

Лабортная работа №11

Операционные системы

Панявкина И.В.

25 апреля 2025

Российский университет дружбы народов, Москва, Россия

Цель данной лабораторной работы - 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

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. По популярности он соперничает с редактором vi и его клонами. В зависимости от ситуации, Emacs может быть:

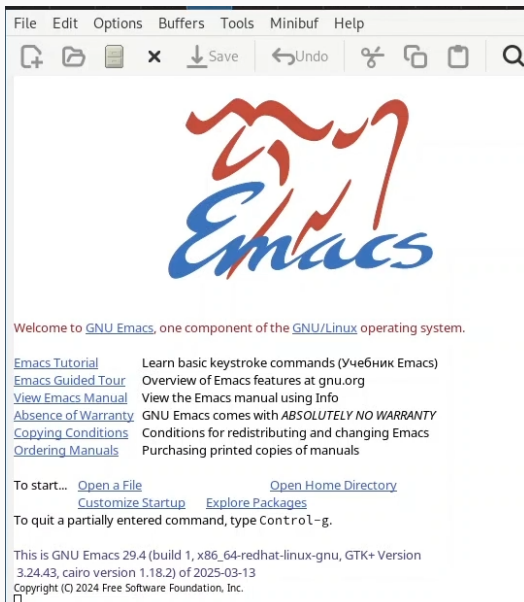
- текстовым редактором;
- программой для чтения почты и новостей Usenet;
- интегрированной средой разработки (IDE);
- операционной системой;

Всё это разнообразие достигается благодаря архитектуре Emacs, которая позволяет расширять возможности редактора при помощи языка Emacs Lisp. На языке C написаны лишь самые базовые и низкоуровневые части Emacs, включая полнофункциональный интерпретатор языка Lisp. Таким образом, Emacs имеет встроенный язык программирования, который может использоваться для настройки, расширения и изменения поведения редактора. В действительности, большая часть того редактора, с которым пользователи Emacs работают в наши дни, написана на языке Lisp.

Первая версия редактора Emacs была написана в 70-х годах 20-го столетия Richard Stallman (Ричардом Столманом) как набор макросов для редактора TECO . В дальнейшем, уже будучи основателем Фонда Свободного программного обеспечения Free Software Foundation и проекта GNU, Stallman разработал GNU Emacs в развитие оригинального Emacs и до сих пор сопровождает эту программу. Emacs является одним из старейших редакторов. Он использовался тысячами программистов на протяжении последних 20 с лишним лет, для него создано много дополнительных пакетов расширений. Эти дополнения позволяют делать с помощью Emacs такие вещи, которые Stallman , вероятно, даже не считал возможными в начале своей работы над редактором.

Открываю Emacs через терминал (рис. 1).

Выполнение лабораторной работы



Создаю файл ab07.sh с помощью комбинации Ctrl-x Ctrl-f (рис. 2).

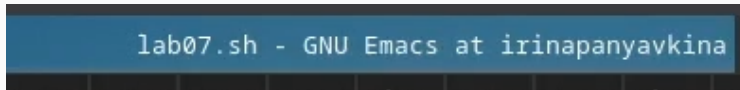
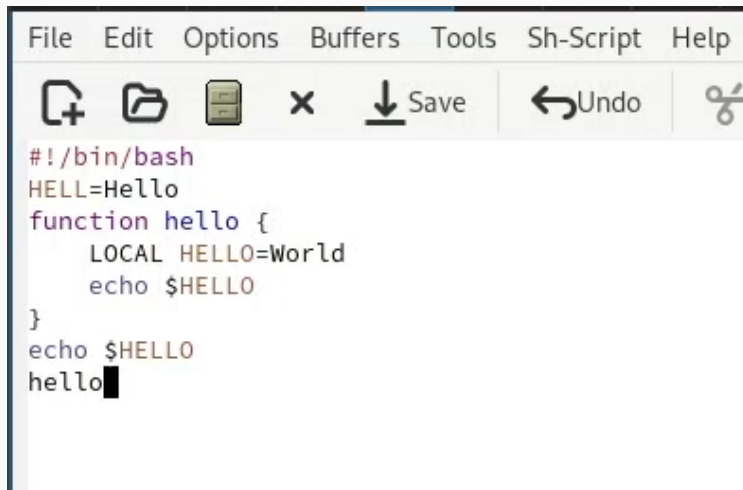


Рис. 2: Создание файла

Выполнение лабораторной работы

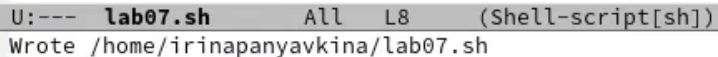
Прописываю в файле текст программы (рис. 3).

A screenshot of a text editor window with a menu bar and a toolbar. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The toolbar contains icons for opening a file, saving a file, closing a file, a close button (X), a save button (floppy disk), a 'Save' button with a download icon, an 'Undo' button with a left arrow, and a cut button (scissors). The text area contains a shell script with the following content:

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

The script defines a function 'hello' that sets a local variable 'HELLO' to 'World' and echoes it. It then echoes the global variable 'HELL' (which is 'Hello') and calls the 'hello' function. The cursor is at the end of the 'hello' command line.

Сохраняю файл с помощью комбинации C-x C-s (рис. 4).



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

Рис. 4: Сохранение изменений в файле

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

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

Вставляю эту строку в конец файла (C-y) (рис. 6).

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

Выполнение лабораторной работы

Выделяю область текста (C-space), копирую область в буфер обмена (M-w), вырезаю эту область с помощью C-w (рис. 7).

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



Рис. 7: Вырезанная область

Вставляю область в конец файла (C-y)(рис. 8).

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

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

Отменяю последнее действие C-/ (рис. 9).

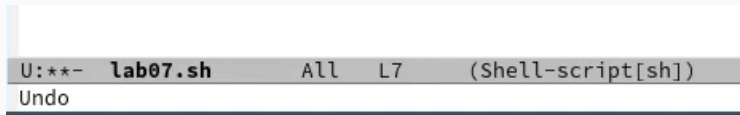
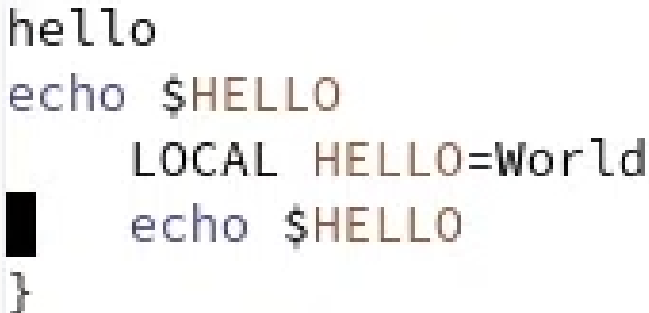


Рис. 9: Отмена последнего действия

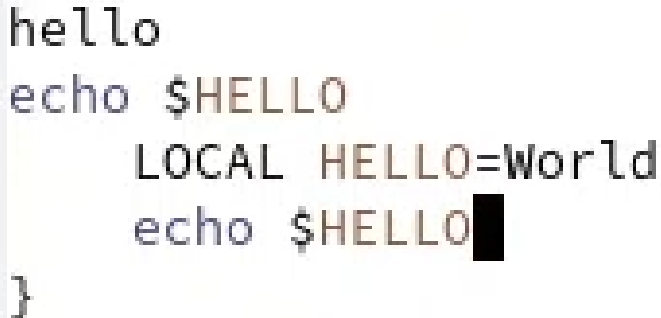
Перевожу курсор в начало строки C-a (рис. 10).



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

Рис. 10: Курсор в начале строки

Перемещаю курсор в конец строки C-e (рис. 11).



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

Рис. 11: Курсор в конце строки

Выполнение лабораторной работы

Перемещаю курсор в начало файла M-< (рис. 12).

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

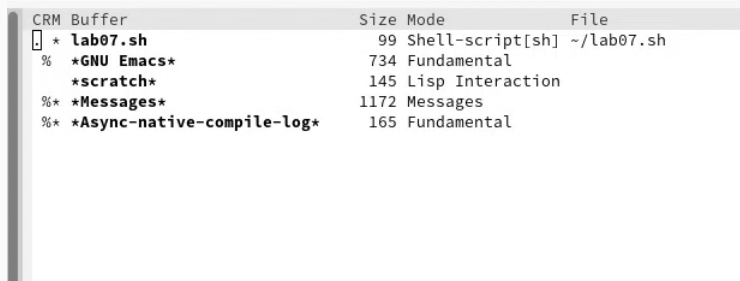
hello
echo $HELLO
    LOCAL HELLO=World
    echo $HELLO
}
```

Перемещаю курсор в конец файлаМ->(рис. 13).

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

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

Открываю список активных буферов в другом окне C-x C-b (рис. 14).

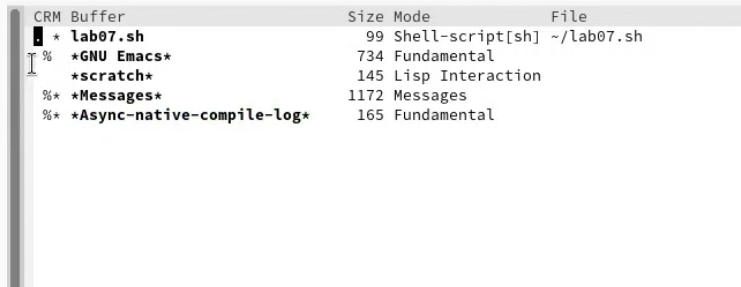


The screenshot shows the Emacs buffer list window, which is a table with five columns: CRM, Buffer, Size, Mode, and File. The table lists several buffers, with the first one, ***lab07.sh**, selected. The other buffers listed are ***GNU Emacs***, ***scratch***, ***Messages***, and ***Async-native-compile-log***.

CRM	Buffer	Size	Mode	File
1	*lab07.sh	99	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	734	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	1172	Messages	
%*	*Async-native-compile-log*	165	Fundamental	

Рис. 14: Список активных буферов

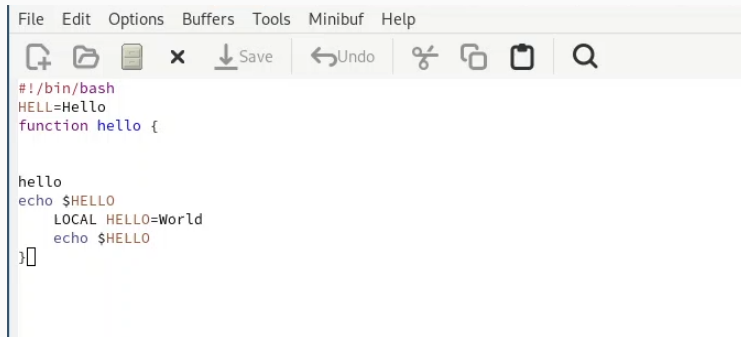
Переключаюсь на другой буфер C-x o (рис. 15).



CRM	Buffer	Size	Mode	File
	* lab07.sh	99	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	734	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	1172	Messages	
%*	*Async-native-compile-log*	165	Fundamental	

Рис. 15: Другое окно буфера

Закрываю окно другого буфера C-x 0 (рис. 16).



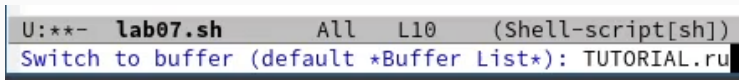
```
File Edit Options Buffers Tools Minibuf Help
[Icons: Open, Save, Undo, Redo, Search]

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

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

Рис. 16: Заккрытие буфера

Открываю другой буфер без вывода их списка на экран с помощью C-x b (рис. 17).


A screenshot of a terminal window. The top line shows the prompt 'U:**-' followed by 'lab07.sh', 'All', 'L10', and '(Shell-script[sh])'. The second line shows the command 'Switch to buffer (default *Buffer List*): TUTORIAL.ru' with a black cursor at the end. The terminal has a light gray background and a dark blue border.

```
U:**- lab07.sh All L10 (Shell-script[sh])  
Switch to buffer (default *Buffer List*): TUTORIAL.ru
```

Рис. 17: Открытие другого буфера

Делю фрейм на 4 части: сначала на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2) (рис. 18).

Выполнение лабораторной работы



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

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

-:***- lab07.sh All L11 (Shell-script(bash))
#!/bin/bash
HELL=Hello
function hello {

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

-:***- lab07.sh All L11 (Shell-script(bash))
#!/bin/bash
HELL=Hello
function hello {

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

-:***- lab07.sh All L11 (Shell-script(bash))
#!/bin/bash
HELL=Hello
function hello {

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

-:***- lab07.sh All L11 (Shell-script(bash))
```

Рис. 18: Четыре окна

В каждом из четырех созданных окон с помощью C-x b открыла разные буферы и (рис. 19).

Выполнение лабораторной работы

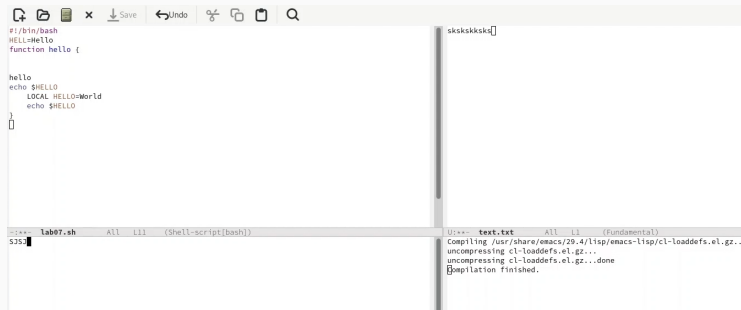


Рис. 19: Открытие буферов в четырех окнах

Перехожу в режим поиска с помощью C-s, ищу слова в тексте, они подсвечиваются (рис. 20).

SJSJDKSK

С помощью той же комбинации C-s я могу перемещаться по результатам поиска (рис. 21).

SJSJDKSK

U:**- TUTORIAL.ru All L1 (Fundamental Isearch)

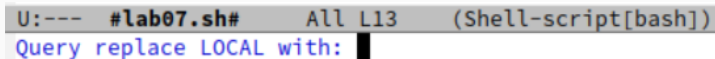
Failing I-search: si

С помощью C-g выхожу из режима поиска, снимается выделение (рис. 22).



Рис. 22: Выход из режима поиска

Перехожу в режим поиска и замены с помощью M-%, ввожу какое слово хочу заменить, затем ввожу на какое хочу заменить (рис. 23).

A screenshot of a terminal window. The top bar shows the user 'U:---', the file '#lab07.sh', the line 'All L13', and the shell '(Shell-script[bash])'. The main area shows the command 'Query replace LOCAL with:' followed by a black cursor block.

```
U:--- #lab07.sh# All L13 (Shell-script[bash])
Query replace LOCAL with: █
```

Рис. 23: Замена слова

Видим, что слова были заменены успешно (рис. 24).

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

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

С помощью M+s перехожу в другой режим поиска. Он отличается от предыдущего тем, что выводит результат в отдельном окне от окна буфера (рис. 25).

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

```
U:**~ #lab07.sh# All L13 (Shell-script[bash])
```

```
2 matches for "local" in buffer: #lab07.sh#
```

```
7:local HELLO= World
```

```
8:local HELLO= World
```

Выводы

В ходе данной лабораторной работы я познакомилась с операционной системой Linux, получила практические навыки работы с редактором Emacs.

Список литературы

1. Лабораторная работа №11 [Электронный ресурс] URL:
<https://esystem.rudn.ru/mod/resource/view.php?id=1224389>