

Лабораторная работа №9

Ханина Людмила Константиновна

Содержание

Цель работы

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.

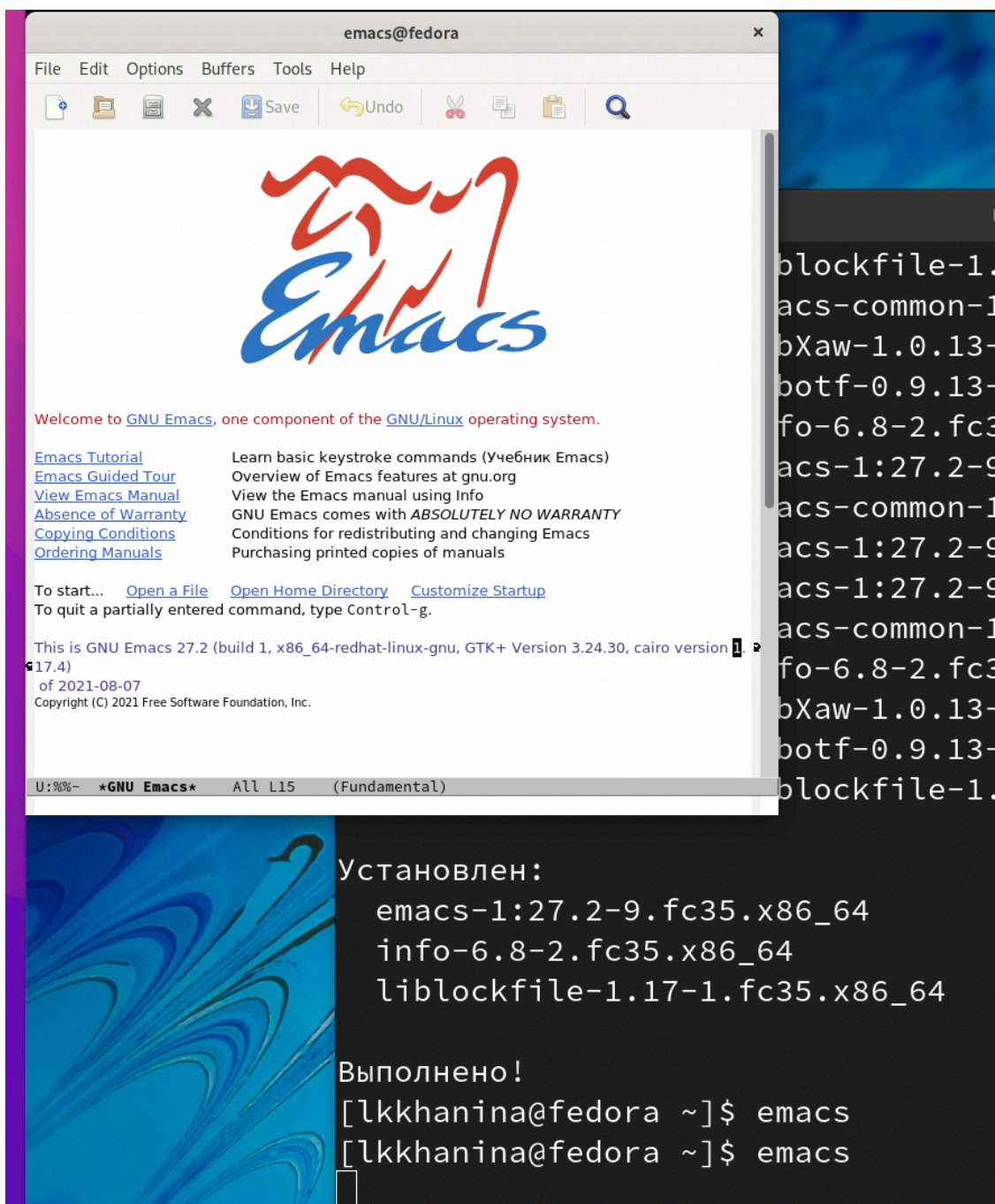
- Выйдите из режима поиска, нажав C-g.
- Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены.
- Испробуйте другой режим поиска, нажав M-s o. Объясните, чем он отличается от обычного режима?

Теоретическое введение

Команда	Значение команды
C-x C-f	Открыть файл
C-x C-s	Сохранить текст в буфер
C-x C-b	Отобразить список открытых буферов в новом окне
C-x 0	Закрыть текущее окно (при этом буфер не удаляется)
C-w	Удалить выделенную область текста в список удалений
C-y	Вставить текст из списка удалений в текущую позицию курсора

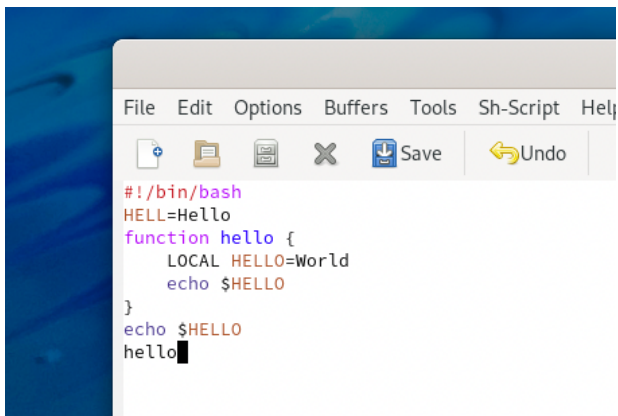
Выполнение лабораторной работы

1. Открываем emacs.



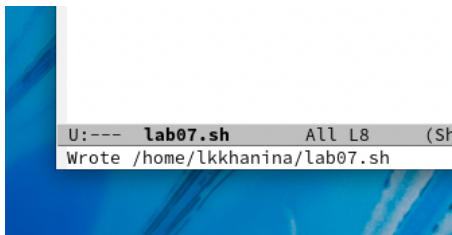
Открываем emacs

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



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

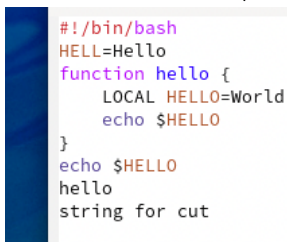
Записываем текст



```
U:--- lab07.sh All L8 (Sf)
Wrote /home/lkhanina/lab07.sh
```

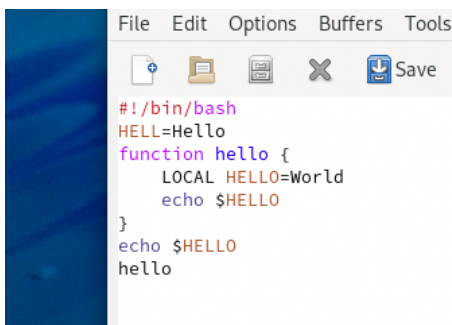
Сохраняем

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



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

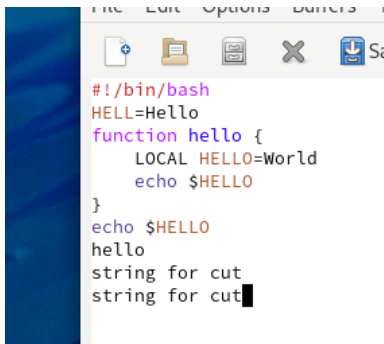
Есть строка



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

Нет строки

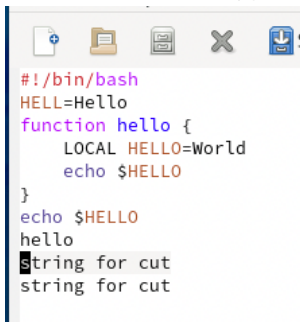
4. Выделяем область текста с помощью C-спре, копируем ее в буфер обмена с помощью Alt-w и вставляем в конец файла с помощью C-y.



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

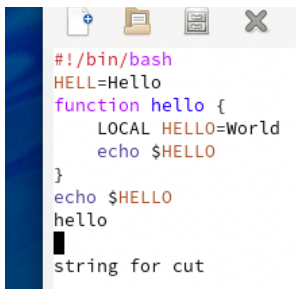
Добавили скопированную строку

5. 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
```

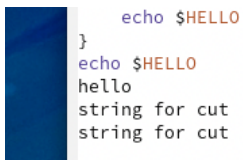
Есть строка



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

Нет строки

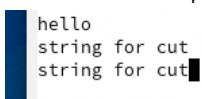
6. Отменим последние действия с помощью сочетания C-/.



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

Строка вернулась

7. 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

8. 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)

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

```
-:***- 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

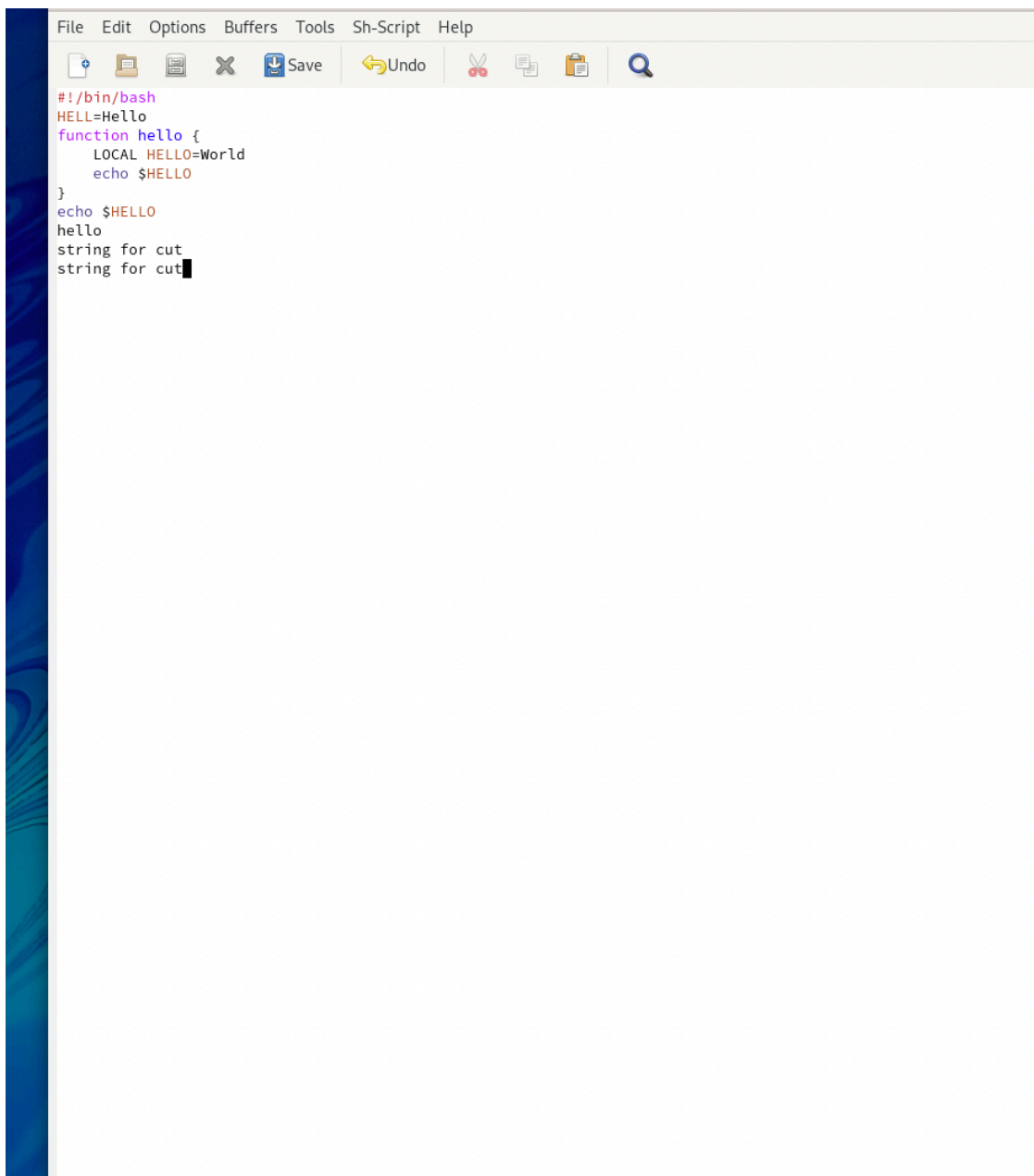
10. Перемещаемся во вновь открытое окно с помощью C-x и закрываем его с помощью C-x 0.

```
#!/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	

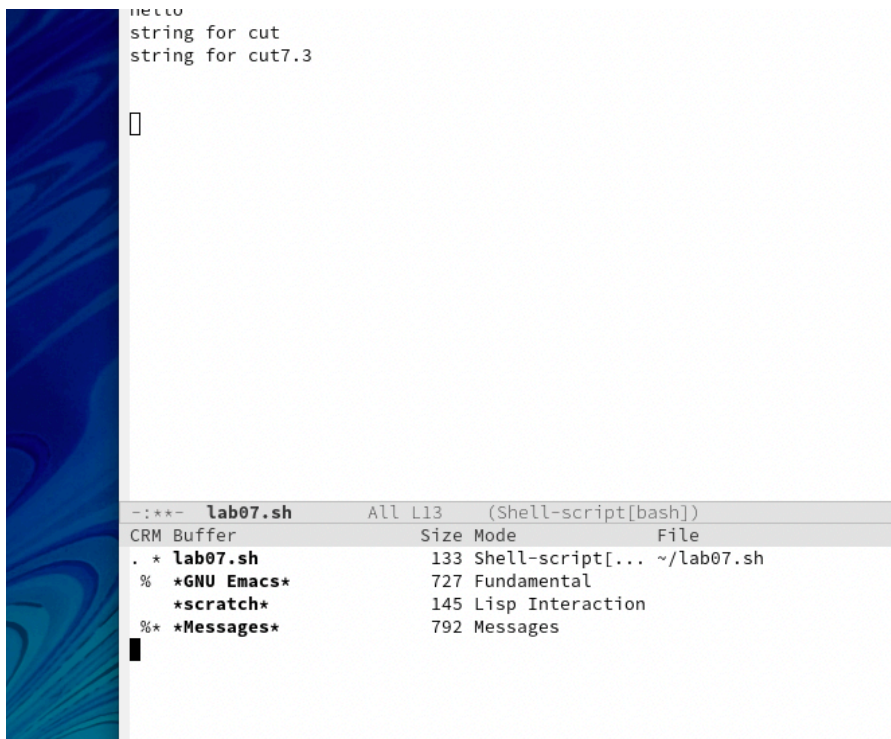
Переключаемся на окно

A screenshot of a Sh-Script editor window. The window has a menu bar with 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with icons for file operations (new, open, save, close), editing (undo, redo, cut, copy, paste), and search. The main text area contains a bash script with the following content:

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

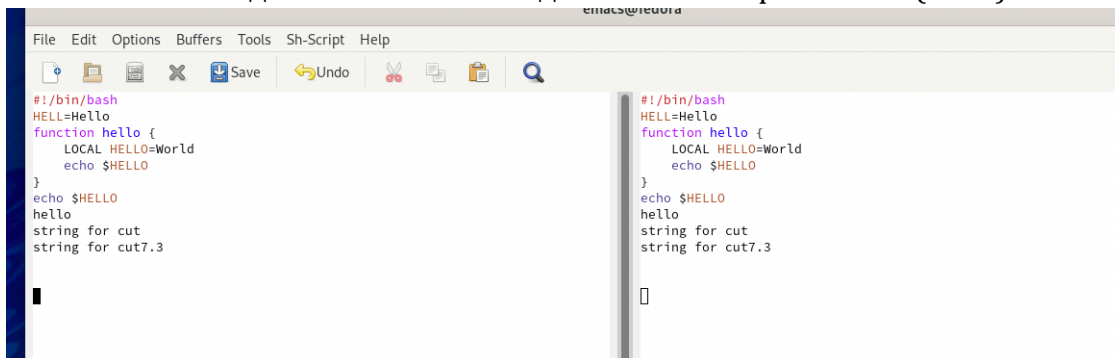
Зарываем окно

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

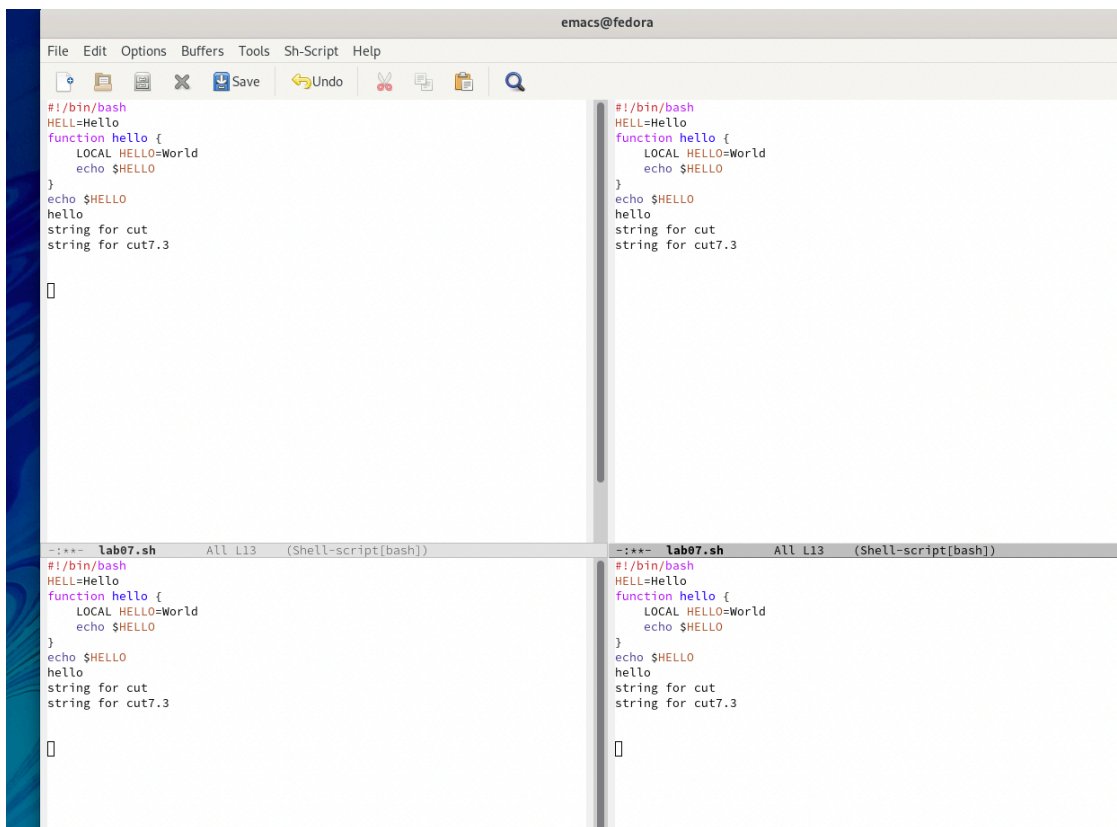


Опять переключаемся на окно

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

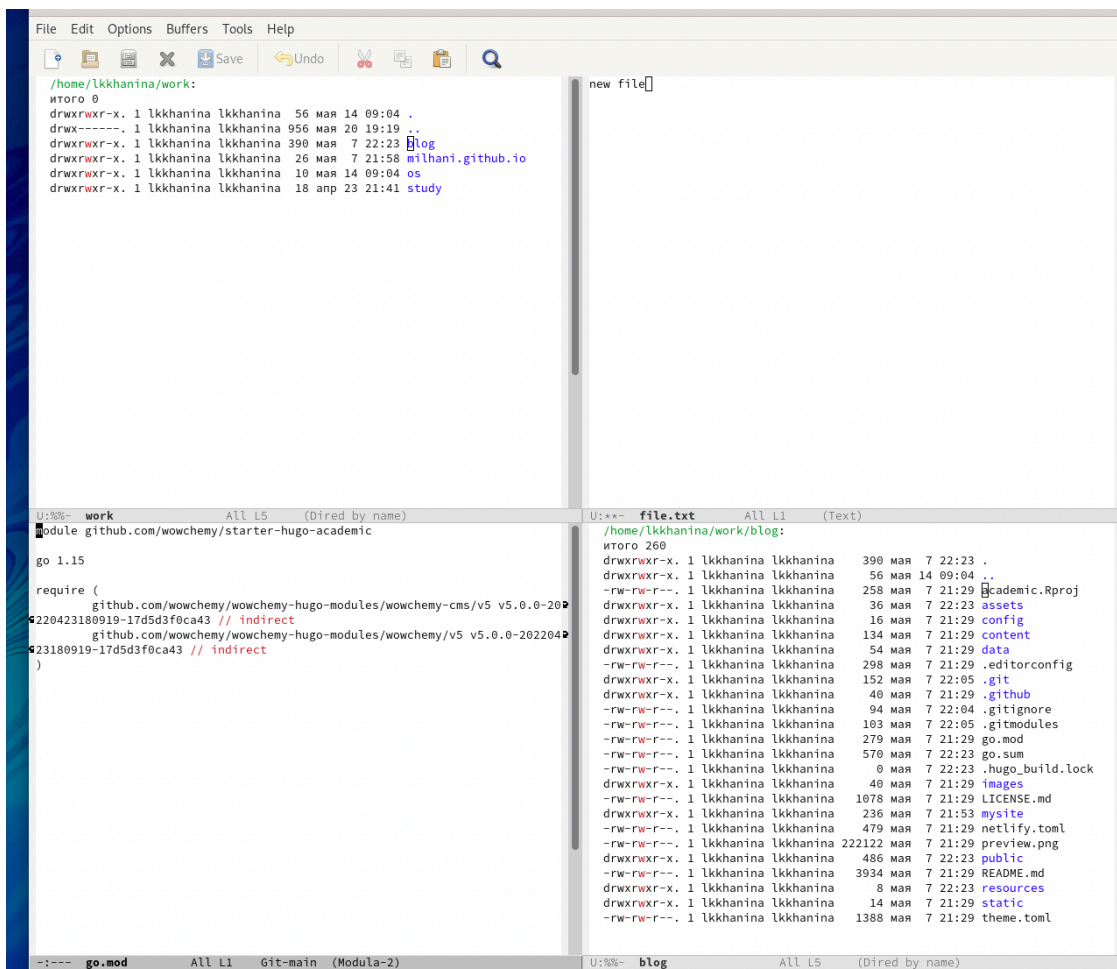


Поделим на 2 части



Поделим на 4 части

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



Открываем файлы

14. В режиме поиска найдем несколько слов с помощью C-s.

```
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

15. 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 L6 Git-main (Modula-2 Isearch)
Wrapped I-search: git
```

Переключаемся в режиме поиска

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

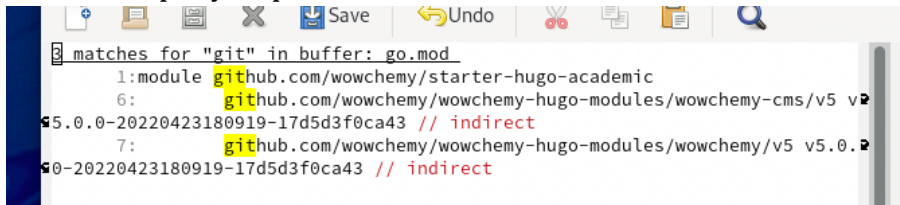
```
new file
i wanna change the word world
world is gonnna be removed
```

Было

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

Стало

17. Попробуем режим поиска Alt-s o.



Другой режим поиска

Контрольные вопросы

1. Emacs представляет собой мощный экраннй редактор текста, написанный на языке высокого уровня Elisp.
2. Проблему могут возникнуть из-за необходимости пользоваться горячими клавишами.
3. Буфер — объект, представляющий какой-либо текст. Окно — это область экрана, в которой отображается буфер.
4. Да, можно.
5. При запуске Emacs по умолчанию создаются буферы: scratch и Help, messages.
6. 'С-с |': зажимаю Ctrl, нажимаю с, отпускаю и нажимаю |. 'С-с С- |': на зажимаю Ctrl, нажимаю с, отпускаю, далее, удерживая Ctrl, нажимаю |.
7. Поделить окно на две части можно с помощью С-х 3 по вертикали и/или С-х 2 по горизонтали.
8. Настройки Emacs хранятся в файле .emacs.
9. Данная клавиша удаляет символ перед курсором. Чтобы ее переназначить, необходимо изменить конфигурацию файла .emacs.
10. Мне более удобным показался vim, так как с помощью функций на панелях можно сделать все необходимое. А вот если ты забудешь сочетания клавиш,

работая в emacs, то придется поднимать документацию, так как меню довольно неинформативное.

Выводы

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