

Отчет по лабораторной работе №11

Операционные системы

Зарицкая Марина Петровна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	20
6	Ответы на контрольные вопросы	21

Список иллюстраций

4.1	Открытие программы	9
4.2	Создание файла	10
4.3	Редактирование файла	10
4.4	Сохранение изменений в файле	10
4.5	Вырезание строки	10
4.6	Вставка строки в конце файла	11
4.7	Вырезанная область	11
4.8	Вставка в конец файла	12
4.9	Отмена последнего действия	12
4.10	Курсор в начале строки	12
4.11	Курсор в конце строки	13
4.12	Начало буфер	13
4.13	Конец буфера	14
4.14	Список активных буферов	14
4.15	Другое окно буфера	15
4.16	Закрытие буфера	15
4.17	Открытие другого буфера	16
4.18	Четыре окна	16
4.19	Открытие буферов в четырех окнах	17
4.20	Режим поиска	17
4.21	Перемещение по найденным выражениям	18
4.22	Выход из режима поиска	18
4.23	Замена слова	18
4.24	Слово заменено	19

Список таблиц

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 емас.
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;

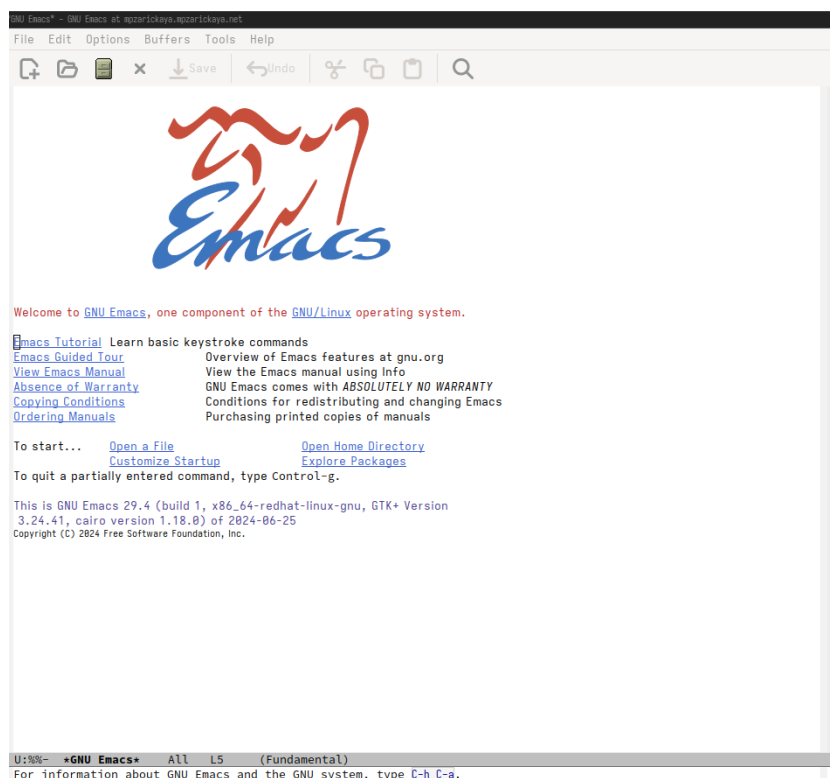
Всё это разнообразие достигается благодаря архитектуре Emacs, которая позволяет расширять возможности редактора при помощи языка Emacs Lisp. На языке C написаны лишь самые базовые и низкоуровневые части Emacs, включая полнофункциональный интерпретатор языка Lisp. Таким образом, Emacs имеет встроенный 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. 4.1).



Создаю файл ab07.sh с помощью комбинации Ctrl-x Ctrl-f (рис. 4.2).

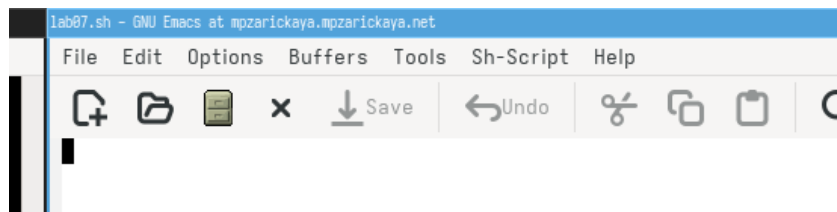


Рис. 4.2: Создание файла

Прописываю в файле текст программы (рис. 4.3).

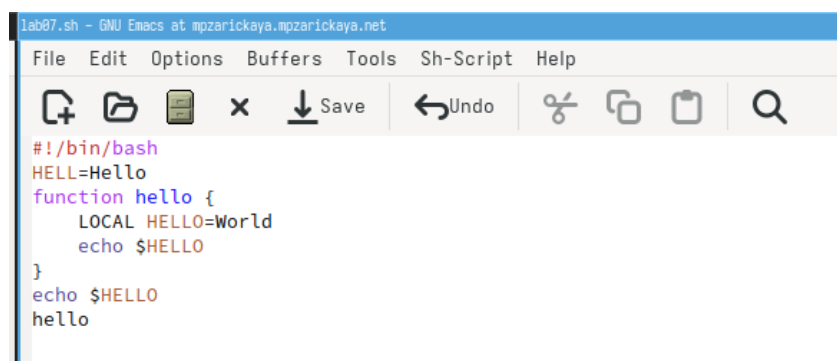


Рис. 4.3: Редактирование файла

Сохраняю файл с помощью комбинации C-x C-s (рис. 4.4).

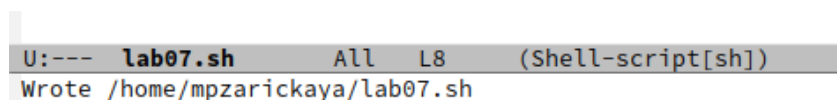


Рис. 4.4: Сохранение изменений в файле

Вырезаю одной командой целую строку (C-k) (рис. 4.5).

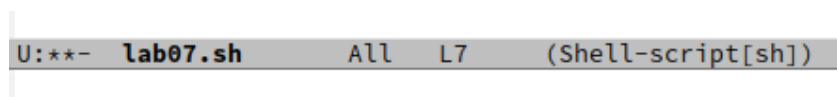


Рис. 4.5: Вырезание строки

Вставляю эту строку в конец файла (C-y) (рис. 4.6).

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

hello
echo $HELLO
```

Рис. 4.6: 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. 4.7).

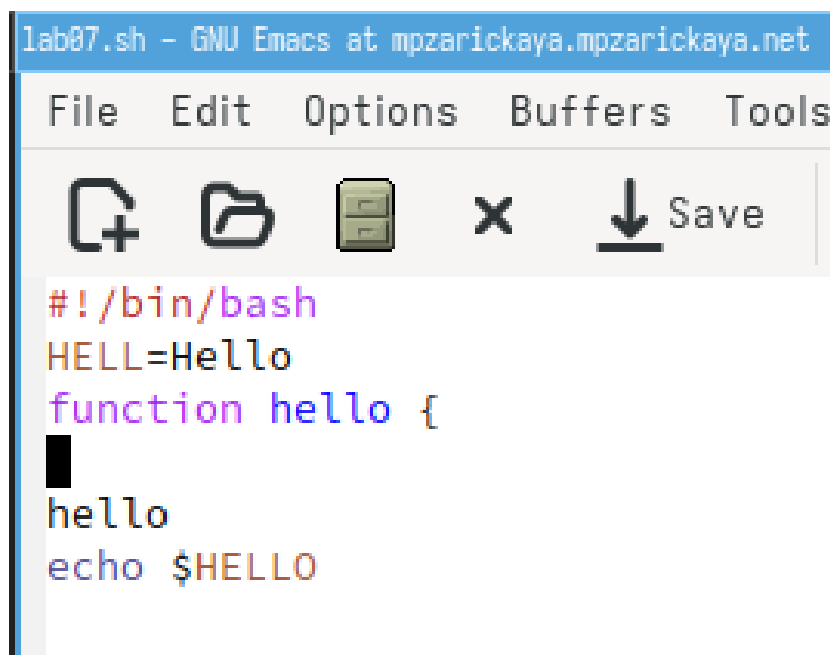
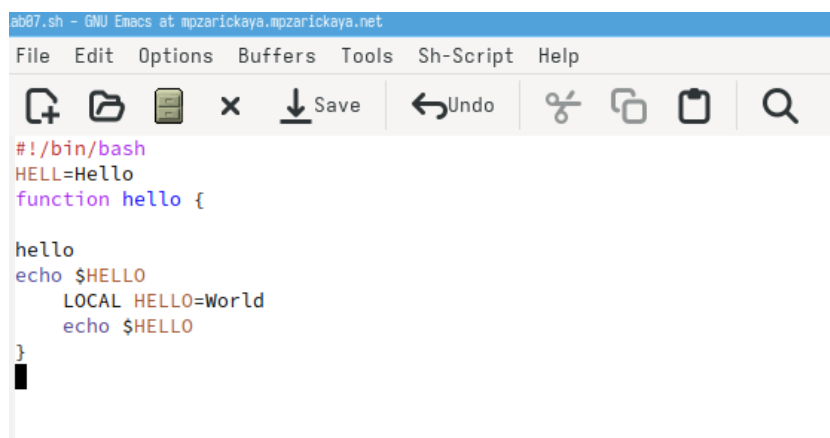


Рис. 4.7: Вырезанная область

Вставляю область в конец файла (C-y)(рис. 4.8).



```
ab07.sh - GNU Emacs at mpzarickaya.mpzarickaya.net
File Edit Options Buffers Tools Sh-Script Help
[Icons] Save Undo [Icons]
#!/bin/bash
HELL=Hello
function hello {
hello
echo $HELLO
  LOCAL HELLO=World
  echo $HELLO
}
```

Рис. 4.8: Вставка в конец файла

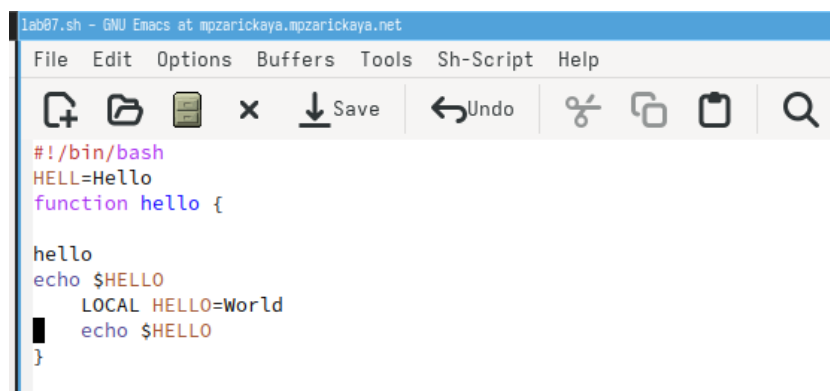
Отменяю последнее действие C-/ (рис. 4.9).



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

Рис. 4.9: Отмена последнего действия

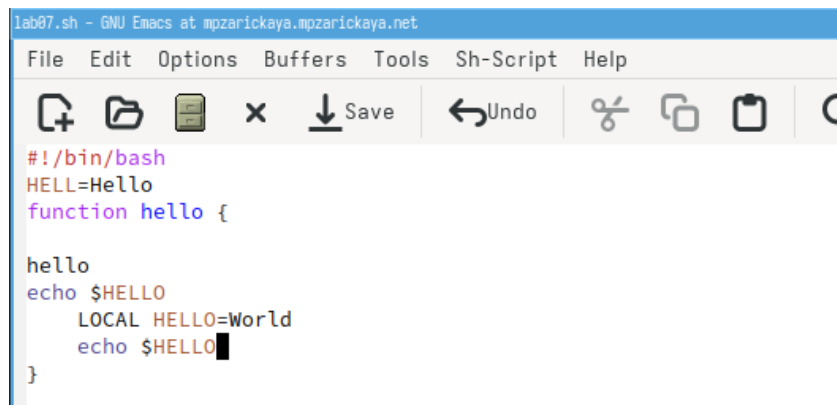
Перевожу курсор в начало строки C-a (рис. 4.10).



```
lab07.sh - GNU Emacs at mpzarickaya.mpzarickaya.net
File Edit Options Buffers Tools Sh-Script Help
[Icons] Save Undo [Icons]
#!/bin/bash
HELL=Hello
function hello {
hello
echo $HELLO
  LOCAL HELLO=World
  echo $HELLO
}
```

Рис. 4.10: Курсор в начале строки

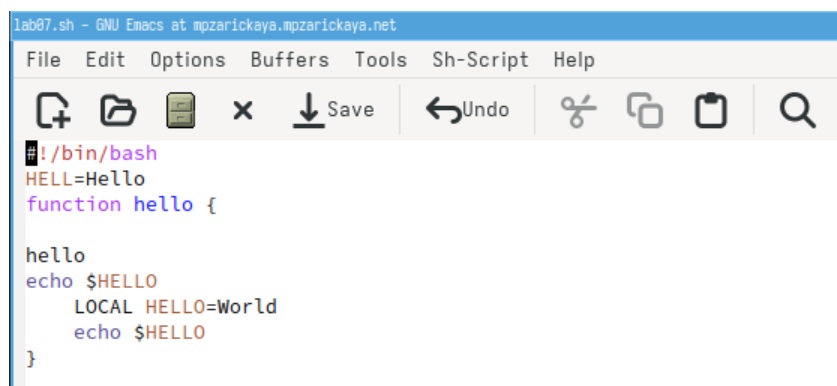
Перемещаю курсор в конец строки C-e (рис. 4.11).



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

Рис. 4.11: Курсор в конце строки

Перемещаю курсор в начало файла M-< (рис. 4.12).



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

Рис. 4.12: Начало буфер

Перемещаю курсор в конец файла M->(рис. 4.13).

```
lab07.sh - GNU Emacs at mpzarickaya.mpzarickaya.net
File Edit Options Buffers Tools Sh-Script Help
[Icons: Copy, Paste, Save, Undo, Redo, Find]
#!/bin/bash
HELL=Hello
function hello {

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

Рис. 4.13: Конец буфера

Открываю список активных буферов в другом окне C-x C-b (рис. 4.14).

```
U:*** lab07.sh All L9 (Shell-script[sh])
CRM Buffer Size Mode File
[ ] * lab07.sh 98 Shell-script[sh] ~/lab07.sh
% *info* 2837704 Info (emacs) Top
% *GNU Emacs* 718 Fundamental
% *scratch* 145 Lisp Interaction
%* *Messages* 2825 Messages
%* *Async-native-compile-log* 165 Fundamental

U:%%- *Buffer List* All L1 (Buffer Menu)
C-x C-b
```

Рис. 4.14: Список активных буферов

Переключаюсь на другой буфер C-x o (рис. 4.15).

```
U:*** Lab07.sh All L9 (Shell-script[sh])
scroll-down-command: Beginning of buffer [5 times]
xref-go-back: At start of xref history
M-o is undefined
M-cup> is undefined
M-cdown> is undefined
M-cup> is undefined
M-cdown> is undefined [2 times]
C-y is undefined
C-w is undefined
C-c C-b is undefined
next-line: End of buffer
funcall-interactively: End of buffer

U:*** *Messages* Bot L73 (Messages)
```

Рис. 4.15: Другое окно буфера

Закрываю окно другого буфера C-x 0 (рис. 4.16).

```
lab07.sh - GNU Emacs at mporickaya.mporickaya.net
File Edit Options Buffers Tools Sh-Script Help
[Icons] Save Undo [Icons]
#!/bin/bash
HELL=Hello
function hello
{
  hello
  echo $HELL
  LOCAL HELLO=World
  echo $HELL
}

U:*** Lab07.sh All L9 (Shell-script[sh])
C-x 0
```

Рис. 4.16: Заккрытие буфера

Открываю другой буфер без вывода их списка на экран с помощью C-x b(рис. 4.17).

```
U:***- lab07.sh All L9 (Shell-script[sh])
Switch to buffer (default *Messages*):
```

Рис. 4.17: Открытие другого буфера

Делю фрейм на 4 части: сначала на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2) (рис. 4.18).

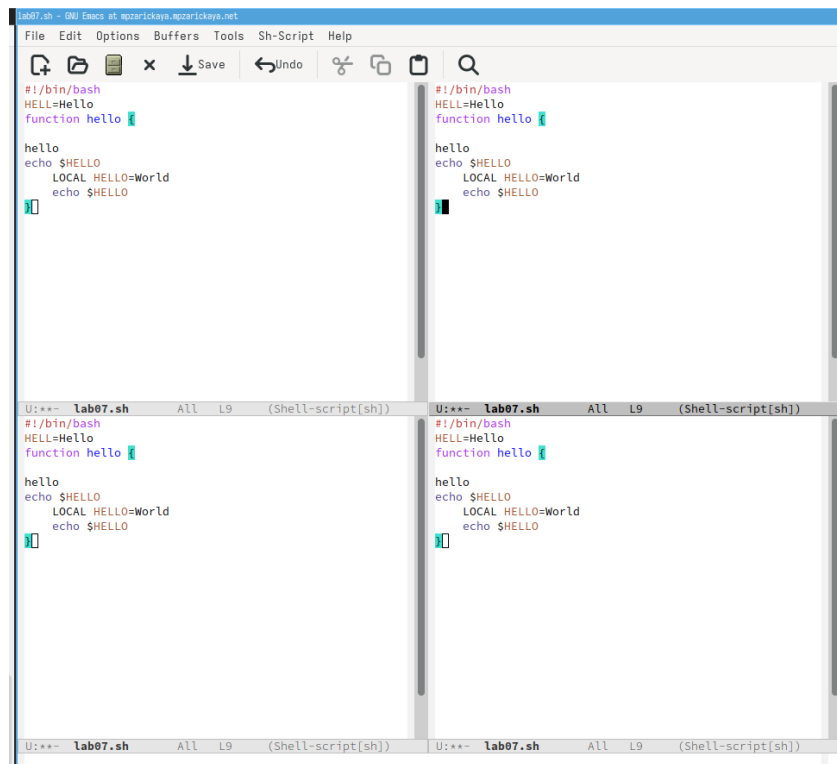


Рис. 4.18: Четыре окна

В каждом из четырех созданных окон с помощью С-х b открыла разные буферы и (рис. 4.19).

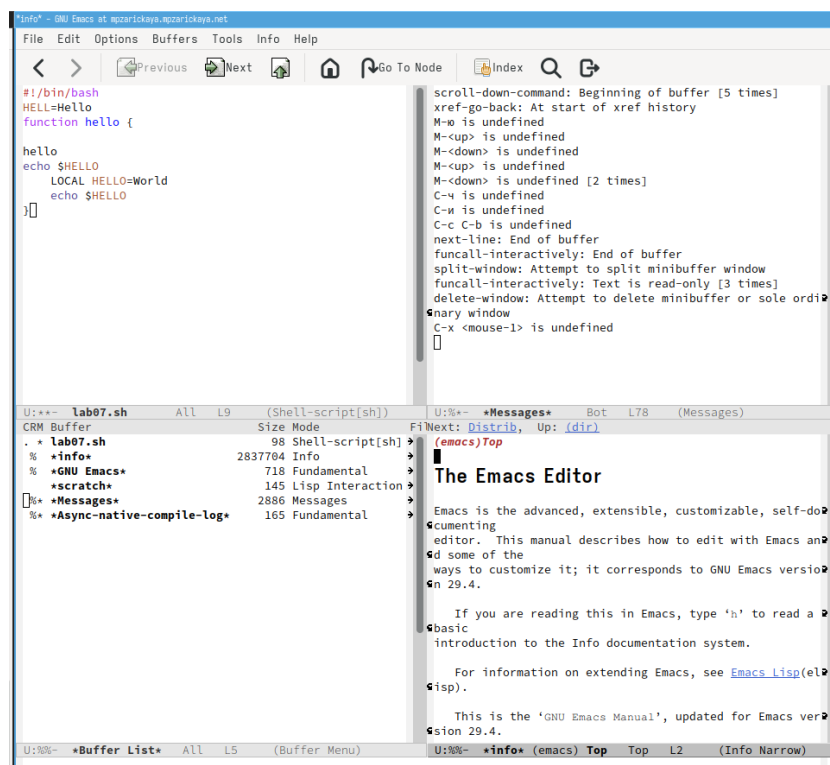


Рис. 4.19: Открытие буферов в четырех окнах

Перехожу в режим поиска с помощью C-s, ищу слова в тексте, они подсвечиваются (рис. 4.20).

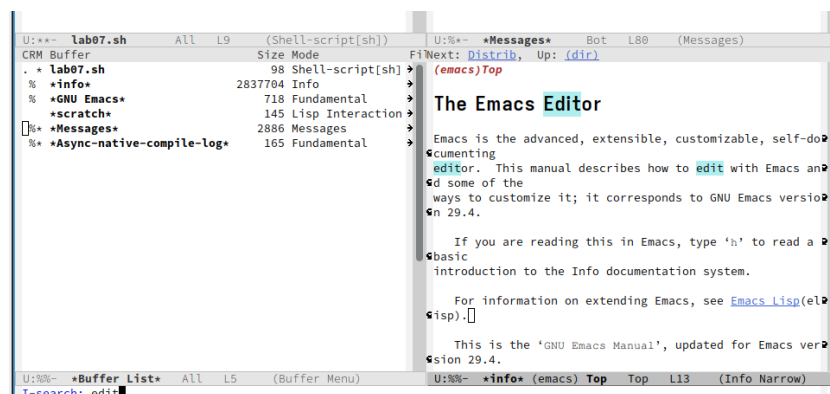


Рис. 4.20: Режим поиска

С помощью той же комбинации C-s я могу перемещаться по результатам поиска (рис. 4.21).

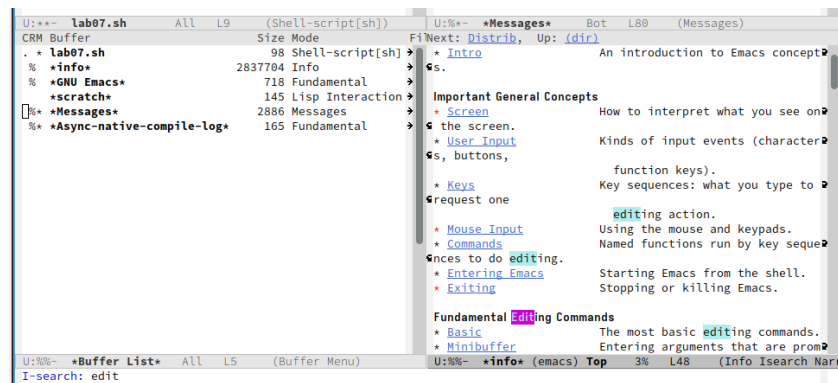


Рис. 4.21: Перемещение по найденным выражениям

С помощью C-g выхожу из режима поиска, снимается выделение (рис. 4.22).



Рис. 4.22: 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. 4.23).

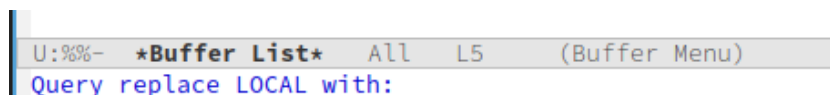
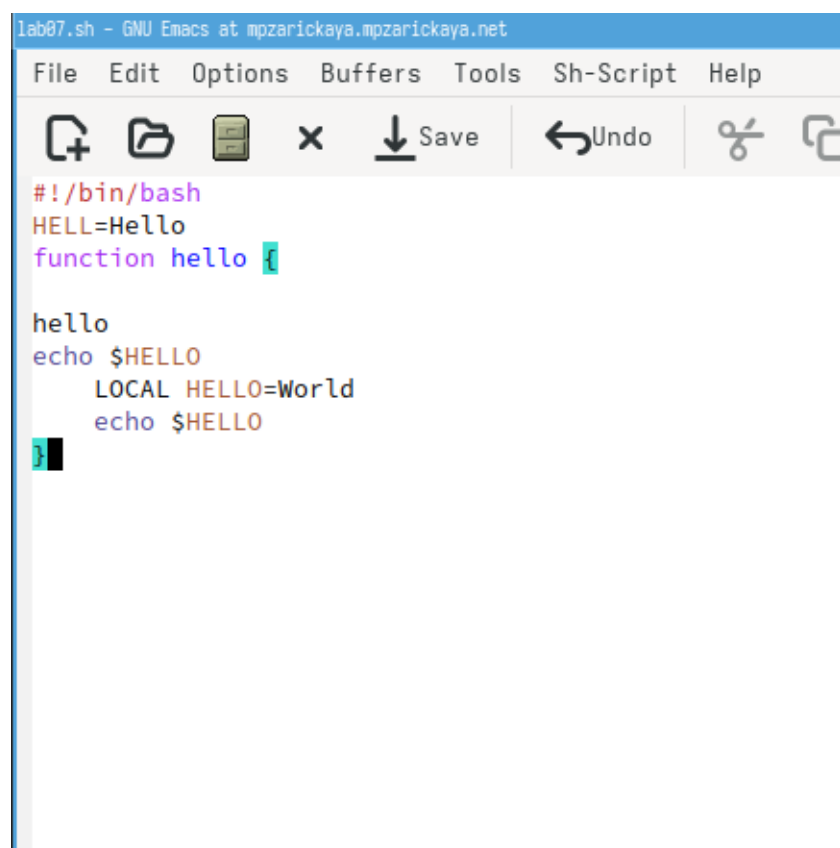


Рис. 4.23: Замена слова

Видим, что слова были заменены успешно (рис. 4.24).



```
lab07.sh - GNU Emacs at mpzarickaya.mpzarickaya.net
File Edit Options Buffers Tools Sh-Script Help
[Icons: Open, Save, Undo, etc.]
#!/bin/bash
HELL=Hello
function hello {
hello
echo $HELLO
    LOCAL HELLO=World
echo $HELLO
}
```

Рис. 4.24: 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'а.

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 особым 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?

C 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.