

Лабораторная работа №9

Текстовый редактор emacs

Лушин Артем Андреевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	24
4	Контрольные вопросы	25

Список иллюстраций

2.1	Создание файла	6
2.2	Текст в файле	6
2.3	Сохранение файла	7
2.4	Вырез строки	7
2.5	Вовращение строки	7
2.6	Выделение области и копирование	8
2.7	Убрал выделенную область	9
2.8	Отмена удаления	10
2.9	Курсор в начале строки	11
2.10	Курсор в конце строки	11
2.11	Курсор в начале буфера обмена	12
2.12	Курсор в конце буфера обмена	12
2.13	Список буферов	13
2.14	Переключение на второй буфер	14
2.15	Закрытие буфера	15
2.16	Переключение без вывода	15
2.17	Деление фрейма	16
2.18	Текст в 4 буферах	17
2.19	Режим поиска	18
2.20	Переключение между результатами	19
2.21	Переключение между результатами	20
2.22	Завершение поиска	21
2.23	Замена слова	22
2.24	Второй режим поиска	23

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1) Я создал файл lab07.sh с помощью комбинаций клавиш.

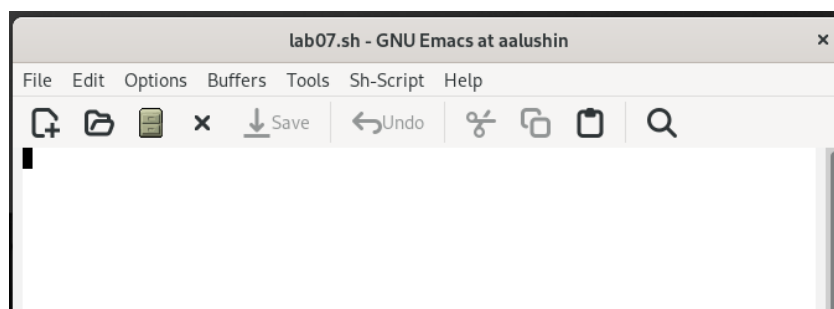


Рис. 2.1: Создание файла

2) Я набрал нужный текст в файле.

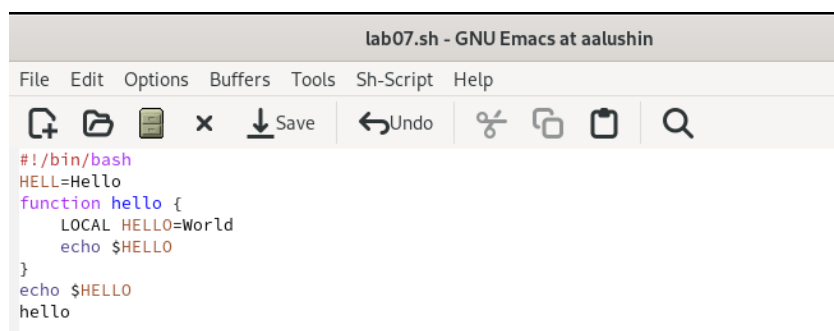


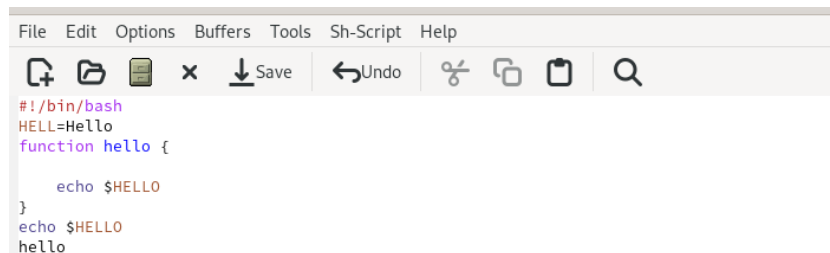
Рис. 2.2: Текст в файле

3) Сохранил файл с помощью комбинации.

```
U:--- lab07.sh All L9 (Shell-script[sh])  
(No changes need to be saved)
```

Рис. 2.3: Сохранение файла

4) Командой C-K я убрал строку.



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

Рис. 2.4: Вырез строки

5) Строку которую я вырезал, я вставил в конец файла.



```
hello  
LOCAL HELLO=World
```

Рис. 2.5: Возвращение строки

6) Я выделил область текста с помощью комбинации C-space, скопировал ее и вставил в конец текста.

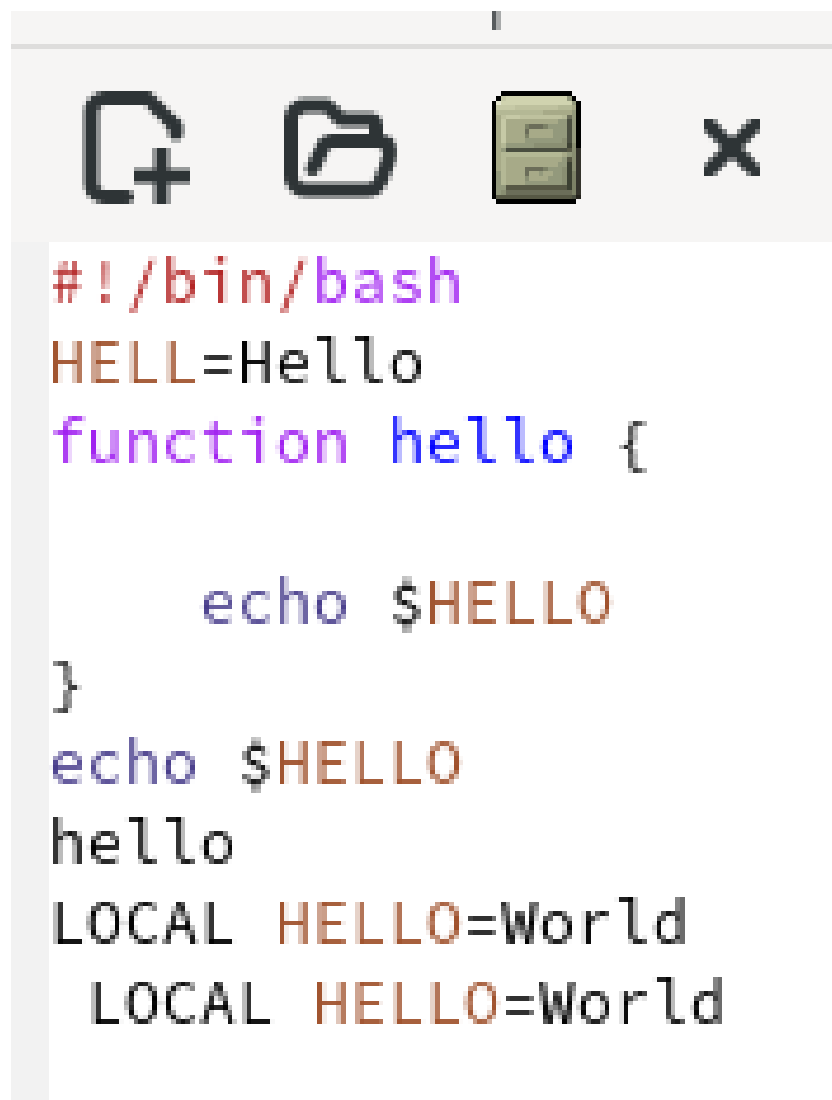
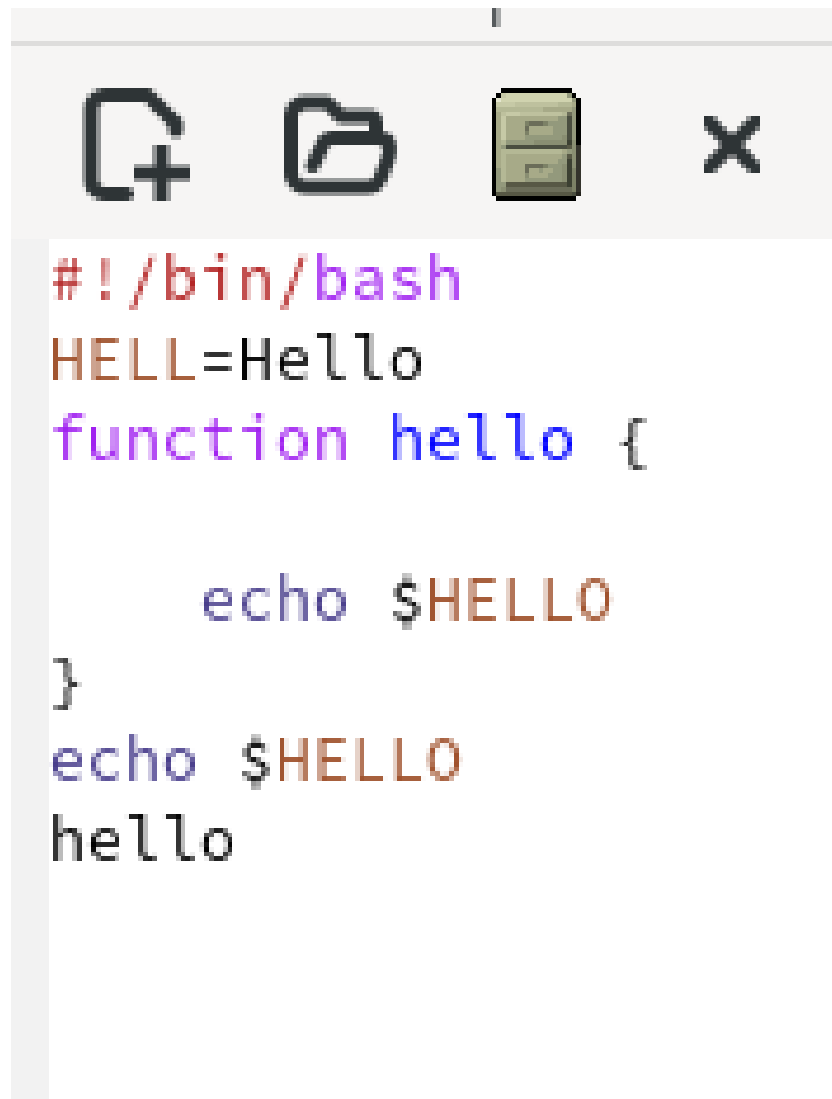


Рис. 2.6: Выделение области и копирование

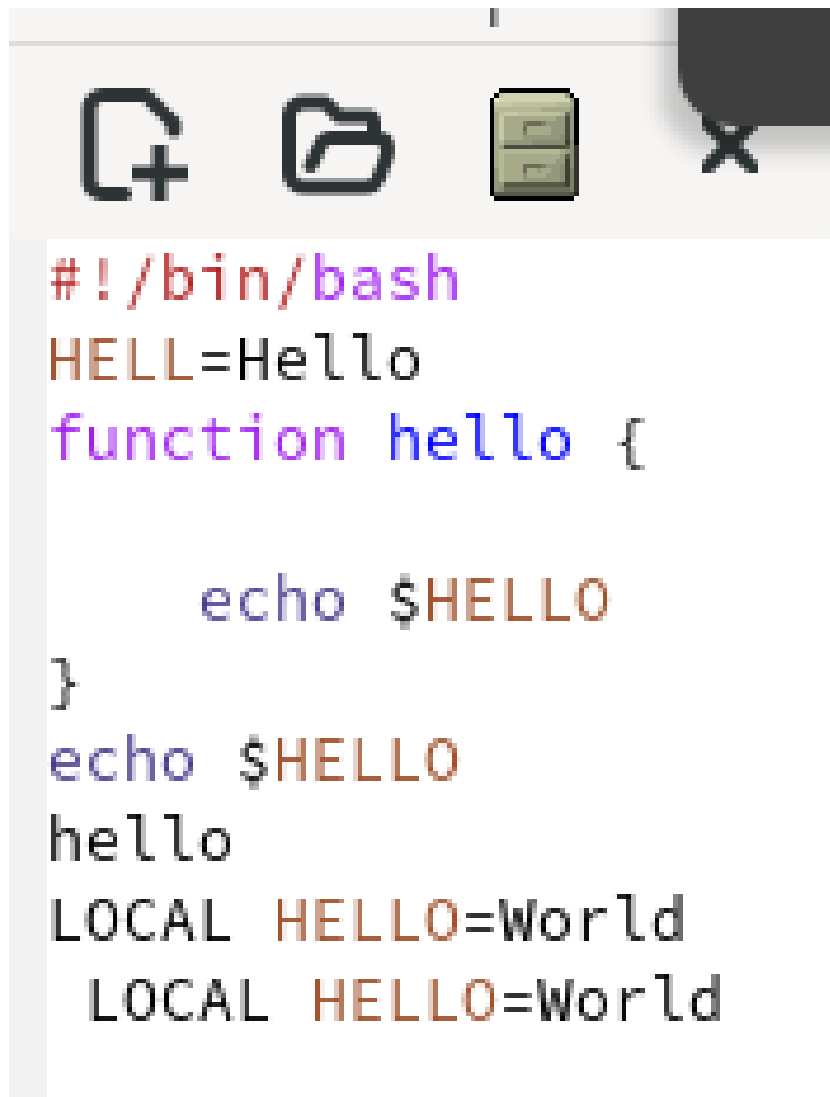
7) Выдел область которую вставил и вырезал ее с помощью комбинации C-w.



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

Рис. 2.7: Убрал выделенную область

8) Отменил удаление выделенной области через комбинацию C-/.

A terminal window with a light gray title bar and a dark gray toolbar containing icons for copy, paste, and window management. The terminal displays a bash script with the following content:

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

Рис. 2.8: Отмена удаления

- 9) Научился использовать команды для перемещения курсора. С-а - курсор в начале строки. С-е - курсор в конце строки, М-< - курсор в начале буфера обмена, и М-> - курсор в конце буфера обмена.



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

Рис. 2.9: Курсор в начале строки



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

Рис. 2.10: Курсор в конце строки



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

Рис. 2.11: Курсор в начале буфера обмена



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

Рис. 2.12: Курсор в конце буфера обмена

- 10) С помощью комбинации клавиш я вывел список активных буферов на экран.

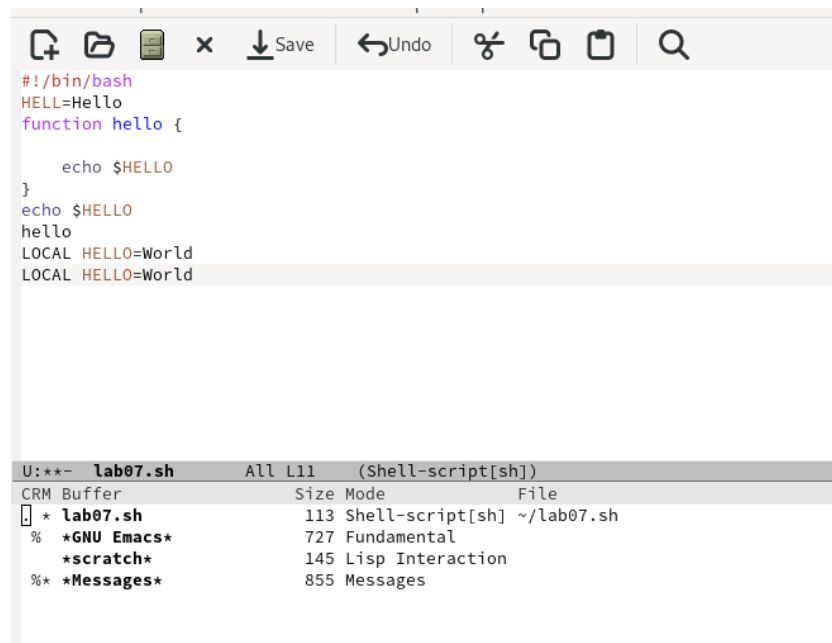
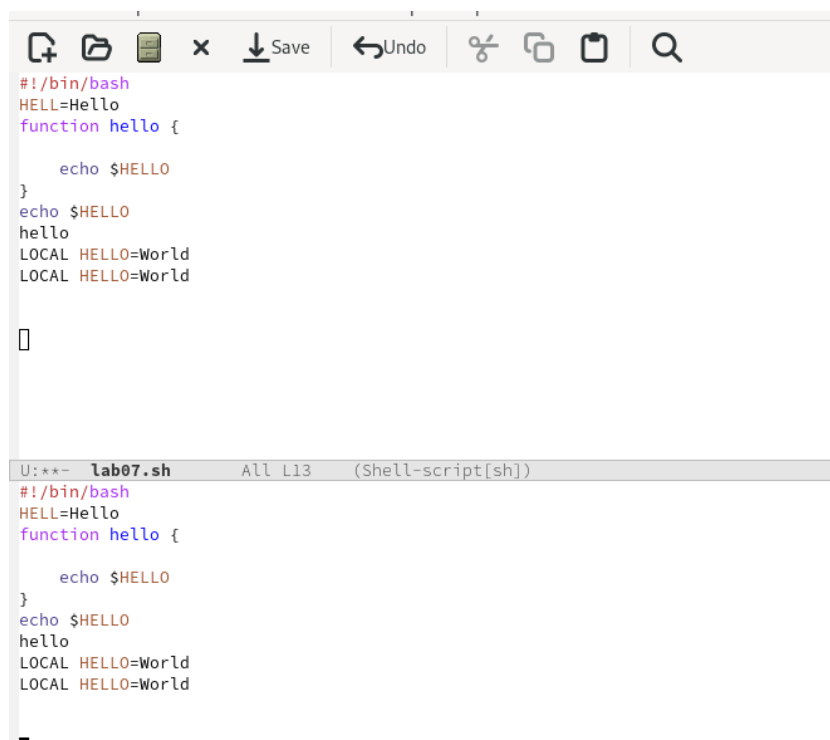


Рис. 2.13: Список буферов

11) Во втором окне переключился на буфер lab07.sh.



The image shows a code editor window with a toolbar at the top containing icons for file operations and editing. The editor contains a shell script with the following content:

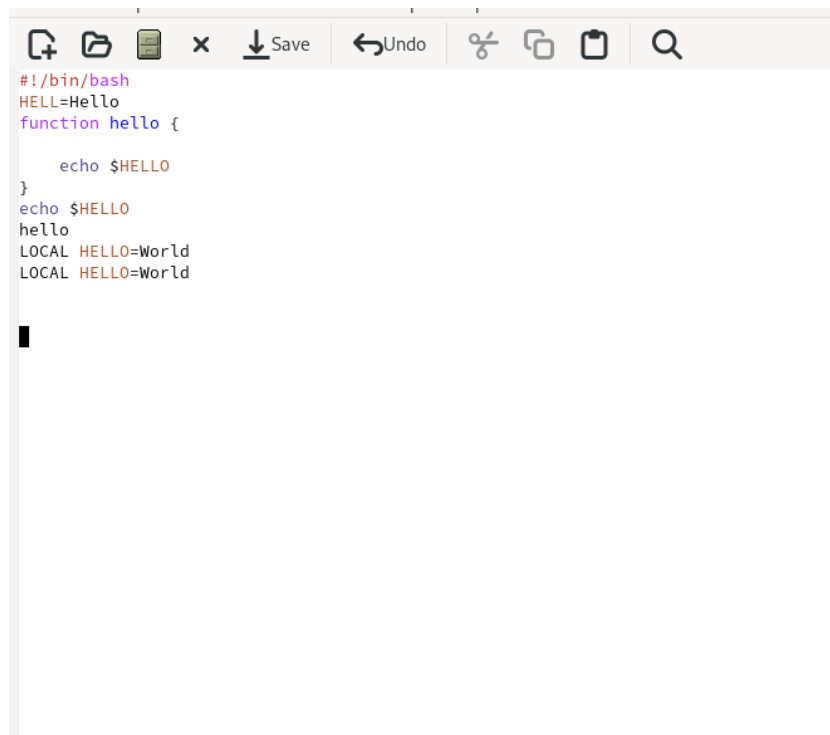
```
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
LOCAL HELLO=World
```

Below the script, there is a terminal window showing the execution of the script. The terminal prompt is 'U:*** lab07.sh' and the output is identical to the script content:

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

Рис. 2.14: Переключение на второй буфер

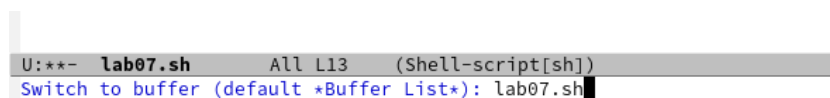
12) С помощью комбинации C-x 0, я закрыл второй буфер.



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

Рис. 2.15: Заккрытие буфера

13) Я переключился на буфер lab07.sh без вывода на экран.



```
U:***- lab07.sh All L13 (Shell-script[sh])
Switch to buffer (default *Buffer List*): lab07.sh
```

Рис. 2.16: Переключение без вывода

14) Поделил фрейм на 4 части и везде открыт файл lab07.sh.

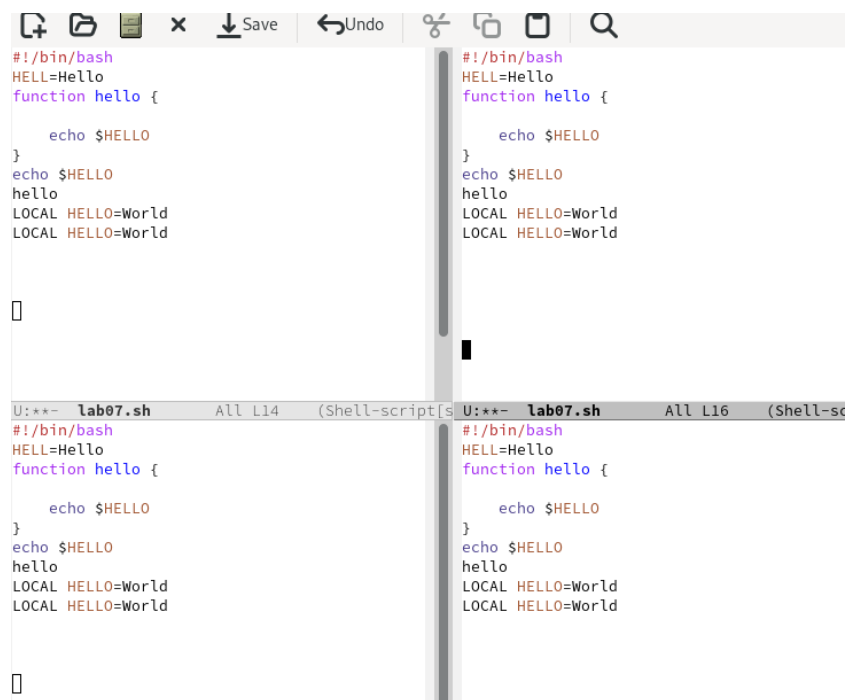


Рис. 2.17: Деление фрейма

- 15) В каждом из буферов я открыл новые файлы и написал там ФИО и номер группы.

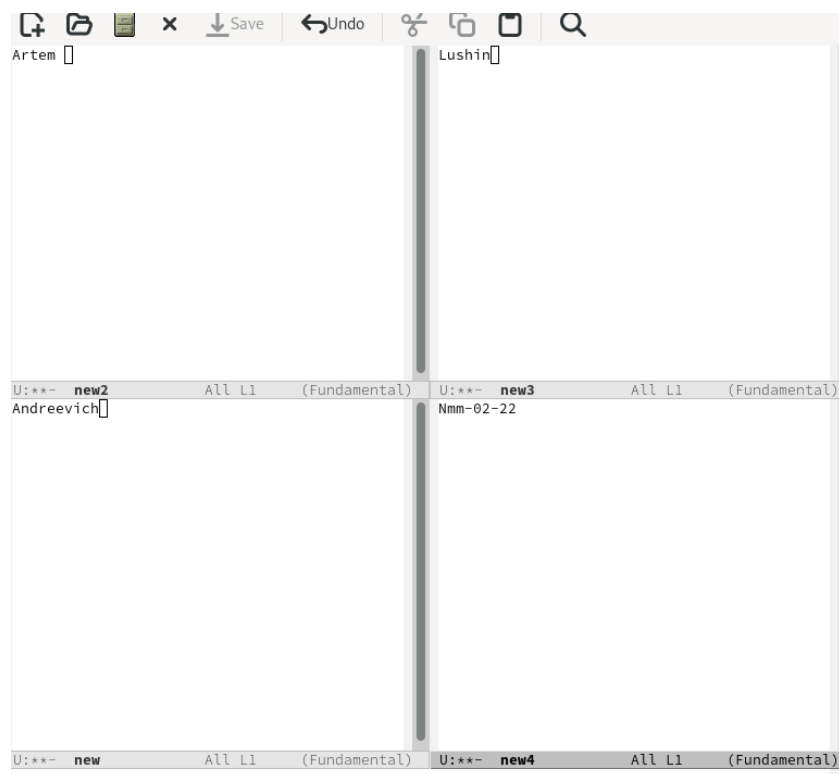
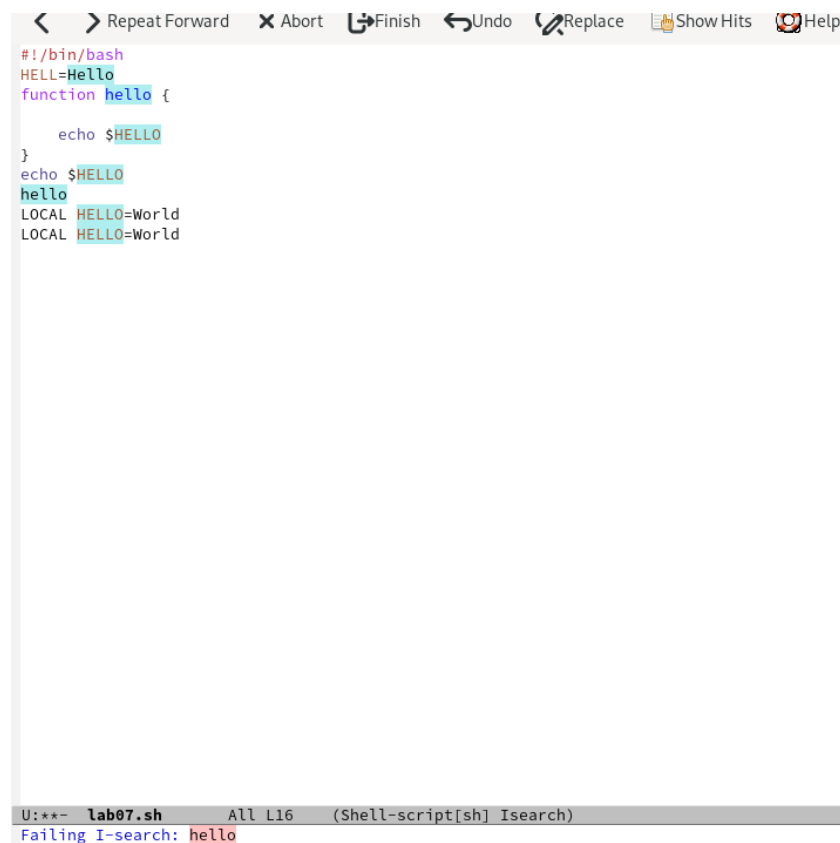


Рис. 2.18: Текст в 4 буферах

16) Я перешел в режим поиска и нашел слово HELLO.

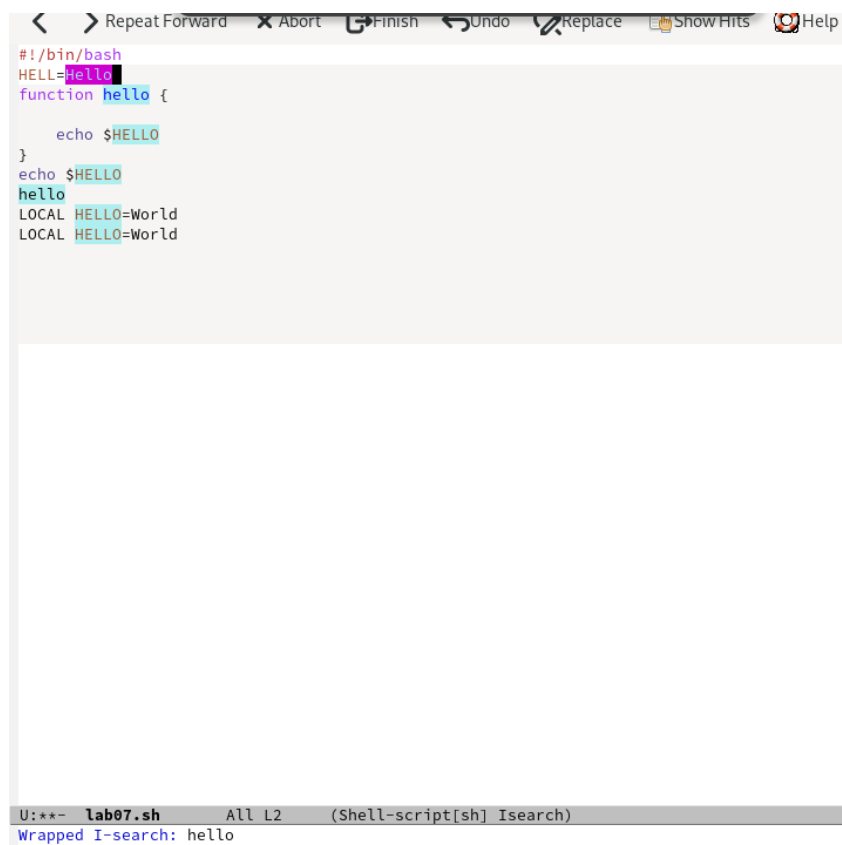


```
Repeat Forward X Abort Finish Undo Replace Show Hits Help
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
LOCAL HELLO=World

U:~ - lab07.sh All L16 (Shell-script[sh] Isearch)
Failing I-search: hello
```

Рис. 2.19: Режим поиска

- 17) Я попытался переключиться между результатами поиска, мне поочередно подсвечивало каждый из результатов.



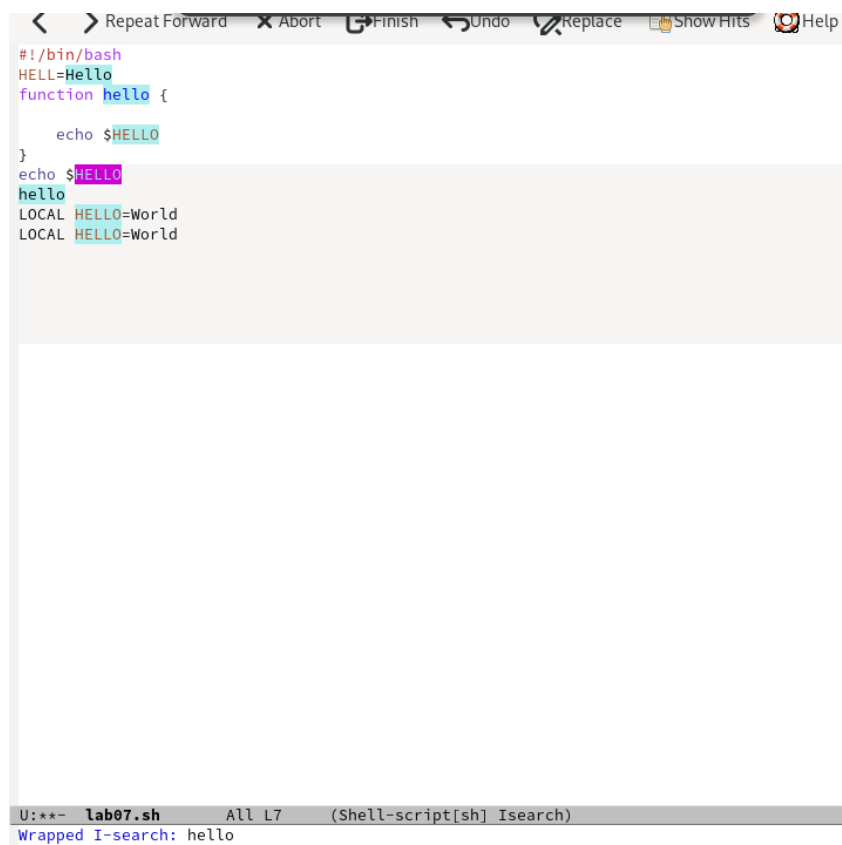
The image shows a code editor window with a search overlay. The code editor displays a shell script with the following content:

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

The search overlay is positioned at the bottom of the editor. It shows the search term "hello" and the results "LOCAL HELLO=World" and "LOCAL HELLO=World". The search results are displayed in a light blue background.

U:***- lab07.sh All L2 (Shell-script[sh] Isearch)
Wrapped I-search: hello

Рис. 2.20: Переключение между результатами



The image shows a code editor window with a search overlay. The code is a shell script snippet:

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

A search overlay is visible, showing the results of a search for the word "hello". The overlay is a light gray rectangle that covers the right side of the code editor. The search results are displayed in a list format, with the word "hello" highlighted in blue. The search results are as follows:

- Line 1: `#!/bin/bash`
- Line 2: `HELL=Hello`
- Line 3: `function hello {`
- Line 4: `echo $HELLO`
- Line 5: `}`
- Line 6: `echo $HELLO`
- Line 7: `hello`
- Line 8: `LOCAL HELLO=World`
- Line 9: `LOCAL HELLO=World`

The search overlay also shows the search term "hello" and the number of matches found, which is 1. The search results are displayed in a list format, with the word "hello" highlighted in blue. The search results are as follows:

- Line 1: `#!/bin/bash`
- Line 2: `HELL=Hello`
- Line 3: `function hello {`
- Line 4: `echo $HELLO`
- Line 5: `}`
- Line 6: `echo $HELLO`
- Line 7: `hello`
- Line 8: `LOCAL HELLO=World`
- Line 9: `LOCAL HELLO=World`

Рис. 2.21: Переключение между результатами

18) Я закрыл поиск.

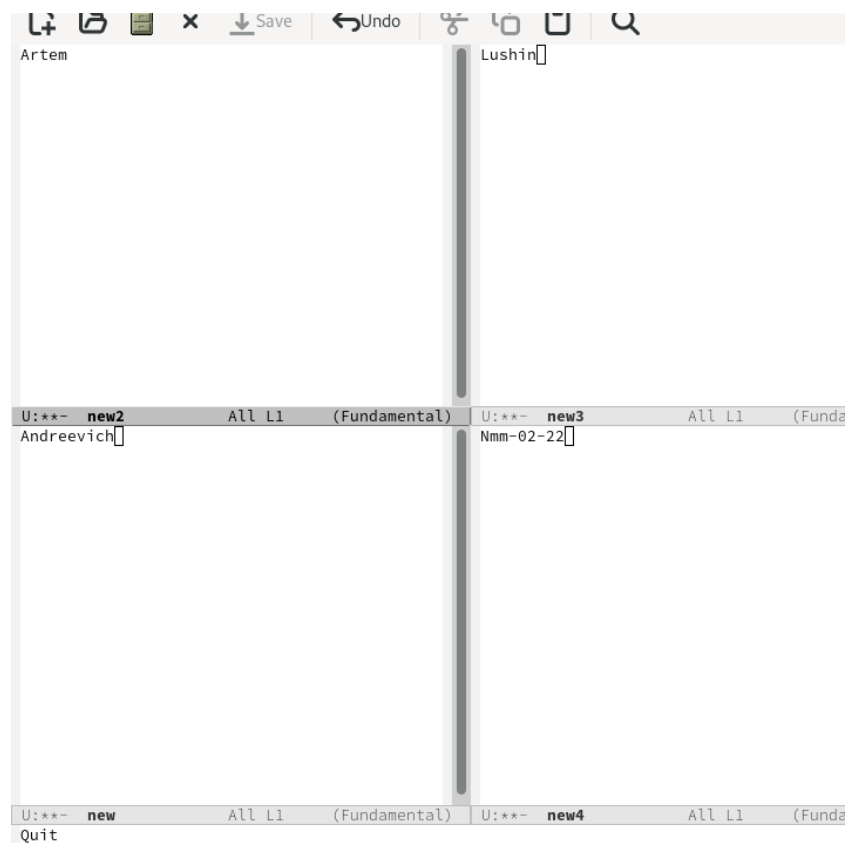


Рис. 2.22: Завершение поиска

19) Я перешел в режим поиска и замены и заменил слово HELLO на слово hi.



The screenshot shows a code editor window with a menu bar at the top containing icons for file operations (open, save, print, close) and editing (undo, redo, cut, copy, paste, find). The main text area contains a shell script:

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

Below the script, there is a search bar with the text "Query replace in region (default Hello → hi):". The status bar at the bottom of the editor displays "U:***- lab07.sh All L15 (Shell-script[sh])".

Рис. 2.23: Замена слова

20) Я испробовал другой режим поиска. Отличие этого режима от того, что я проверял до этого, это разный вывод результата. Если в первом случае, поиск показывал результаты прямо в тексте, то здесь он выводит их в отдельном окне.

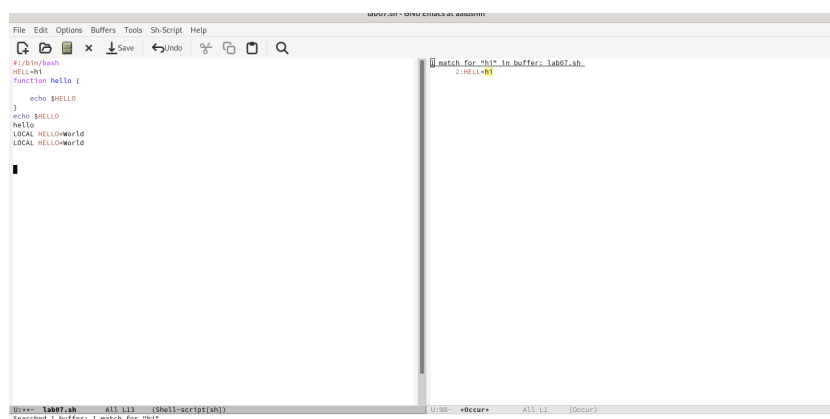


Рис. 2.24: Второй режим поиска

3 Выводы

Я познакомился с ос Linux. Приобрел практические навыки в работе с редактором Emacs.

4 Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs представляет собой мощный экранный редактор текста, написанный на языке высокого уровня Elisp.

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

Многие рутинные операции в Emacs удобнее производить с помощью клавиатуры, а не графического меню. Наиболее часто в командах Emacs используются сочетания с клавишами Ctrl и Meta (в обозначениях Emacs: C- и M-; клавиша Shift в Emacs обозначается как S-). Так как на клавиатуре для IBM PC совместимых ПК клавиши Meta нет, то вместо неё можно использовать Alt или Esc.

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

Если своими словами, то буфер - это файл, содержащий какой-либо текст. Окно же можно сказать область, где вы водится текст определенного буфера.

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

Можно открыть больше 10 буферов в одном окне.

5. Какие буферы создаются по умолчанию при запуске emacs?

Только что запущенный Emacs несет один буфер с именем 'scratch', который может быть использован для вычисления выражений Лиспа в Emacs.

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

Ctrl-c |(первые две нажму вместе, а третью отдельно), Ctrl-c Ctrl-|(каждую пару нажму раздельно).

7. Как поделить текущее окно на две части?

Разделить фрейм на два окна по вертикали (C-x 3),а по горизонтали (C-x 2)

8. В каком файле хранятся настройки редактора emacs?

В файле Emacs хранятся настройки редактора.

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

Кнопка BACKSPACE = функции C-k и ее можно переназначить.

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

Редактор Emacs мне показался удобнее, так как в нем больше возможностей по сравнению с vi.