

Отчет по лабораторной работе №11

Операционные системы

Палкина Нина Сергеевна

Содержание

1	Цель работы.....	1
2	Задание	1
3	Теоретическое введение.....	1
4	Выполнение лабораторной работы.....	2
5	Выводы	11
6	Ответы на контрольные вопросы	11

1 Цель работы

Цель данной лабораторной работы - познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

2 Задание

1. Ознакомиться с теоретическим материалом.
2. Ознакомиться с редактором emacs.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы

3 Теоретическое введение

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.

4 Выполнение лабораторной работы

Открываю Emacs через терминал (рис. 1).

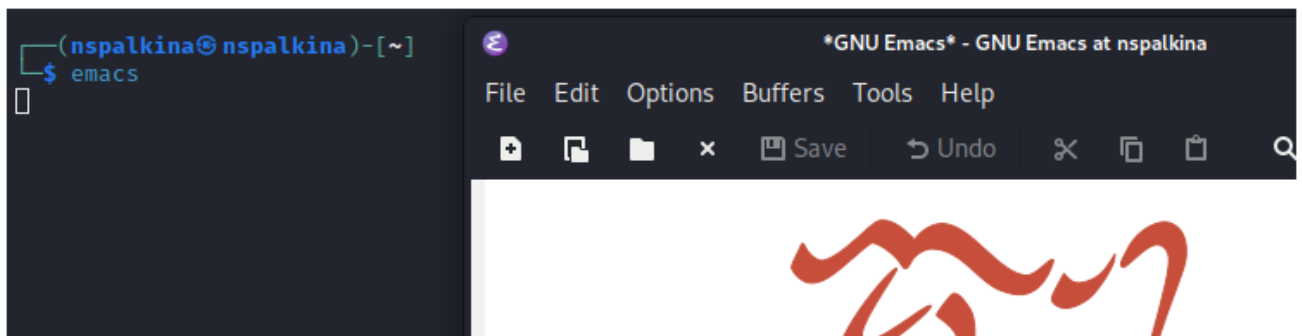


Figure 1: Открытие программы

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

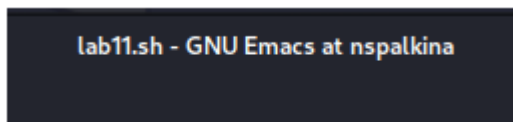
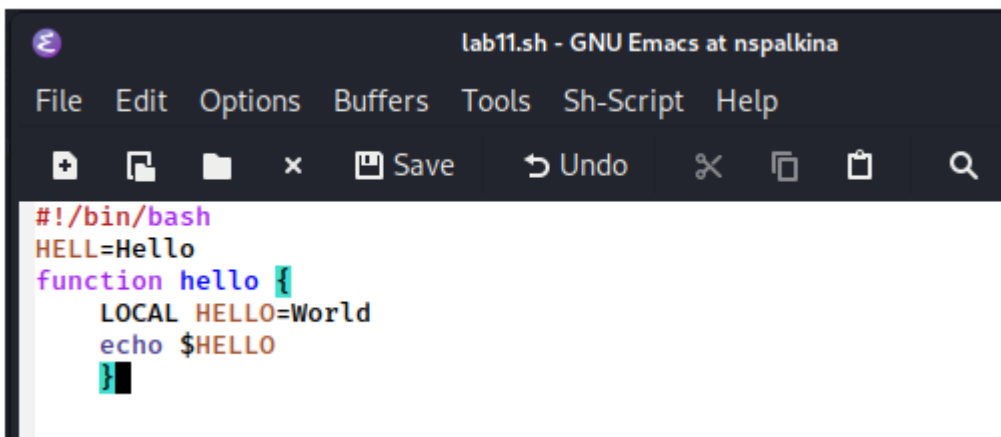


Figure 2: Создание файла

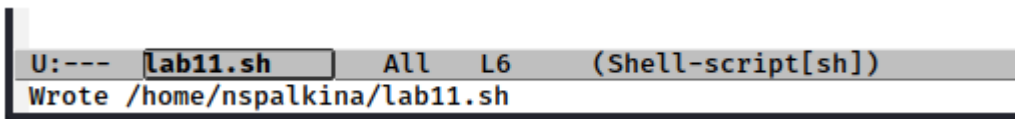
Прописываю в файле текст программы (рис. 3).



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

Figure 3: Редактирование файла

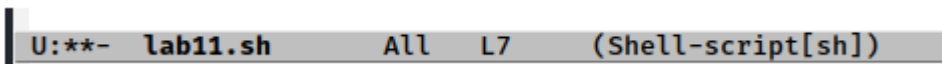
Сохраняю файл с помощью комбинации C-x C-s (рис. 4).



```
U:--- lab11.sh All L6 (Shell-script[sh])
Wrote /home/nspalkina/lab11.sh
```

Figure 4: Сохранение изменений в файле

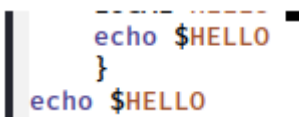
Вырезаю одной командой целую строку (C-k) (рис. 5).



```
U:**- lab11.sh All L7 (Shell-script[sh])
```

Figure 5: Вырезание строки

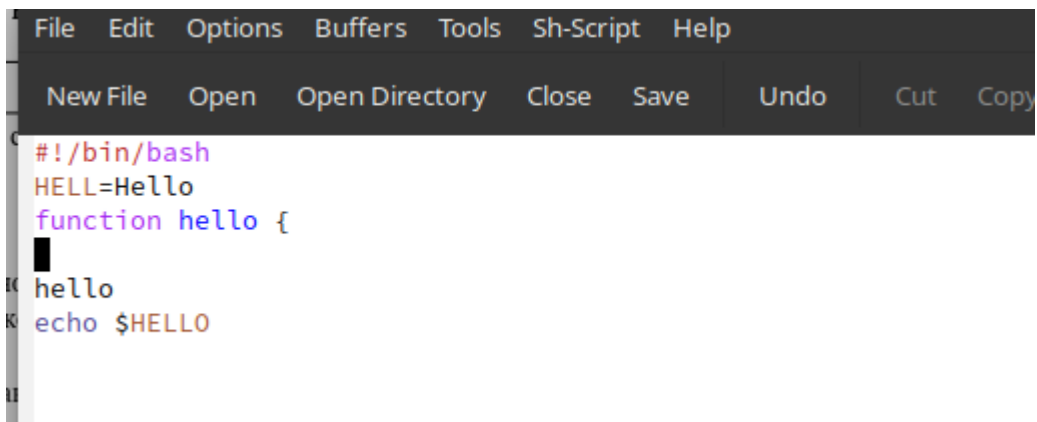
Вставляю эту строку в конец файла (C-y) (рис. 6).



```
echo $HELLO
}
echo $HELLO
```

Figure 6: Вставка строки в конце файла

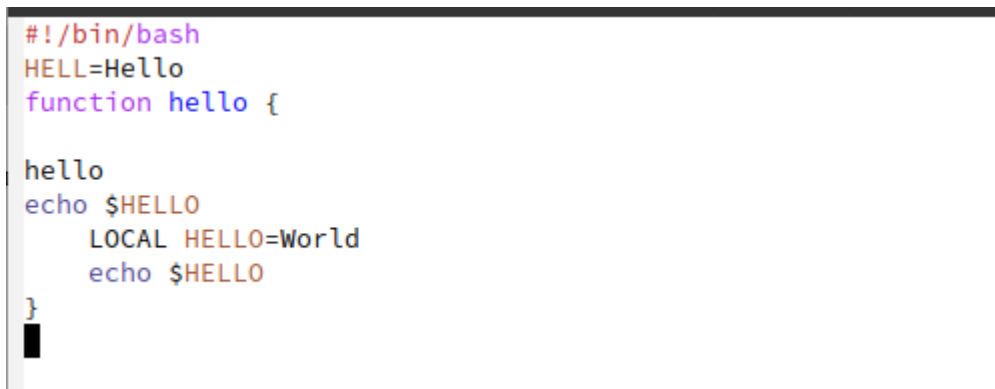
Выделяю область текста (C-space), копирую область в буфер обмена (M-w), вырезаю эту область с помощью C-w (рис. 7).



```
File Edit Options Buffers Tools Sh-Script Help
New File Open Open Directory Close Save Undo Cut Copy
#!/bin/bash
HELL=Hello
function hello {
hello
echo $HELLO
```

Figure 7: Вырезанная область

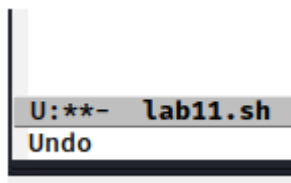
Вставляю область в конец файла (C-y)(рис. 8).



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

Figure 8: Вставка в конец файла

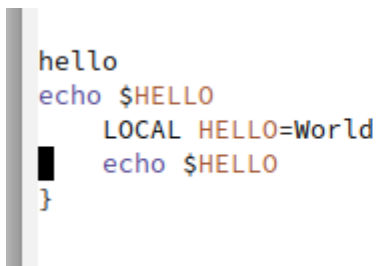
Отменяю последнее действие C-/ (рис. 9).



```
U:**- lab11.sh
Undo
```

Figure 9: Отмена последнего действия

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



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

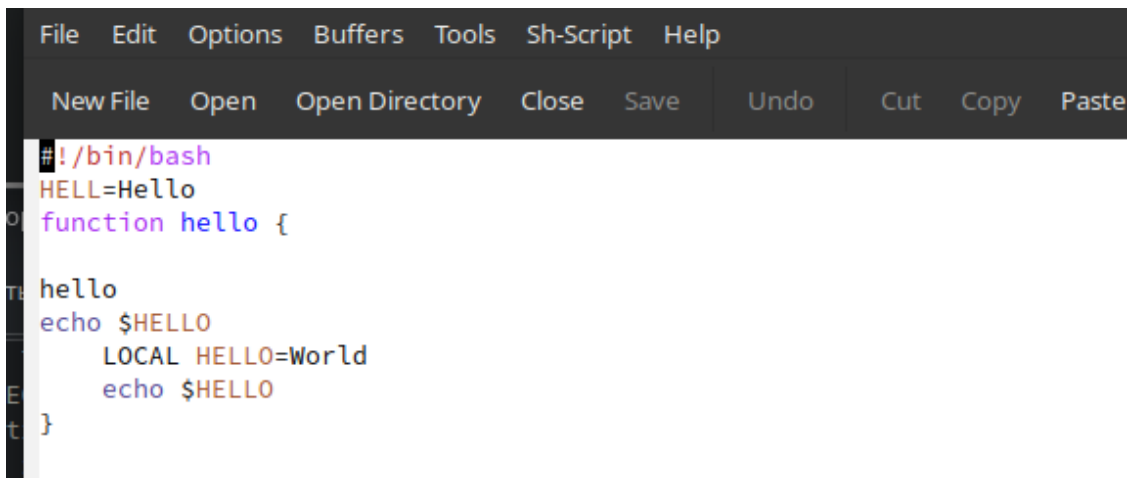
Figure 10: Курсор в начале строки

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

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

Figure 11: Курсор в конце строки

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

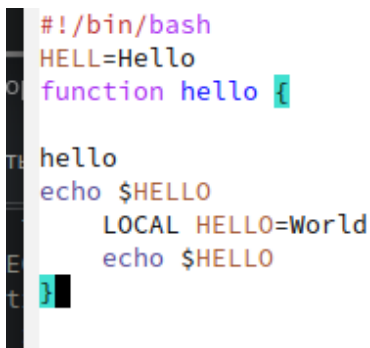


The screenshot shows a terminal window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar (New File, Open, Open Directory, Close, Save, Undo, Cut, Copy, Paste). The script content is as follows:

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

Figure 12: Начало буфера

Перемещаю курсор в конец файла M-> (рис. 13).



The screenshot shows the same terminal window as Figure 12, but the cursor is now at the end of the script, after the closing brace of the function definition.

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

Figure 13: Конец буфера

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

CRM	Buffer	Size	Mode	File
1	* lab11.sh	128	Shell-script[sh]	~/lab11.sh
%	*GNU Emacs*	734	Fundamental	
	scratch	145	Lisp Interaction	
%*	*Messages*	1025	Messages	
%*	*Async-native-compile-log*	704	Fundamental	

Figure 14: Список активных буферов

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

```

C-<chr> -- следует удерживать клавишу CONTROL, пока набирается с
<chr>. Так, C-f должно означать: одновременно нажать кла
f.
M-<chr> -- следует удерживать клавишу META, пока набирается симв
<chr>. Если нет клавиши META, ALT или EDIT, то нажмите <
отпустите ее, а потом наберите символ <chr>.

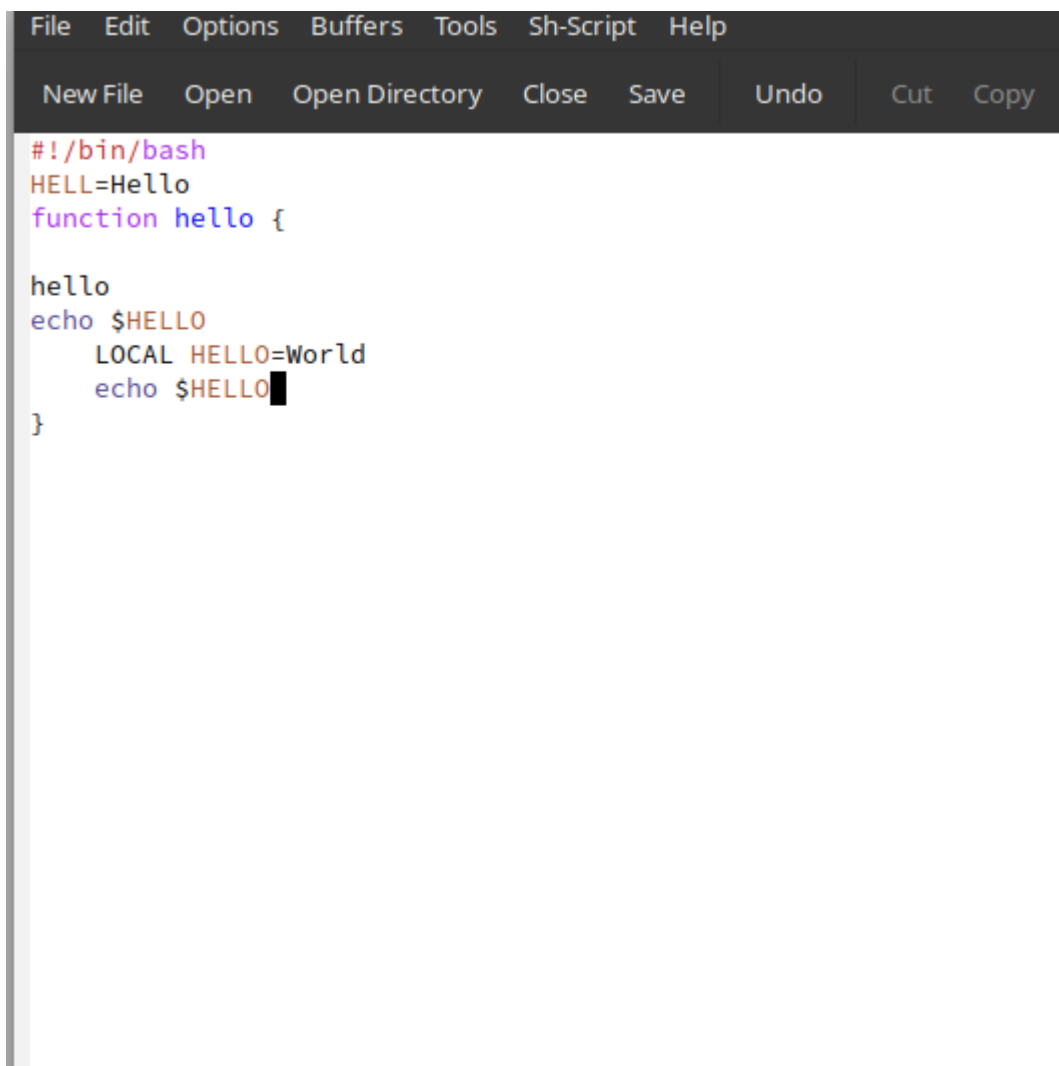
Важное замечание: для завершения работы Emacs, наберите C-x C-c (
клавиш). Чтобы прервать частично набранную команду, нажмите C-g.
Символы ">>" с левой стороны указывают, что вам нужно делать, что
попробовать применить команду. Например:

U:--- TUTORIAL.ru 1% L14 (Fundamental)

```

Figure 15: Другое окно буфера

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

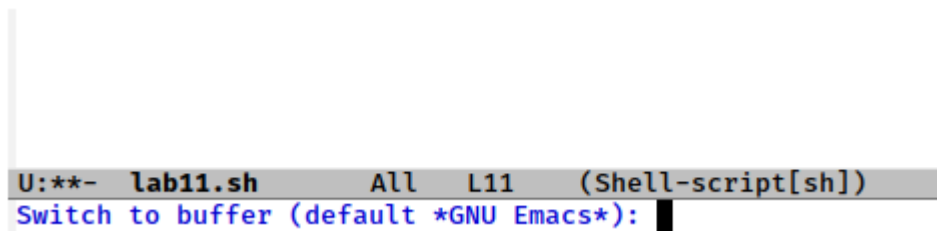


The screenshot shows the Emacs editor interface with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. Below the menu bar is a toolbar with buttons for 'New File', 'Open', 'Open Directory', 'Close', 'Save', 'Undo', 'Cut', and 'Copy'. The main editing area contains a shell script with the following content:

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

Figure 16: Заккрытие буфера

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



The screenshot shows the Emacs status bar at the bottom of the window. It displays the following information: 'U:**~ lab11.sh All L11 (Shell-script[sh])'. Below this, a prompt 'Switch to buffer (default *GNU Emacs*):' is shown with a cursor.

Figure 17: Открытие другого буфера

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).



Figure 18: Четыре окна

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

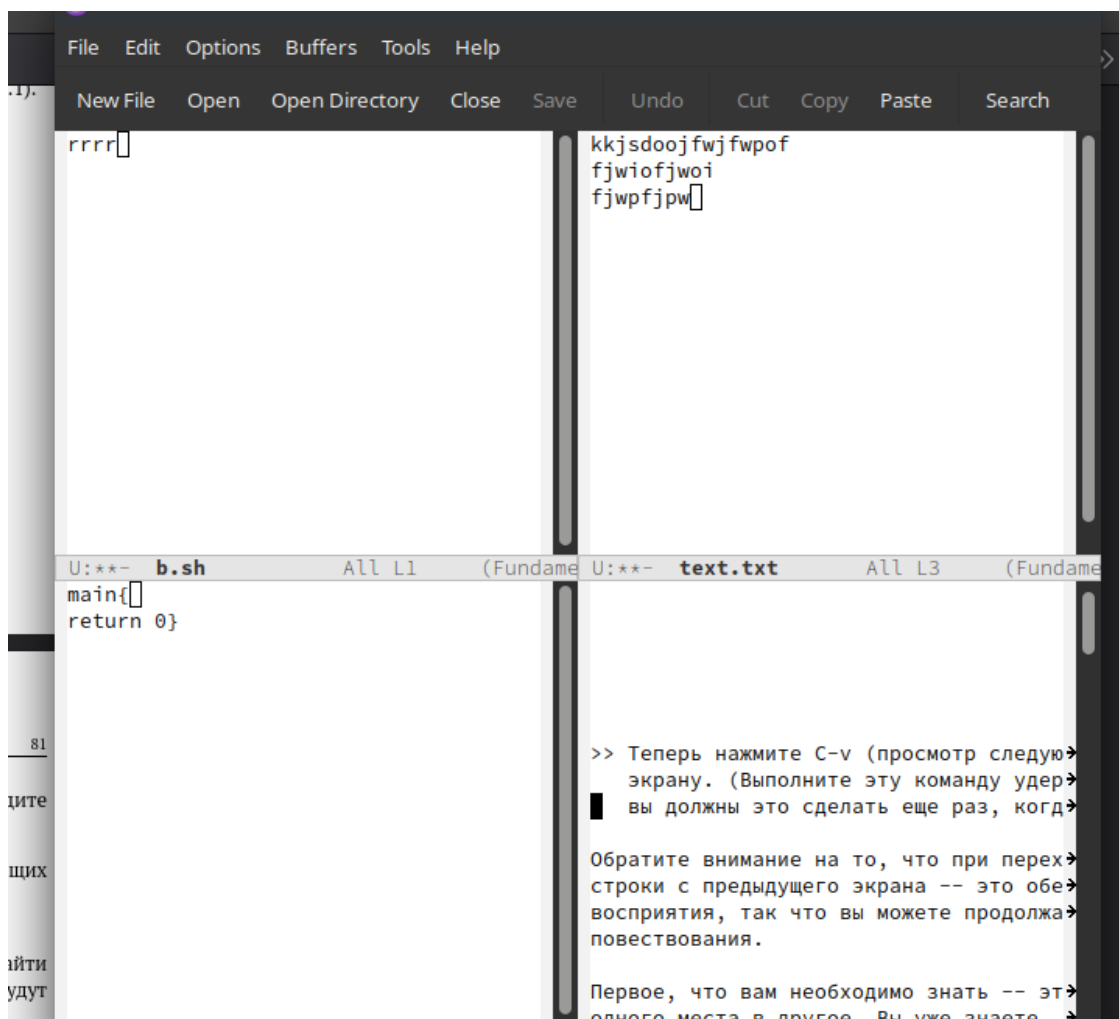


Figure 19: Открытие буферов в четырех окнах

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

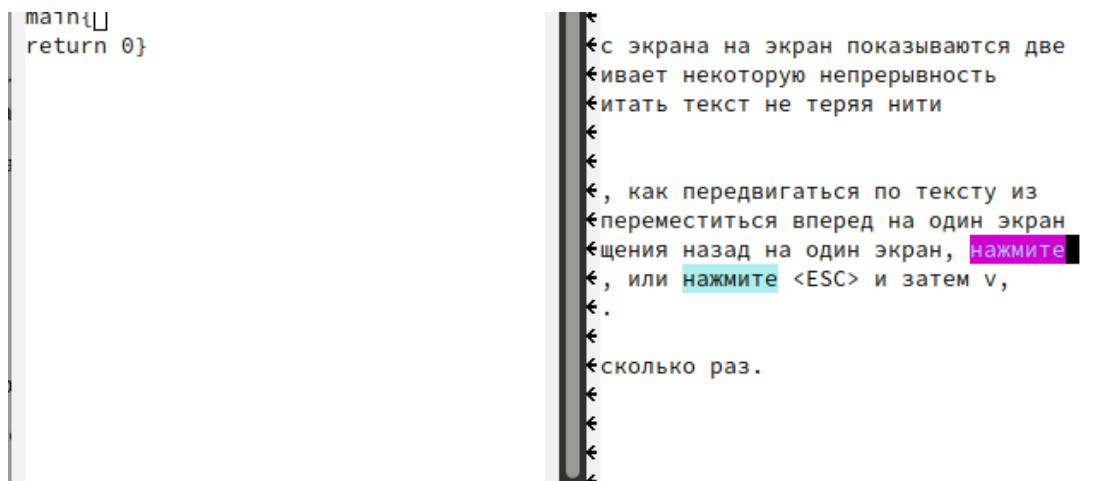


Figure 20: Режим поиска

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

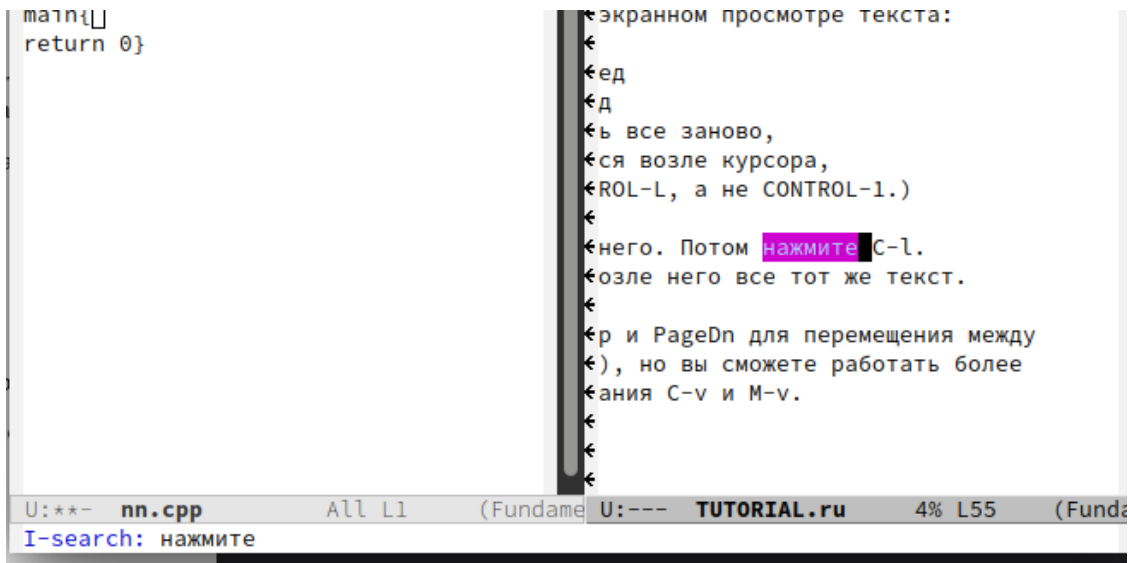


Figure 21: Перемещение по найденным выражениям

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

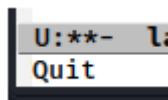


Figure 22: Выход из режима поиска

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

Query replace LOCAL with:

Figure 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  
}
```

Figure 24: Слово заменено

5 Выводы

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

6 Ответы на контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. Написан на языке высокого уровня Lisp.

2. Какие особенности данного редактора могут сделать его сложным для освоения новичком?

Большое разнообразие сложных комбинаций клавиш, которые необходимы для редактирования файла и в принципе для работа с Emacs.

3. Своими словами опишите, что такое буфер и окно в терминологии emacs'a.

Буфер - это объект в виде текста. Окно - это прямоугольная область, в которой отображен буфер.

4. Можно ли открыть больше 10 буферов в одном окне?

Да, можно.

5. Какие буферы создаются по умолчанию при запуске emacs?

Emacs использует буферы с именами, начинающимися с пробела, для внутренних целей. Отчасти он обращается с буферами с такими именами особенным образом — например, по умолчанию в них не записывается информация для отмены изменений.

6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?

Ctrl + c, а потом | и Ctrl + c Ctrl + |

7. Как поделить текущее окно на две части?

С помощью команды Ctrl + x 3 (по вертикали) и Ctrl + x 2 (по горизонтали).

8. В каком файле хранятся настройки редактора emacs?

Настройки emacs хранятся в файле .emacs, который хранится в домашней директории пользователя. Кроме этого файла есть ещё папка .emacs.

9. Какую функцию выполняет клавиша и можно ли её переназначить?

Выполняет функцию стереть, думаю можно переназначить.

10. Какой редактор вам показался удобнее в работе vi или emacs? Поясните почему.

Для меня удобнее был редактор Emacs, так как у него есть командная оболочка. А vi открывается в терминале, и выглядит своеобразно.