

# Лабораторная работа 11

Текстовый редактор emacs

---

Власов Артем Сергеевич

26 апреля 2025

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

## Информация

---

::::::::: {.columns align=center} ::: {.column width="70%"}

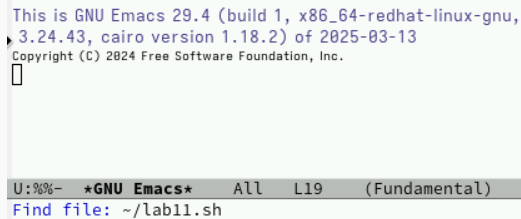
- Власов Артем Сергеевич
- Группа НПИбд-01-24
- Студент
- Российский университет дружбы народов
- 1132246841@pfur.ru

Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором emacs. # Задание

Выполнить последовательность действий по заданному сценарию.

## Выполнение лабораторной работы 11.

---




```
This is GNU Emacs 29.4 (build 1, x86_64-redhat-linux-gnu,  
3.24.43, cairo version 1.18.2) of 2025-03-13  
Copyright (C) 2024 Free Software Foundation, Inc.  
█  
  
U:%%- *GNU Emacs* All L19 (Fundamental)  
Find file: ~/lab11.sh
```

Рис. 1: Создание каталога и открытие файла с помощью emacs



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

Рис. 2: Вставка текста в файл



A terminal window with a light green background. At the top, a mouse cursor is positioned over the shell prompt `#!/bin/bash`. Below the prompt, a function `hello` is defined with the following code:  
`function hello {`  
    `LOCAL HELLO=World`  
    `echo $HELLO`  
`}`  
Below the function definition, the command `echo $HELLO` is entered. Then, the command `hello` is entered. Finally, the command `HELL=Hello` is entered, with a black cursor block at the end of the line.

```
#!/bin/bash

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

Рис. 3: вырезать и вставить строку



```
#!/bin/bash

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

```
#!/bin/bash

function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
█
}
echo $HELLO
hello
HELL=Hello
```

```
#!/bin/bash

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

# Перемещение курсора в начало и конец строки и буфера

```
#!/bin/bash
```

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

```
echo $HELLO
```

```
hello
```

```
HELL=Hello
```

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

```
echo $HELLO
```

```
hello
```

```
HELL=Hello
```

```
U:*** lab11.sh All L9 (Shell-script[sh])
```

```
Beginning of buffer
```

```
completing-read-default: Buffer is read-only: #<buffer *GNU Emacs*> [3 times]  
(New file)
```

```
Setting up indent for shell type sh
```

```
Indentation variables are now local.
```

```
Indentation setup for shell type sh
```

```
Mark set
```

```
C-x <mouse-1> is undefined
```

```
Auto-saving...done
```

```
funcall-interactively: Text is read-only
```

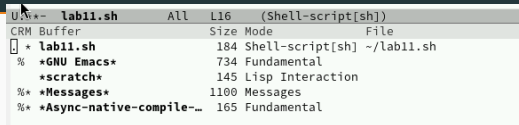
```
Undo
```

```
xref-go-back: At start of xref history
```

```
read-file-name-default: Command attempted to use minibuffer while in minibuffer
```

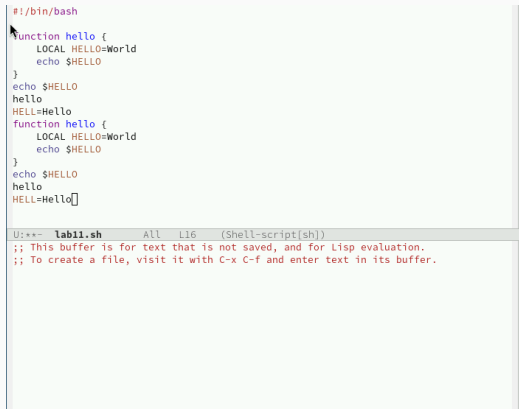
Рис. 7: Перемещение курсора в начало и конец строки и буфера

# Список активных буферов, переключение между буферами и закрытие окна буфера

A screenshot of the Emacs buffer list window. The title bar shows 'U:\*\*\*- lab11.sh All L16 (Shell-script[sh])'. The table lists several buffers: 'CRM Buffer' with size 184 and mode 'Shell-script[sh]', '~/' with size 734 and mode 'Fundamental', '\*lab11.sh\*' with size 145 and mode 'Lisp Interaction', '\*GNU Emacs\*' with size 1100 and mode 'Messages', '\*scratch\*' with size 165 and mode 'Fundamental', and '\*Messages\*' with size 165 and mode 'Fundamental'.

CRM Buffer	Size	Mode	File
*lab11.sh*	184	Shell-script[sh]	~/lab11.sh
% *GNU Emacs*	734	Fundamental	
% *scratch*	145	Lisp Interaction	
% *Messages*	1100	Messages	
% *Async-native-compile-...	165	Fundamental	

Рис. 8: Список активных буферов

A screenshot of the Emacs editor showing a shell script. The title bar shows 'U:\*\*\*- lab11.sh All L16 (Shell-script[sh])'. The script content is as follows:

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

;; This buffer is for text that is not saved, and for Lisp evaluation.  
;; To create a file, visit it with C-x C-f and enter text in its buffer.

# Переключение буферов комбинацией клавиш

```
#!/bin/bash
```

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

```
echo $HELLO
```

```
hello
```

```
HELL=Hello
```

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

```
echo $HELLO
```

```
hello
```

```
HELL=Hello
```

```
U:***- lab11.sh All L16 (Shell-script[sh])
```

Click on a completion to select it.

In this buffer, type RET to select the completion near point.

4 possible completions:

```
*Async-native-compile-log*      *Buffer List*
```

```
*Messages*                      *scratch*
```

```
U:%*- *Completions* All L1 (Completion List)
```

Switch to buffer (default \*scratch\*):

Рис. 11: Переключение буферов комбинацией клавиш

## Разделение экрана на 4 части

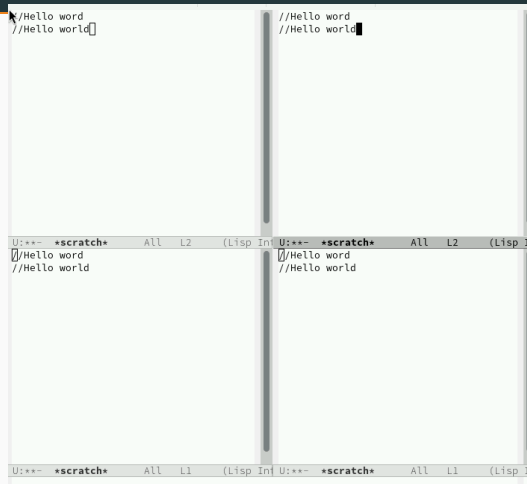


Рис. 12: Разделение экрана

## Поиск по файлу и переключение между результатами поиска

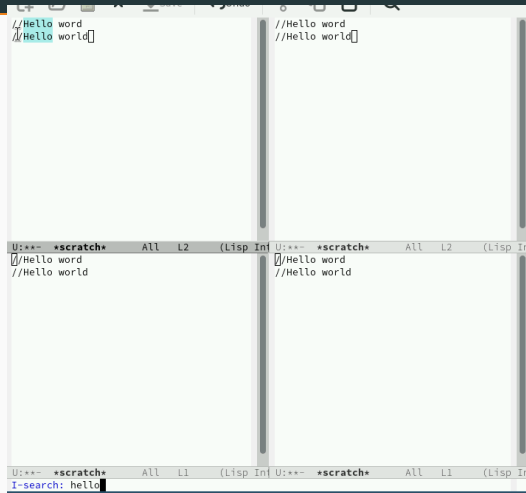


Рис. 13: Поиск по файлу







Рис. 15: Замена

Поиск с помощью другого режима(показывает строки вхождения)



The screenshot shows an Emacs search window titled "U:%%- \*0ccur\*" with search options "All L1 (Occur)". The search results are displayed in a list box, showing 2 matches for "Hello" in buffer: \*scratch\*. The matches are:

- 1://Hello word
- 2://Hello world

The word "Hello" in both lines is highlighted in yellow. The search bar at the top of the window contains the text "2 matches for 'Hello' in buffer: \*scratch\*" and a cursor is positioned at the end of the text.

Рис. 16: Поиск

Мы освоили основные возможности текстового редактора emacs, научились выполнять действия, используя горячие клавиши.