

Презентация лабораторной работы №9

Презентация лабораторной работы №9

Цель

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

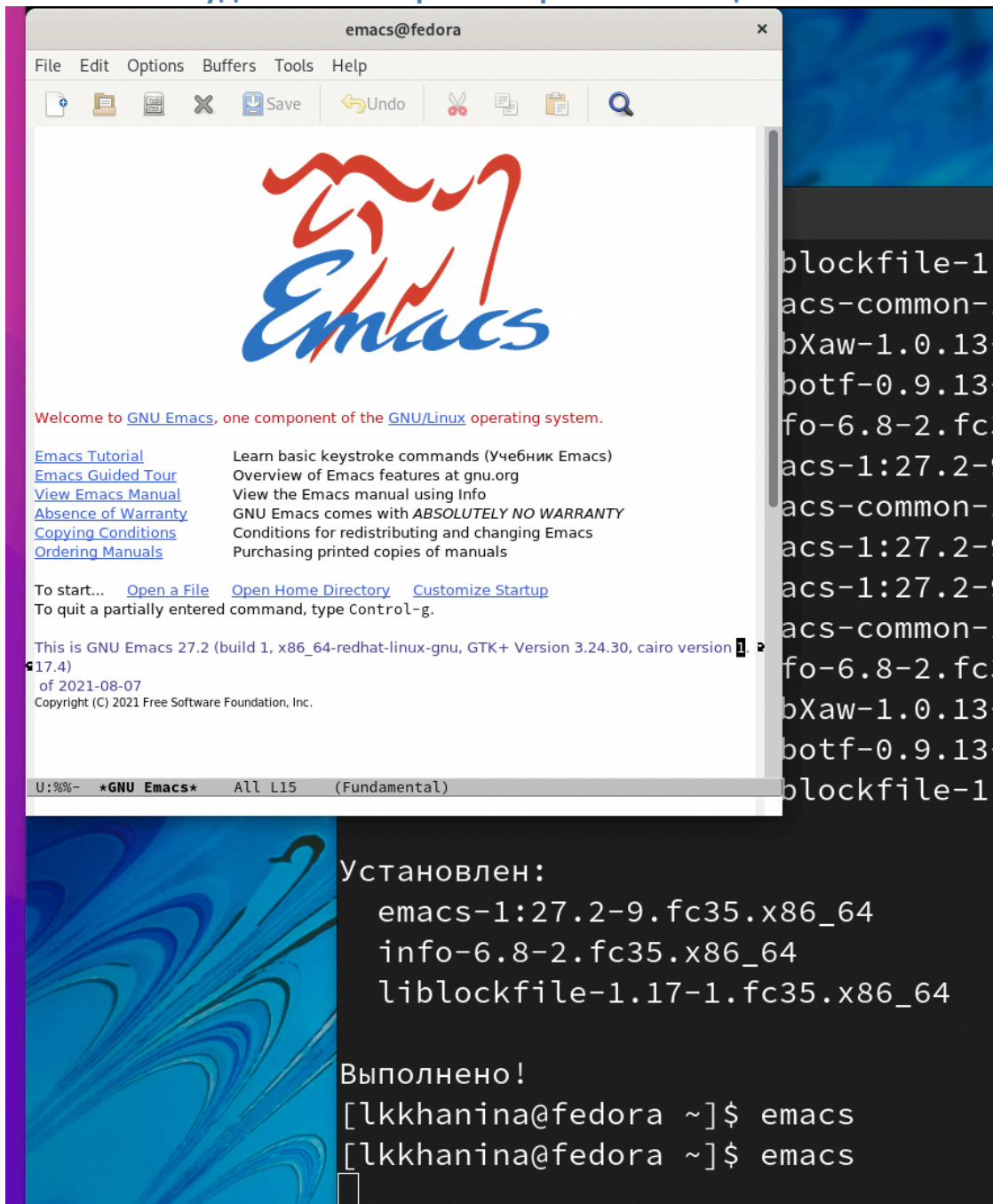
Задание

- Открыть emacs.
- Создать файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f. Набрать текст.
- Сохранить файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
- Вырезать одной командой целую строку (C-k).
- Вставить эту строку в конец файла (C-y).
- Выделить область текста (C-space).
- Скопировать область в буфер обмена (M-w).
- Вставить область в конец файла.
- Вновь выделить эту область и на этот раз вырезать её (C-w).
- Отменить последнее действие (C-/).
- Переместите курсор в начало строки (C-a).
- Переместите курсор в конец строки (C-e).
- Переместите курсор в начало буфера (M-<).
- Переместите курсор в конец буфера (M->).
- Вывести список активных буферов на экран (C-x C-b).
- Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер.
- Закройте это окно (C-x 0).
- Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран (C-x b).
- Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2).
- В каждом из четырёх созданных окон откройте новый буфер (файл) и введите несколько строк текста.
- Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте.
- Переключайтесь между результатами поиска, нажимая C-s.
- Выйдите из режима поиска, нажав C-g.
- Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены.

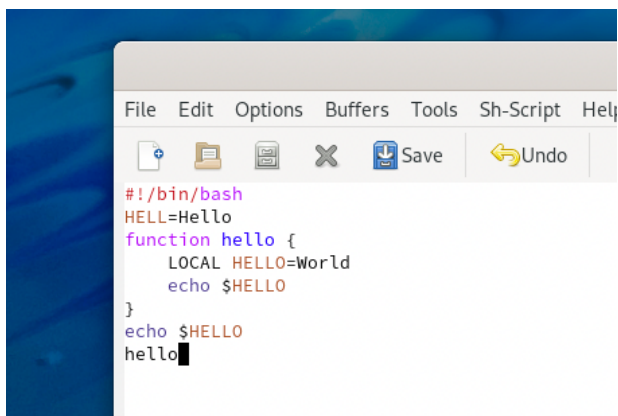
- Испробуйте другой режим поиска, нажав M-s o. Объясните, чем он отличается от обычного режима?

Выполнение лабораторной работы

Открываем emacs. Создаем файл lab07.sh с помощью комбинации C-x C-f, записываем туда текст и сохраняем файл с помощью C-x C-s



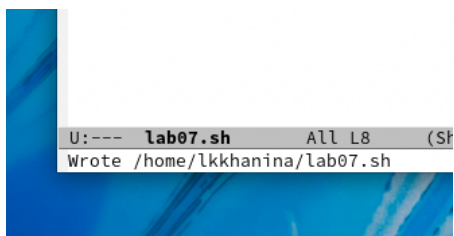
Открываем emacs



A screenshot of a text editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for new, open, save, close, and undo. The editor contains the following text:

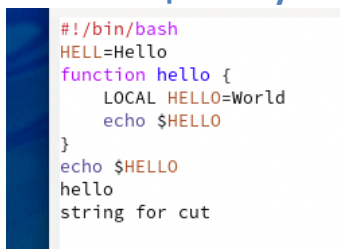
```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Записываем текст



Сохраняем

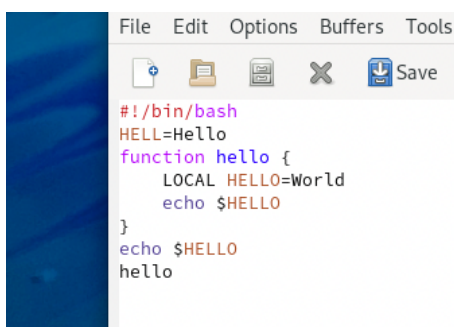
Вырезаем командой C-k целую строку. Вставляем строку в конец файла комбинацией C-y



A screenshot of a text editor window showing the same script as before, but with an additional line at the end:

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

Есть строка

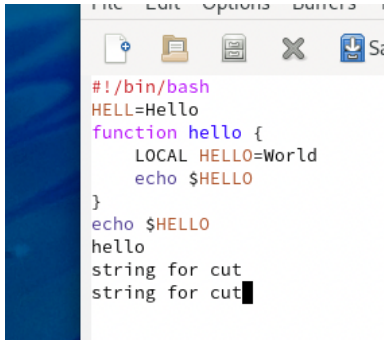


A screenshot of a text editor window showing the same script as before, but with an additional line at the end:

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

Нет строки

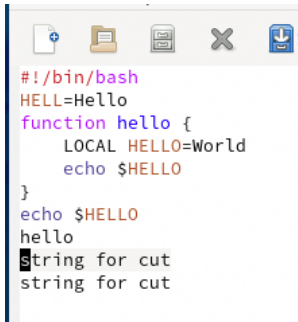
Выделяем область текста с помощью C-space, копируем ее в буфер обмена с помощью Alt-w и вставляем в конец файла с помощью C-y



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

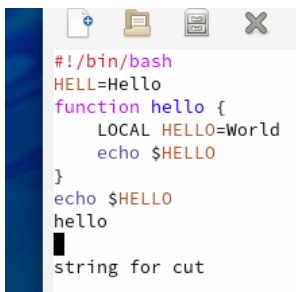
Добавили скопированную строку

Вновь выделяем эту область и уже удаляем с помощью C-w



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

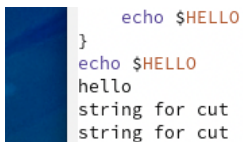
Есть строка



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

Нет строки

Отменим последние действия с помощью сочетания C-/



```
echo $HELLO
}
echo $HELLO
hello
string for cut
string for cut
```

Строка вернулась

С помощью команды C-a перемещаем курсор в начало строки, а с помощью команды C-e — в конец

```
hello
string for cut
string for cut
```

Курсор в конце

```
echo $HELLO
hello
string for cut
String for cut
```

Курсор в начале

```
hello
string for cut
string for cut
```

Курсор в конце

С помощью команды M-< перемещаем курсор в начало буфера, а с помощью команды M-> — в конец

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

Курсор в начале

```
echo $HELLO
}
echo $HELLO
hello
string for cut
string for cut
```

Курсор в конце (не запечатлен)

С помощью C-x C-b выводим список активных буферов на экран.

Перемещаемся во вновь открытое окно с помощью C-x и закрываем его с помощью C-x 0

```
-:***- lab07.sh All L11 (Shell-script[bash])
CRM Buffer Size Mode File
* lab07.sh 128 Shell-script[... ~/lab07.sh
% *GNU Emacs* 727 Fundamental
*scratch* 145 Lisp Interaction
%* *Messages* 562 Messages
```

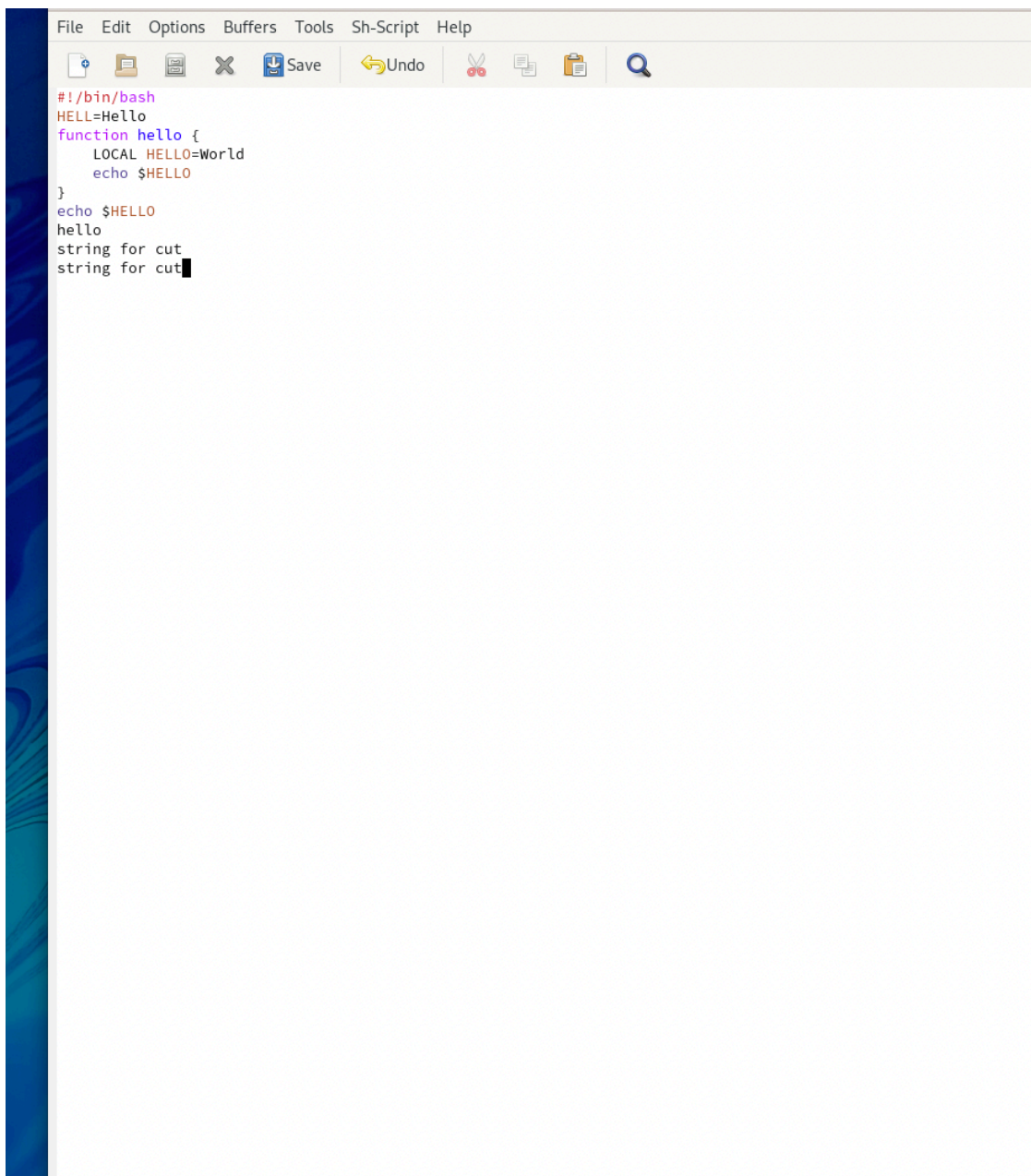
Список активных буферов

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

```

CRM Buffer	Size	Mode	File
. * lab07.sh	128	Shell-script[... ~/lab07.sh	
% *GNU Emacs*	727	Fundamental	
scratch	145	Lisp Interaction	
%* *Messages*	562	Messages	

Переключаемся на окно



The image shows a screenshot of a Sh-Script editor window. The window has a menu bar at the top with the following items: File, Edit, Options, Buffers, Tools, Sh-Script, and Help. Below the menu bar is a toolbar with icons for opening a file, saving, closing, undo, redo, and search. The main editing area contains the following text:

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

Заркываем окно

Затем переключаемся между буферами, но уже с помощью C-x b

```
hello
string for cut
string for cut7.3

[]
```

CRM Buffer	Size	Mode	File
. * lab07.sh	133	Shell-script[... ~/lab07.sh	
% *GNU Emacs*	727	Fundamental	
scratch	145	Lisp Interaction	
%* *Messages*	792	Messages	

Опять переключаемся на окно

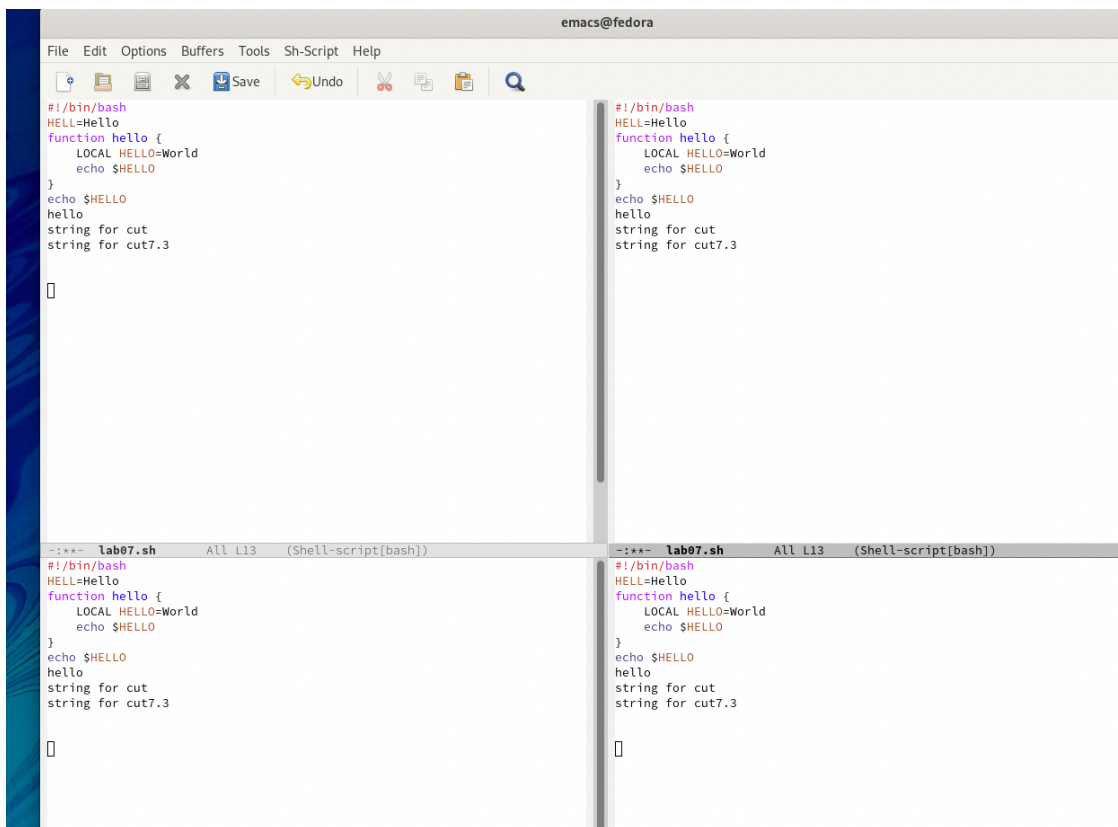
Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали(C-x 2)

The screenshot shows the Emacs editor interface with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar. The main editing area is split vertically into two windows, each containing a shell buffer. The left window shows the following code:

```
#!/bin/bash
HELL=Hello
function hello {
  LOCAL HELLO=World
  echo $HELLO
}
echo $HELLO
hello
string for cut
string for cut7.3
```

The right window shows the same code. Both windows have a cursor at the end of the last line.

Поделим на 2 части



Поделит на 4 части

В каждой части откроем файл с помощью C-x C-f

The screenshot shows the Emacs editor interface with two panes. The left pane displays the contents of the directory `/home/lkhanina/work:` using `ls -l`. The right pane displays the contents of the file `/home/lkhanina/work/blog:` using `ls -l`.

```
U:~% work All L5 (Dired by name)
module github.com/wowchemy/starter-hugo-academic

go 1.15

require (
  github.com/wowchemy/wowchemy-hugo-modules/wowchemy-cms/v5 v5.0.0-2020
  220423180919-17d5d3f0ca43 // indirect
  github.com/wowchemy/wowchemy-hugo-modules/wowchemy/v5 v5.0.0-2022042
  23180919-17d5d3f0ca43 // indirect
)

U:~% file.txt All L1 (Text)
/home/lkhanina/work/blog:
итого 260
drwxrwxr-x. 1 lkhanina lkhanina 390 мая 7 22:23 .
drwxrwxr-x. 1 lkhanina lkhanina 56 мая 14 09:04 ..
-rw-rw-r--. 1 lkhanina lkhanina 258 мая 7 21:29 academic.Rproj
drwxrwxr-x. 1 lkhanina lkhanina 36 мая 7 22:23 assets
drwxrwxr-x. 1 lkhanina lkhanina 16 мая 7 21:29 config
drwxrwxr-x. 1 lkhanina lkhanina 134 мая 7 21:29 content
drwxrwxr-x. 1 lkhanina lkhanina 54 мая 7 21:29 data
-rw-rw-r--. 1 lkhanina lkhanina 298 мая 7 21:29 editorconfig
drwxrwxr-x. 1 lkhanina lkhanina 152 мая 7 22:05 .git
drwxrwxr-x. 1 lkhanina lkhanina 40 мая 7 21:29 .github
-rw-rw-r--. 1 lkhanina lkhanina 94 мая 7 22:04 .gitignore
-rw-rw-r--. 1 lkhanina lkhanina 103 мая 7 22:05 .gitmodules
-rw-rw-r--. 1 lkhanina lkhanina 279 мая 7 21:29 go.mod
-rw-rw-r--. 1 lkhanina lkhanina 570 мая 7 22:23 go.sum
-rw-rw-r--. 1 lkhanina lkhanina 0 мая 7 22:23 hugo_build.lock
drwxrwxr-x. 1 lkhanina lkhanina 40 мая 7 21:29 images
-rw-rw-r--. 1 lkhanina lkhanina 1078 мая 7 21:29 LICENSE.md
drwxrwxr-x. 1 lkhanina lkhanina 236 мая 7 21:53 mysite
-rw-rw-r--. 1 lkhanina lkhanina 479 мая 7 21:29 netlify.toml
-rw-rw-r--. 1 lkhanina lkhanina 222122 мая 7 21:29 preview.png
drwxrwxr-x. 1 lkhanina lkhanina 486 мая 7 22:23 public
-rw-rw-r--. 1 lkhanina lkhanina 3934 мая 7 21:29 README.md
drwxrwxr-x. 1 lkhanina lkhanina 8 мая 7 22:23 resources
drwxrwxr-x. 1 lkhanina lkhanina 14 мая 7 21:29 static
-rw-rw-r--. 1 lkhanina lkhanina 1388 мая 7 21:29 theme.toml
```

Открываем файлы

В режиме поиска найдем несколько слов с помощью C-s. Переключимся между результатами поиска с помощью C-s и выйдем из режима с помощью C-g

```
U:%%- work All L5 (Dired by name)
module github.com/wowchemy/starter-hugo-academic

go 1.15

require (
  github.com/wowchemy/wowchemy-hugo-modules/wowchemy-cms/v5 v5.0.0-20220423180919-17d5d3f0ca43 // indirect
  github.com/wowchemy/wowchemy-hugo-modules/wowchemy/v5 v5.0.0-20220423180919-17d5d3f0ca43 // indirect
)

-:--- go.mod All L8 Git-main (Modula-2 Isearch)
Failing I-search: git
```

Режим поиска

```
U:%%- work All L5 (Dired by name)
module github.com/wowchemy/starter-hugo-academic

go 1.15

require (
  github.com/wowchemy/wowchemy-hugo-modules/wowchemy-cms/v5 v5.0.0-20220423180919-17d5d3f0ca43 // indirect
  github.com/wowchemy/wowchemy-hugo-modules/wowchemy/v5 v5.0.0-20220423180919-17d5d3f0ca43 // indirect
)

-:*** go.mod All L6 Git-main (Modula-2 Isearch)
Wrapped I-search: git
```

Переключаемся в режиме поиска

В режиме поиска и замены (Alt-%) введем текст, который нужно найти и заменить, введем текст-замену и подтвердим ее

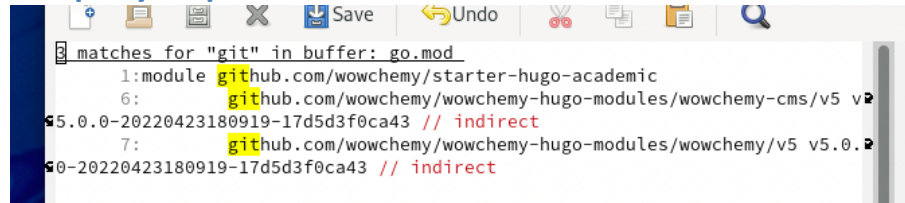
```
new file
i wanna change the word world
world is gonnna be removed
```

Было

```
new file
i wanna change the word world
it has been removed
```

Стало

Попробуем режим поиска Alt-s o



The screenshot shows the Emacs search results window. The title bar reads "3 matches for 'git' in buffer: go.mod". The results are listed as follows:

- 1: module github.com/wowchemy/starter-hugo-academic
- 6: github.com/wowchemy/wowchemy-hugo-modules/wowchemy-cms/v5 v5.0.0-20220423180919-17d5d3f0ca43 // indirect
- 7: github.com/wowchemy/wowchemy-hugo-modules/wowchemy/v5 v5.0.0-20220423180919-17d5d3f0ca43 // indirect

Другой режим поиска

Выводы

Я научилась работать в emacs. Познакомилась с режимами работы над буферами и с режимами работы над текстом