

Презентация Лабораторной работы №11

Операционные системы

Петрова А.А.

20 апреля 2024

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;

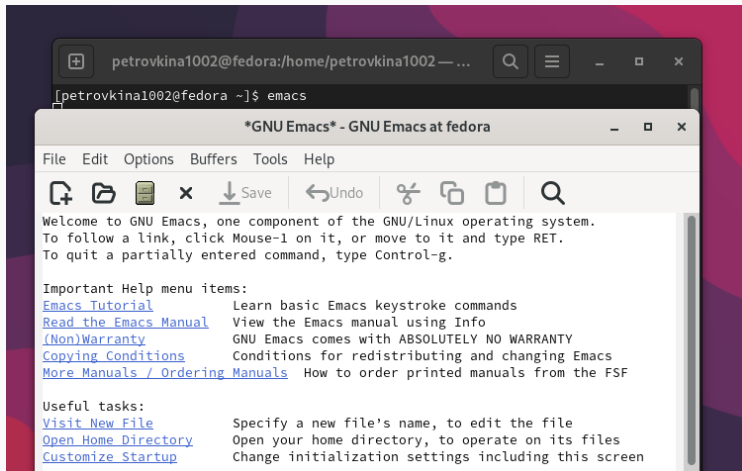
Всё это разнообразие достигается благодаря архитектуре Emacs, которая позволяет расширять возможности редактора при помощи языка Emacs Lisp. На языке C написаны лишь самые базовые и низкоуровневые части Emacs, включая полнофункциональный интерпретатор языка Lisp. Таким образом, Emacs имеет встроенный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

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.

Выполнение лабораторной работы

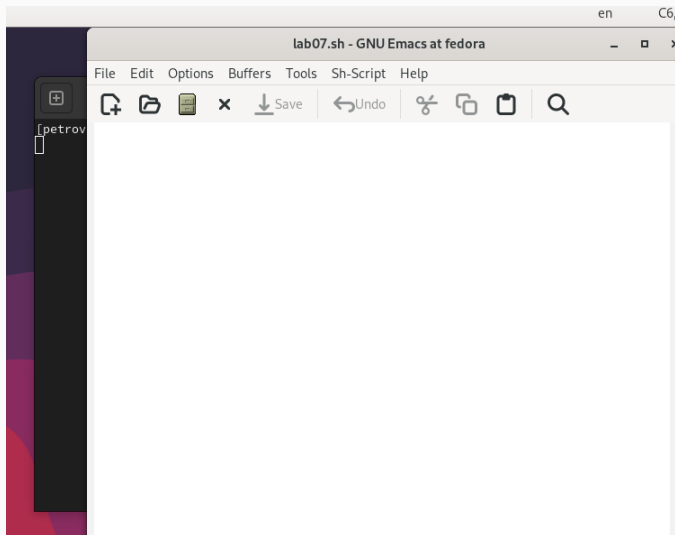
Сначала я установила Emacs.

Открываю Emacs через терминал (рис.1).



Выполнение лабораторной работы

Создаю файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (рис.2).



Прописываю в файле текст программы (рис.3).

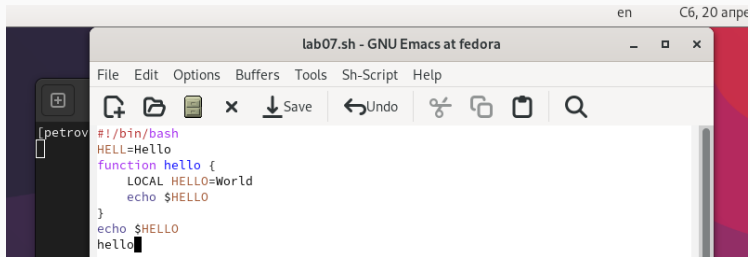
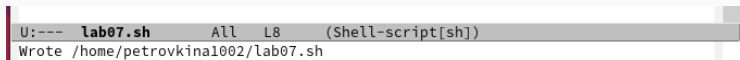


Рис. 3: Редактирование файла

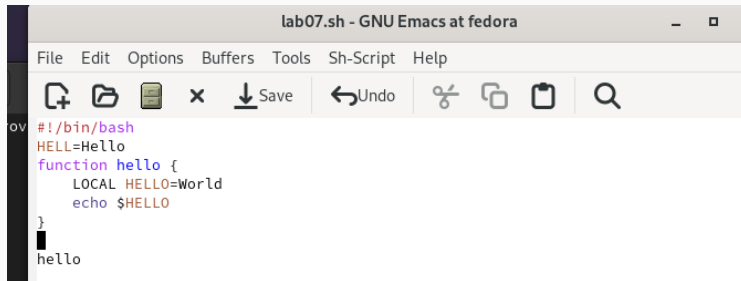
Сохраняю файл с помощью комбинации C-x C-s (рис.4).

A screenshot of a terminal window. The prompt is 'U:---'. The current file is 'lab07.sh'. The user is in 'All' mode and 'L8' line. The terminal shows the command 'C-x C-s' being entered, which results in the message 'Wrote /home/petrovkina1002/lab07.sh'.

```
U:--- lab07.sh All L8 (Shell-script[sh])  
Wrote /home/petrovkina1002/lab07.sh
```

Рис. 4: Сохранение изменений в файле

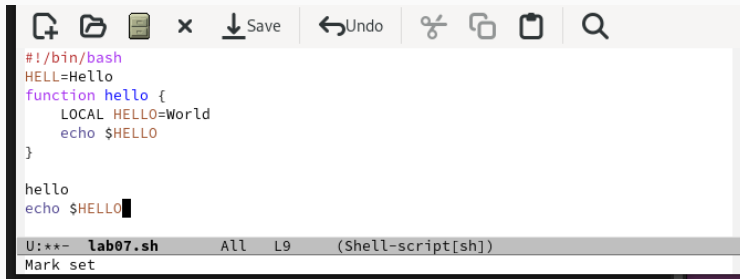
Вырезаю одной командой целую строку (C-k) (рис.5).



```
lab07.sh - GNU Emacs at fedora
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Close, Save All, Undo, Cut, Copy, Paste, Find]
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
```

Рис. 5: Вырезание строки

Вставляю эту строку в конец файла (С-у) (рис.6).



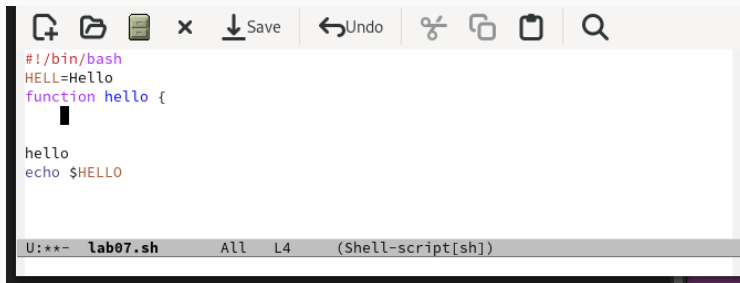
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}

hello
echo $HELLO
```

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

Рис. 6: Вставка строки в конце файла

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



The image shows a terminal window with a light gray toolbar at the top. The toolbar contains icons for opening a file, saving, undo, redo, cut, copy, paste, and search. Below the toolbar, the terminal displays a shell script with syntax highlighting: a shebang line, an assignment, a function definition, and two function calls. The status bar at the bottom shows the current directory, filename, and shell type.

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

U:*- lab07.sh All L4 (Shell-script[sh])

Рис. 7: Вырезанная область

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



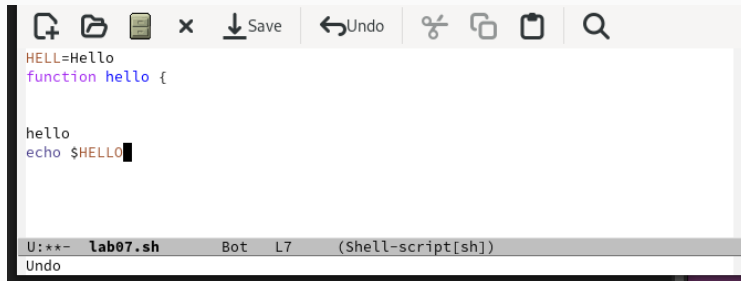
```
HELL=Hello
function hello {

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

U:*** lab07.sh Bot L10 (Shell-script[sh])
```

Рис. 8: Вставка в конец файла

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



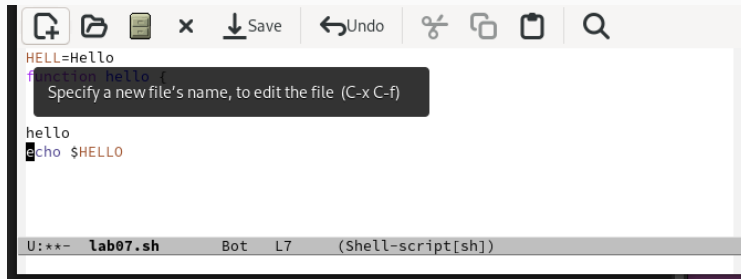
The screenshot shows a terminal window with a light gray title bar and a dark gray status bar. The title bar contains icons for file operations (new, open, save, close) and editing (undo, redo, copy, paste, search). The main area of the terminal displays a shell script with the following content:

```
HELL=Hello  
function hello {  
  
hello  
echo $HELLO
```

The cursor is positioned at the end of the last line. The status bar at the bottom shows the prompt 'U:*** lab07.sh Bot L7 (Shell-script[sh])' and the command 'Undo' being entered.

Рис. 9: Отмена последнего действия

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



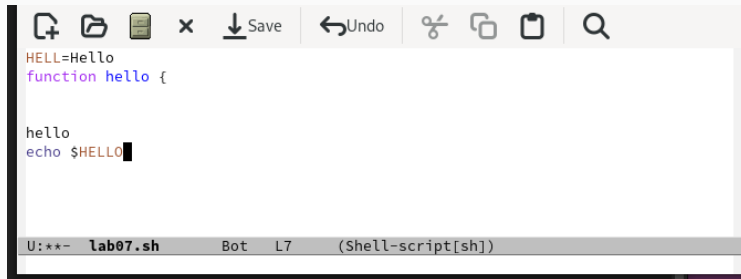
The screenshot shows a terminal window with a light gray title bar containing icons for file operations (add, open, save, close) and editing (undo, redo, copy, paste, search). The terminal content displays a shell script with the following lines: `HELL=Hello`, `function hello {`, `hello`, and `echo $HELLO`. A dark gray tooltip is visible over the `function hello {` line, containing the text "Specify a new file's name, to edit the file (C-x C-f)". The cursor is positioned at the start of the `echo $HELLO` line. The terminal status bar at the bottom shows the prompt `U:~$`, the filename `lab07.sh`, and the shell type `(Shell-script[sh])`.

```
HELL=Hello
function hello {
hello
echo $HELLO
```

U:~\$ lab07.sh Bot L7 (Shell-script[sh])

Рис. 10: Курсор в начале строки

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



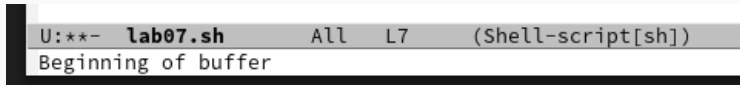
The image shows a terminal window with a light gray title bar and a dark gray border. The title bar contains several icons: a plus sign, a folder, a document, a close button (X), a save button (floppy disk), an undo button (curved arrow), a cut button (scissors), a copy button (two overlapping squares), a paste button (clipboard), and a search button (magnifying glass). The terminal content is as follows:

```
HELL=Hello  
function hello {  
  
hello  
echo $HELLO
```

The cursor is positioned at the end of the last line, after the text `echo $HELLO`. The bottom status bar of the terminal shows the prompt `U:~ - lab07.sh`, the user `Bot`, the host `L7`, and the shell `(Shell-script[sh])`.

Рис. 11: Курсор в конце строки

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

A screenshot of a terminal window. The top bar is grey and contains the text 'U:**- lab07.sh All L7 (Shell-script[sh])'. Below this, the text 'Beginning of buffer' is displayed on a white background. A black cursor is positioned at the start of the text.

```
U:**- lab07.sh All L7 (Shell-script[sh])  
Beginning of buffer
```

Рис. 12: Начало буфер

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

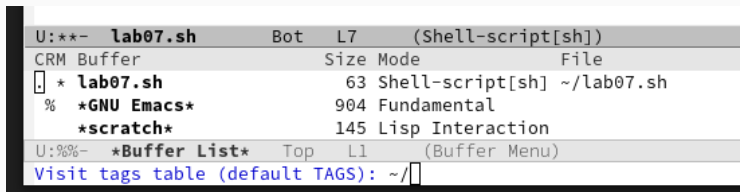
```
#!/bin/bash
HELL=Hello
function hello {

hello
echo $HELLO

U:**~ lab07.sh All L7 (Shell-script[sh])
Visit tags table (default TAGS): ~/
```

Рис. 13: Конец буфера

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 of active buffers. The window title is 'U:%%- *Buffer List* Top L1 (Buffer Menu)'. The table has columns for 'Buffer', 'Size', 'Mode', and 'File'. The buffers listed are: '*lab07.sh' (63 bytes, Shell-script[sh], ~/lab07.sh), '*GNU Emacs*' (904 bytes, Fundamental), and '*scratch*' (145 bytes, Lisp Interaction). The cursor is on the first buffer, '*lab07.sh'.

U:%%- *Buffer List* Top L1 (Buffer Menu)			
Buffer	Size	Mode	File
*lab07.sh	63	Shell-script[sh]	~/lab07.sh
GNU Emacs	904	Fundamental	
scratch	145	Lisp Interaction	

Рис. 14: Список активных буферов

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

```
To quit a partially entered command, type Control-g.
```

```
Important Help menu items:
```

```
Emacs Tutorial      Learn basic Emacs keystroke commands
Read the Emacs Manual View the Emacs manual using Info
\(Non\)Warranty      GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions Conditions for redistributing and changing E
More Manuals / Ordering Manuals How to order printed manuals from t
```

```
█
```

```
Useful tasks:
```

```
U:%%~ *GNU Emacs* 16% L11 (Fundamental)
```

```
visit-tags-table-buffer: File /home/petrovkina1002/TAGS does not exi
```

```
C-x <down> is undefined
```

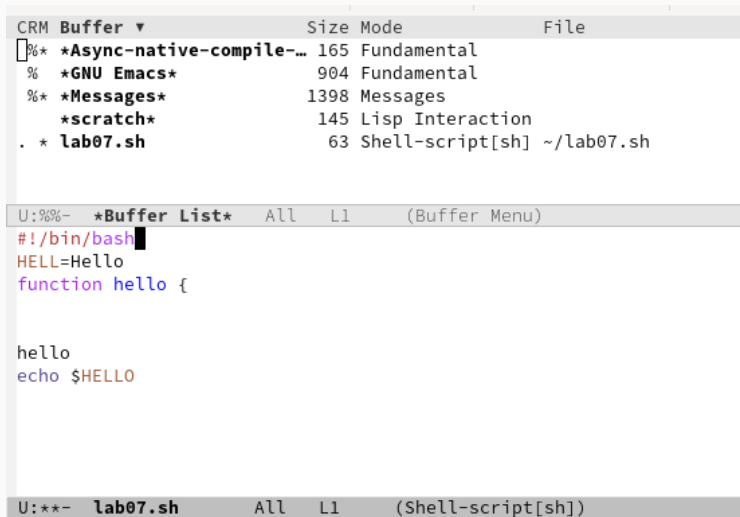
```
End of buffer [2 times]
```

```
█
```

```
U:%*- *Messages* Bot L38 (Messages)
```

Выполнение лабораторной работы

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



The image shows a screenshot of the Emacs editor interface. At the top, the 'Buffer List' window is open, displaying a table of buffers. Below this, the main Emacs window shows a shell prompt 'U:%%-' and some shell code. At the bottom, the status bar indicates the current buffer is 'lab07.sh'.

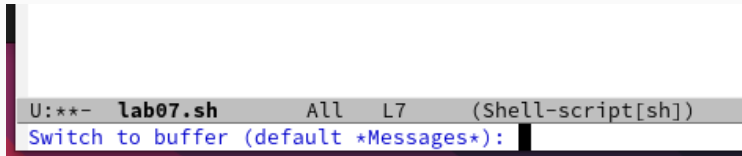
CRM Buffer ▾	Size	Mode	File
%* *Async-native-compile-...	165	Fundamental	
% *GNU Emacs*	904	Fundamental	
%* *Messages*	1398	Messages	
scratch	145	Lisp Interaction	
. * lab07.sh	63	Shell-script[sh]	~/lab07.sh


```
U:%%- *Buffer List* All L1 (Buffer Menu)
#!/bin/bash
HELL=Hello
function hello {

hello
echo $HELLO

U:*** lab07.sh All L1 (Shell-script[sh])
```

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

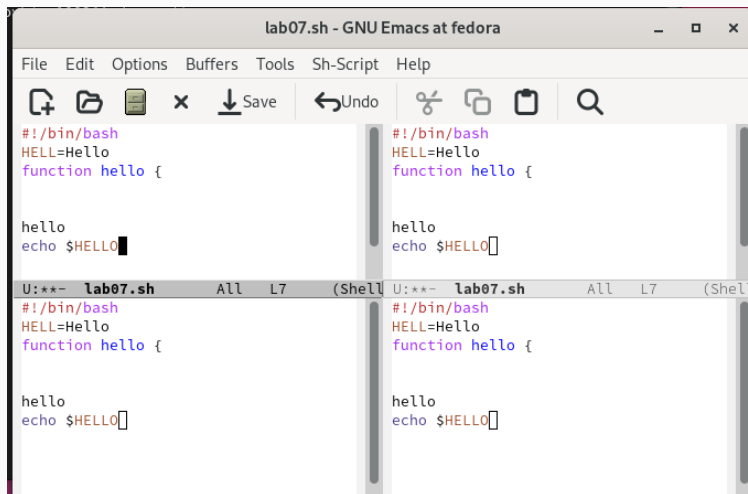
A screenshot of a terminal window. The top part of the window is a grey header bar with the text 'U:**- lab07.sh All L7 (Shell-script[sh])'. Below the header, the text 'Switch to buffer (default *Messages*):' is displayed in blue. A black cursor is positioned at the end of this line, ready for input.

```
U:**- lab07.sh All L7 (Shell-script[sh])  
Switch to buffer (default *Messages*):
```

Рис. 17: Открытие другого буфера

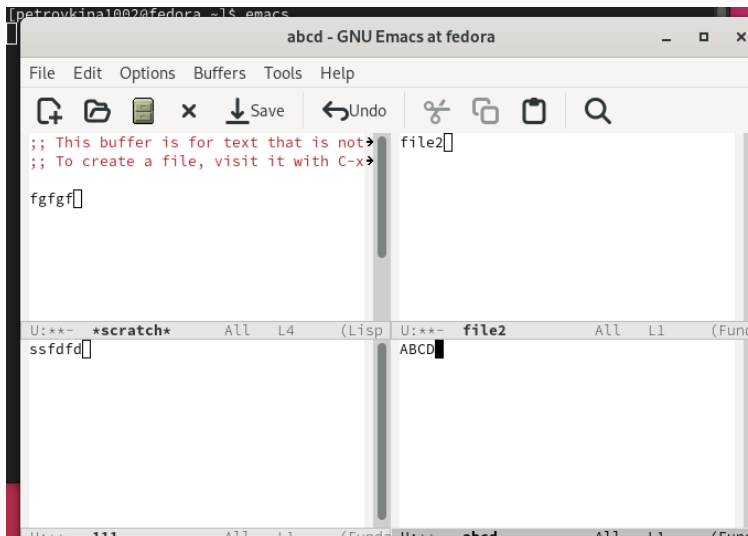
Выполнение лабораторной работы

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



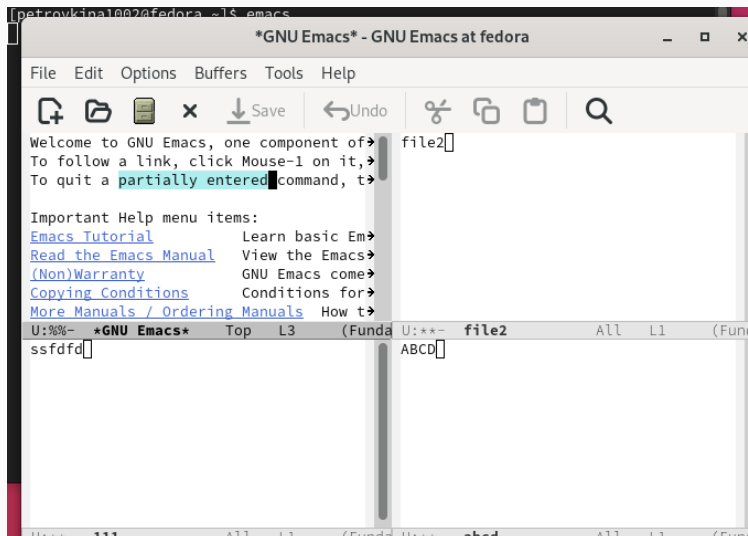
Выполнение лабораторной работы

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



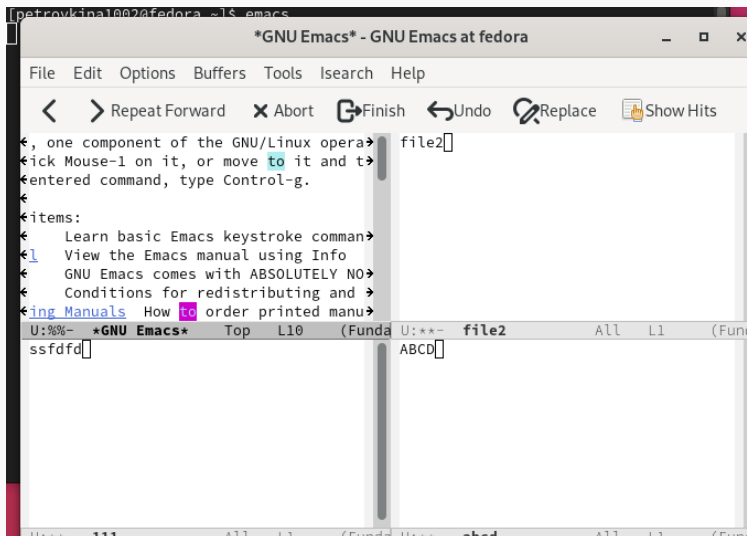
Выполнение лабораторной работы

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



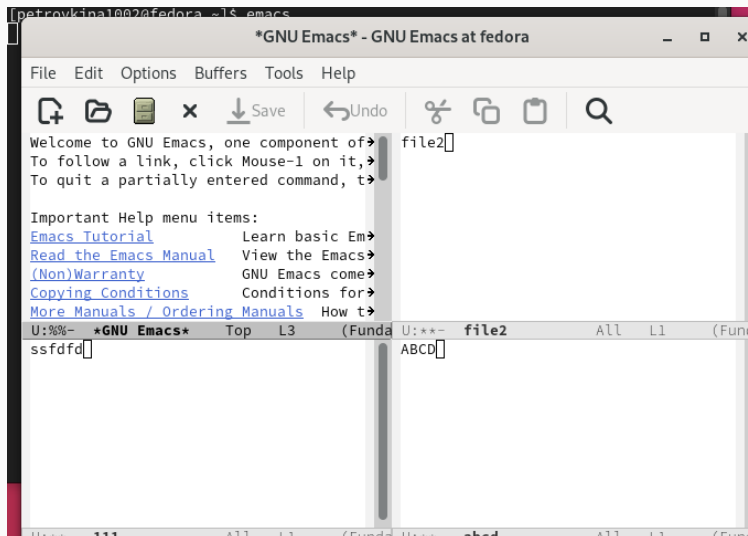
Выполнение лабораторной работы

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



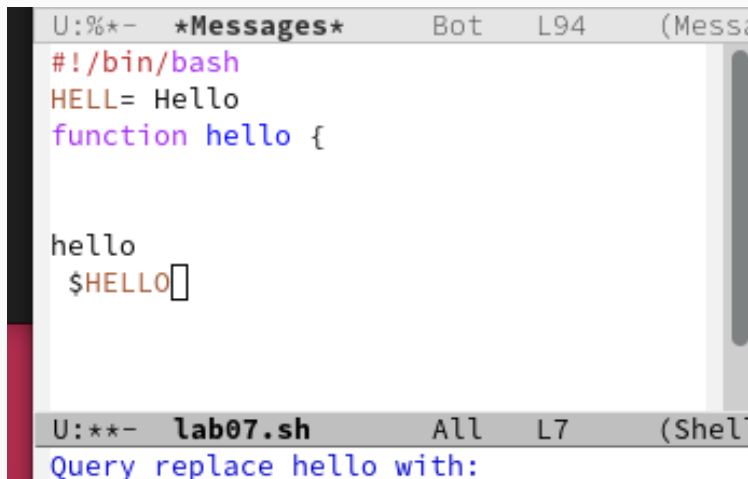
Выполнение лабораторной работы

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



Выполнение лабораторной работы

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



The screenshot shows a Vim editor window with a dark background. The top status bar displays 'U:%*- *Messages* Bot L94 (Mess...'. The main text area contains the following code:

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

hello
  $HELLLO
```

The bottom status bar displays 'U:**- lab07.sh All L7 (Shel...'. Below the status bar, the command 'Query replace hello with:' is visible.

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



```
U:%*- *Messages* Bot L95 (Mess
#!/bin/bash
HELL= Hello
function hello {

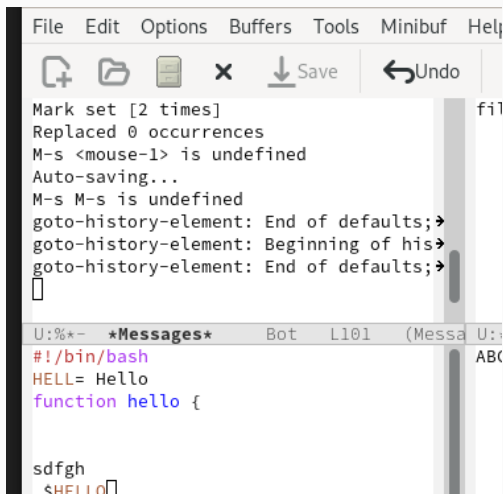
sdfgh
$HELLO

U:**- lab07.sh All L7 (Shel
```

Рис. 24: Слово заменено

Выполнение лабораторной работы

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).



The screenshot shows a Vim editor interface. The top menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Minibuf', and 'Help'. Below the menu is a toolbar with icons for file operations and 'Save' and 'Undo' buttons. The main window displays a message window with the following text:

```
Mark set [2 times]
Replaced 0 occurrences
M-s <mouse-1> is undefined
Auto-saving...
M-s M-s is undefined
goto-history-element: End of defaults;→
goto-history-element: Beginning of his→
goto-history-element: End of defaults;→

```

Below the message window is a terminal window with the following text:

```
U:%*- *Messages* Bot L101 (Messa U:s
#!/bin/bash
HELL= Hello
function hello {
sdfgh
$HELLO
```

В ходе данной работы я получила практические навыки работы с редактором Emacs.