRM	Buffer	Size	Mode	File
[.]	* lab07.sh	121	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	718	Fundamental	
	*scratch*	145	Lisp Interaction	
%*	*Messages*	1590	Messages	
%*	*Async-native-compile-log*	165	Fundamental	

Рис. 16: Список активных буферов

Закрываю список активных буферов комбинацией Ctrl-x с цифрой 0 (Ctrl-x 0) (рис. 17)

A screenshot of the Emacs buffer list window, which is a table with four columns: CRM, Buffer, Size, and Mode. The 'File' column is also present but empty. The first row is highlighted with a blue background. The 'Buffer' column contains various buffers, some with a '\*' prefix and some with a '%' prefix. The 'Size' column shows the size of each buffer in characters. The 'Mode' column shows the current mode of each buffer. The 'File' column shows the file name for each buffer.

CRM	Buffer	Size	Mode	File
1	* lab07.sh	122	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	718	Fundamental	
	*scratch*	145	Lisp Interaction	
%*	*Messages*	1949	Messages	
%*	*Async-native-compile-log*	165	Fundamental	
%*	*Completions*	198	Completion List	

Рис. 17: Заккрытие буфера

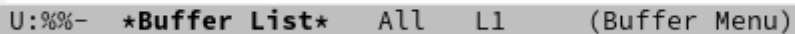
Переключаюсь между буферами без отображения списка с помощью Ctrl-x и буквы b (Ctrl-x b). Переключаюсь на буфер (Buffer List) (рис. 18 и рис. 19)

```
U:*~ lab07.sh All L10 (Shell-script[sh])
Click on a completion to select it.
In this buffer, type RET to select the completion near point.

6 possible completions:
*Async-native-compile-log*      *Buffer List*      *Completions*
*GNU Emacs*                    *Messages*         *scratch*
U:*~ *Completions* All L1 (Completion List)
Switch to buffer (default *Buffer List*):
```

Рис. 18: Переключение на буфер





The image shows a terminal window with a dark background. A horizontal menu bar is highlighted in light gray. The text in the menu bar is as follows:

U:%%-	<b>*Buffer List*</b>	All	L1	(Buffer Menu)
-------	----------------------	-----	----	---------------

Рис. 19: Буфер Buffer List

Возвращаюсь к lab07.sh, затем с помощью Ctrl-x с цифрой 3 (Ctrl-x 3) делю фрейм буфера по вертикали, а комбинацией Ctrl-x с цифрой 2 (Ctrl-x 2) — по горизонтали (рис. 20)

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

U:\*\*\* lab07.sh All L9 (Shell-script[sh])

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

```

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

U:\*\*\* lab07.sh All L9 (Shell-script

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

```

Открываю новый буфер в каждом из четырёх окон и ввожу текст (рис. 21)

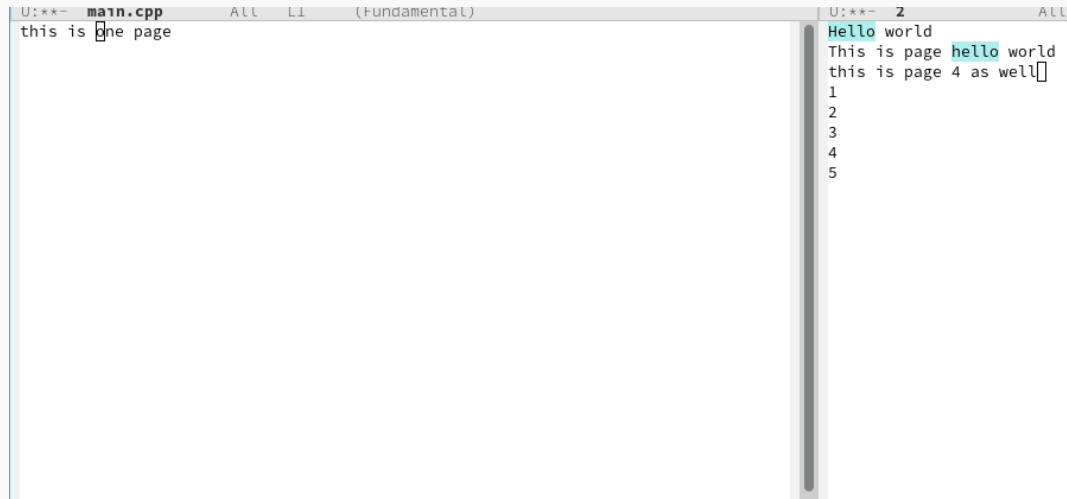


Активирую режим поиска комбинацией Ctrl-s (C-s), нахожу несколько слов в тексте (рис. 22)

this is one page

Hello world  
This is page hello world  
1  
2  
3  
4  
5

(Добавляю ещё текст для дополнительного поиска) (рис. 23)



```
U:**- main.cpp      ALL  L1  (Fundamental)
this is one page

U:**- 2             ALL
Hello world
This is page hello world
this is page 4 as well
1
2
3
4
5
```

Переключаюсь между результатами поиска с помощью Ctrl-S (C-S) (рис. 24)



Выхожу из режима поиска комбинацией Ctrl-g (C-g) (рис. 25)



Рис. 25: Выход из режима поиска

Активирую режим поиска и замены комбинацией Alt-% или Esc-% (M-%), ввожу текст для замены, затем текст замены и нажимаю Enter для применения (рис. 26)

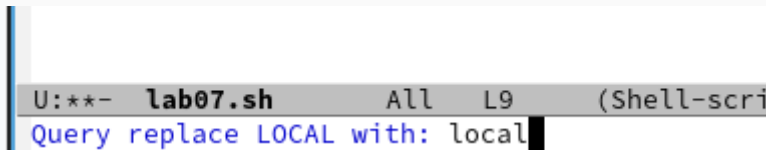


Рис. 26: Режим поиска и замены



Проверяю результат — текст был заменён ('Local' -> 'local') (рис. 27)

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

Используя Alt-s или Esc-s с буквой o (M-s o), пробуем другой режим поиска: ввожу искомый текст, и результаты отображаются в отдельном окне (не в основном буфере) (рис. 28)

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

```
8 matches for "hello" in buffer: lab07.sh_
2:HELL=HELLO
3:function hello {
4:    local HELLO=World
5:    echo $HELLO
7:echo $HELLO
8:echo $HELLO
9:hello
10:echo $HELLO
```

Рис. 28: Другой режим поиска

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

Спасибо за внимание

---