

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 10

дисциплина: Операционные системы

Студент: Губина Ольга Вячеславовна Группа: НПИбд-01-20

Преподаватель: Велиева Татьяна Рефатовна

МОСКВА

2021 г.

Цель работы:

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

Теоретические сведения:

В данной лабораторной работе нам предстоит познакомиться с текстовым редактором emacs. Для этого нам сначала нужно ознакомиться с *информацией*^[1], посвященной этой теме.

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

Основные термины Emacs

Буфер — объект, представляющий какой-либо текст.

Буфер может содержать что угодно, например, результаты компиляции программы или встроенные подсказки. Практически всё взаимодействие с пользователем, в том числе интерактивное, происходит посредством буферов.

Фрейм соответствует окну в обычном понимании этого слова. Каждый фрейм содержит область вывода и одно или несколько окон Emacs.

Окно — прямоугольная область фрейма, отображающая один из буферов.

Каждое окно имеет свою *строку состояния*, в которой выводится следующая информация: - название буфера; - его основной режим; - изменялся ли текст буфера - как далеко вниз по буферу расположен курсор.

Каждый буфер находится только в одном из возможных основных режимов. Существующие основные режимы включают режим Fundamental (наименее специализированный), режим Text, режим Lisp, режим C, режим Texinfo и другие. Под второстепенными режимами понимается список режимов, которые включены в данный момент в буфере выбранного окна.

Область вывода — одна или несколько строк внизу фрейма, в которой Emacs выводит различные сообщения, а также запрашивает подтверждения и дополнительную информацию от пользователя.

Минибуфер используется для ввода дополнительной информации и всегда отображается в области вывода.

Точка вставки — место вставки (удаления) данных в буфере.

Основы работы в Emacs

Для запуска Emacs необходимо в командной строке набрать emacs (или emacs & для работы в фоновом режиме относительно консоли).

Для работы с Emacs можно использовать как элементы меню, так и различные сочетания клавиш. Например, для выхода из Emacs можно воспользоваться меню File и выбрать пункт Quit , а можно нажать последовательно Ctrl-х Ctrl-с (в обозначениях Emacs: C-x C-c).

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 .

Для доступа к системе меню используйте клавишу F10 .

Клавиши Ctrl , Meta и Shift принято называть префиксными. Например, запись M-x означает, что надо удерживая клавишу Meta (или Alt), нажать на клавишу x. Для открытия файла следует использовать команду C-x C-f (надо, удерживая клавишу Ctrl , нажать на клавишу x , затем отпустить обе клавиши и снова, удерживая клавишу Ctrl , нажать на клавишу f).

По назначению префиксные сочетания клавиш различаются следующим образом:

- C-x — префикс ввода основных команд редактора (например, открытия, закрытия, сохранения файла и т.д.);
- C-c — префикс вызова функций, зависящих от используемого режима.

Режим — пакет расширений, изменяющий поведение буфера Emacs при редактировании и просмотре текста (например, для редактирования исходного текста программ на языках C или Perl).

Подробнее с командами и горячими клавишами редактора emacs можно ознакомиться в *справочнике команд GNU Emacs*^[2].

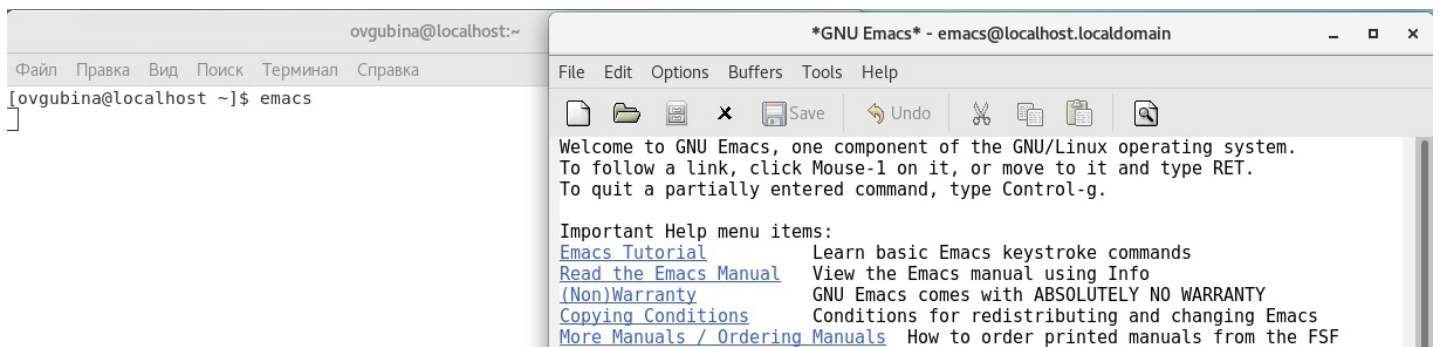
Задачи:

1. Научиться продельывать с текстом стандартные процедуры редактирования;
2. Научиться использовать команды по перемещению курсора;
3. Научиться управлять буферами, окнами и режимом поиска.

Выполнение работы:

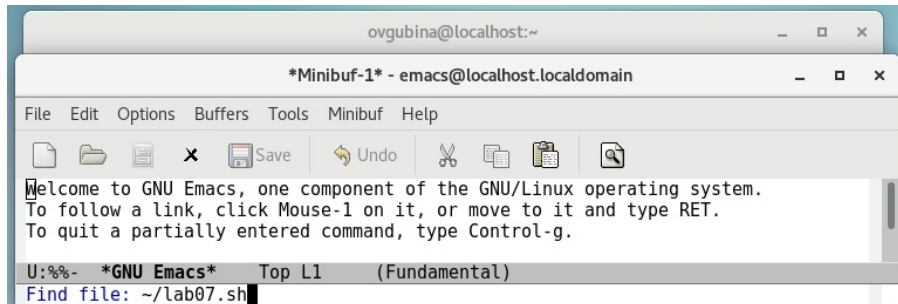
1. Ознакомиась с теоретическим материалом и редактором emacs, после чего открываю сам редактор, вводя в командную строку emacs (*рисунок 1*), видим, что в отдельном окне открывается редактор (*рисунок 1*).

Рисунок 1: открываем редактор emacs:



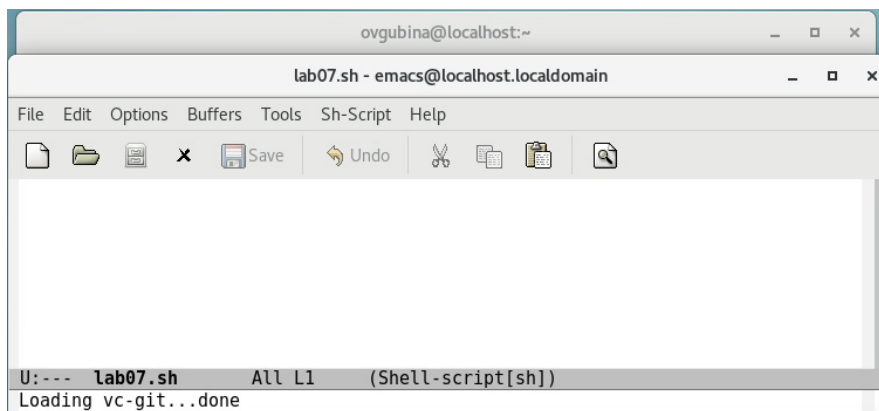
2. Создадим файл lab07.sh с помощью комбинации `Ctrl-x Ctrl-f` (C-x C-f) (рисунки 2). Появляется фрейм, в котором нам предлагают ввести имя редактируемого файла (поскольку такого файла изначально не было, он создается)

Рисунок 2: создание файла lab07.sh:



Видим, что наш новый файл открылся (рисунки 3).

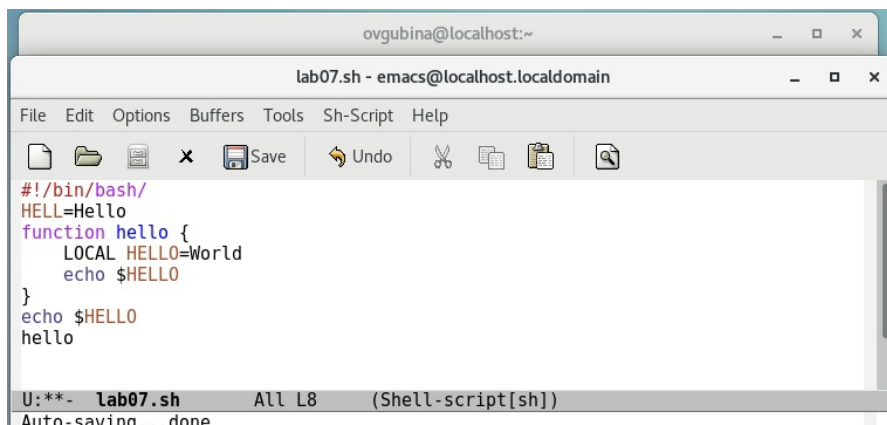
Рисунок 3: файл lab07.sh:



3. Наберем предложенный текст (рисунки 4):

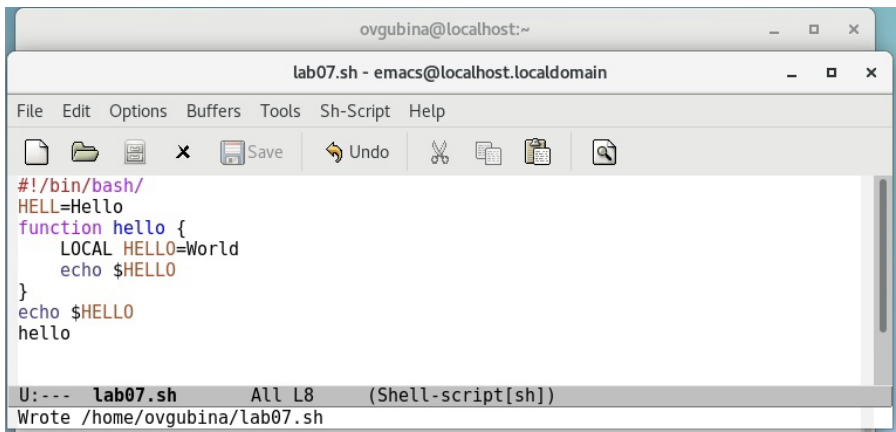
```
#!/bin/bash
HELL=Hello
function hello {
  LOCAL_HELL=World
  echo $HELL
}
echo $HELL
hello
```

Рисунок 4: набор текста:



4. Сохраним файл с помощью комбинации `Ctrl-x Ctrl-s` (C-x C-s) (рисунки 5).

Рисунок 5: сохранение файла:



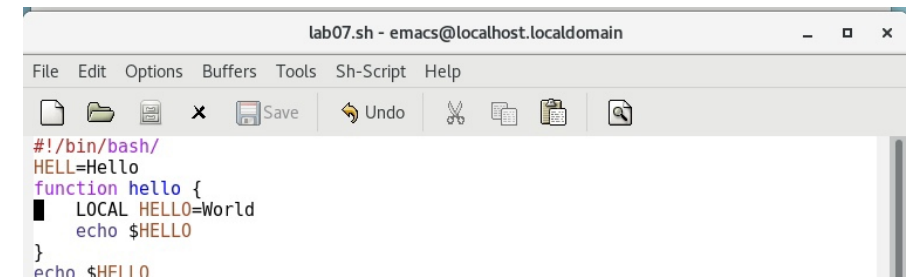
```
#!/bin/bash/
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

U: --- lab07.sh All L8 (Shell-script[sh])
Wrote /home/ovgubina/lab07.sh

5. Проделаем с текстом стандартные процедуры редактирования, каждое действие будет осуществляться комбинацией клавиш.

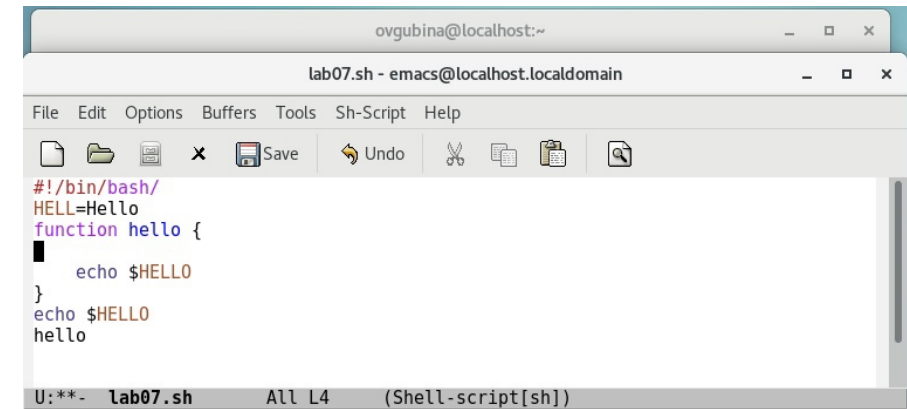
5.1. Вырезаем командой `C - k` целую строку, предварительно установив курсор на ее начало (рисунк 6-7).

Рисунок 6: устанавливаем курсор на начало 4ой строки, ее мы хотим вырезать:



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

Рисунок 7: вырезали строку:

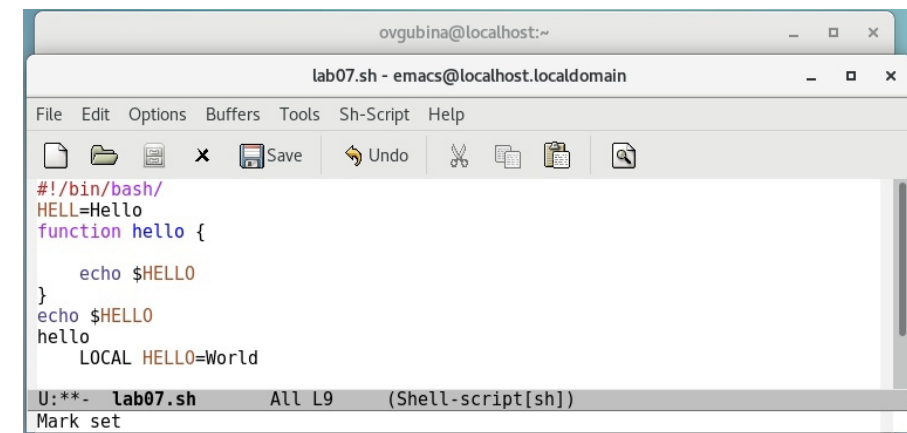


```
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
```

U: *- lab07.sh All L4 (Shell-script[sh])

5.2. Теперь вставим эту строку в конец файла командой `C - y` (рисунк 8). Предварительно установили курсор в конец буфера - `Esc - >`.

Рисунок 8: вставка строки:

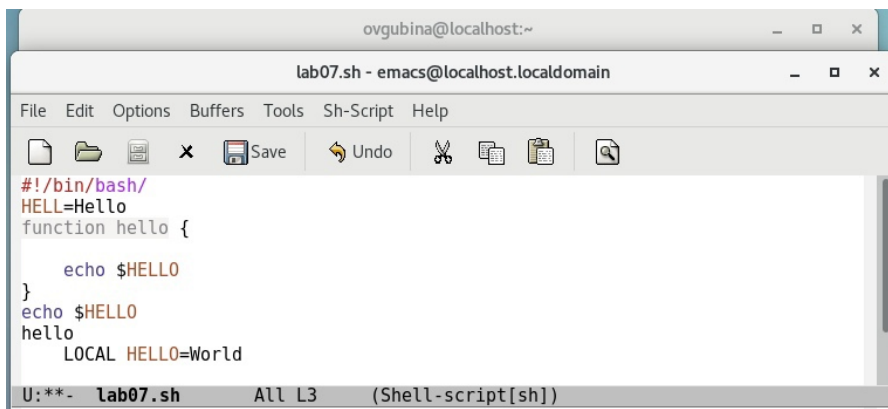


```
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
    LOCAL HELLO=World
```

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

5.3. Выделим область текста при помощи команды `C - s p a c e` (выделенный текст подсвечен серым) (рисунк 9). Для этого сначала устанавливаем курсор на начало нужного участка и двигаем конец выделения клавишами стрелок.

Рисунок 9: выделение текста:

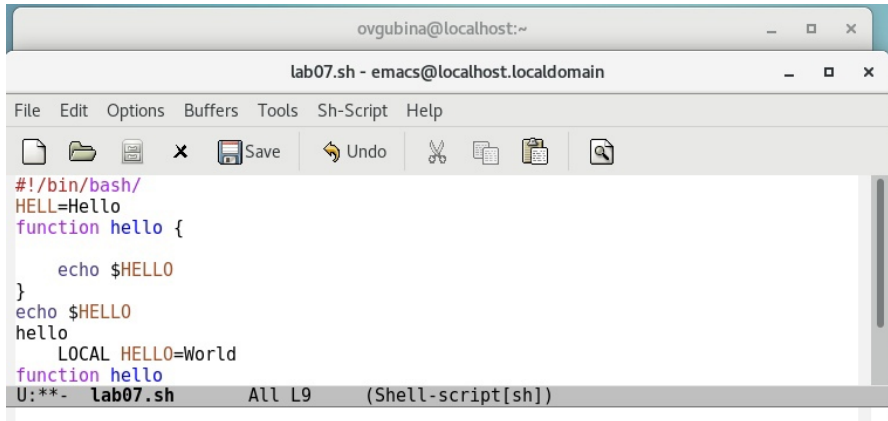


```
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
U:***- lab07.sh All L3 (Shell-script[sh])
```

5.4. Копируем эту область текста в буфер обмена командой `Alt - w`.

5.5. Вставив область в конец файла командой `C - y` (рисунки 10). Предварительно установили курсор в конец буфера - `Esc - >`.

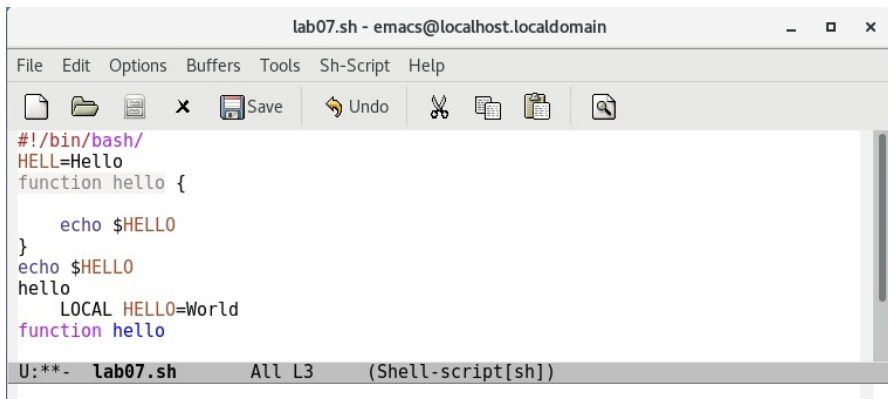
Рисунок 10: вставка выделенного текста:



```
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
U:***- lab07.sh All L9 (Shell-script[sh])
```

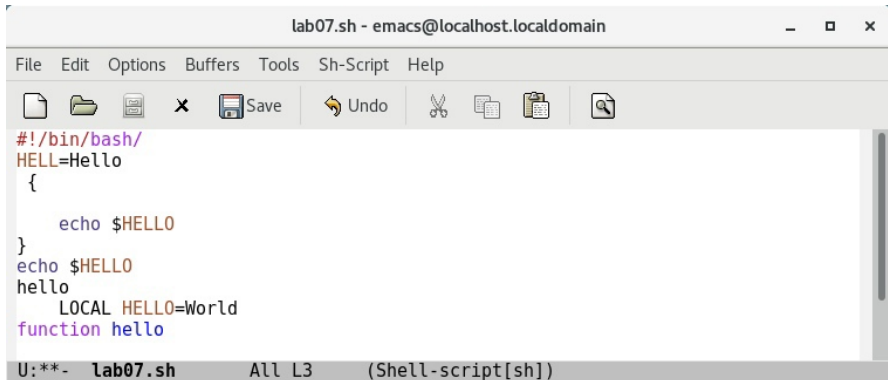
5.6. Вновь выделим ту же область командой `C - s` (рисунки 11) и на этот раз вырежем её командой `C - w` (рисунки 12).

Рисунок 11: выделение текста:



```
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
U:***- lab07.sh All L3 (Shell-script[sh])
```

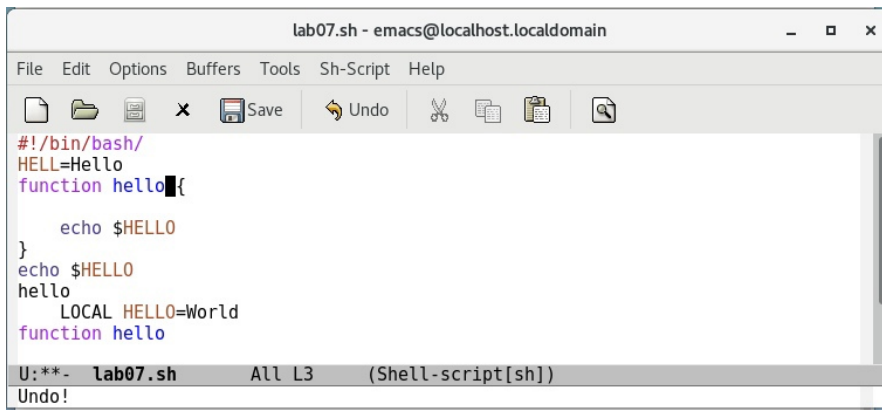
Рисунок 12: вырезка текста:



```
#!/bin/bash/
HELL=Hello
{
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello
U:***- lab07.sh All L3 (Shell-script[sh])
```

5.7. Отменим последнее действие командой `C - /` (рисунки 13), видим, что вырезанный текст вернулся на свое место.

Рисунок 13: отмена последнего изменения:

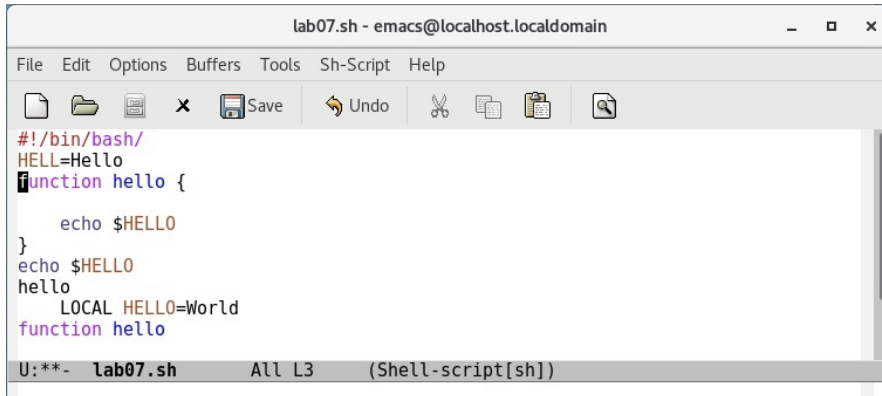


```
lab07.sh - emacs@localhost.localdomain
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash/
HELL=Hello
function hello{
    echo $HELLO
}
echo $HELLO
hello
    LOCAL HELLO=World
function hello
U:***- lab07.sh All L3 (Shell-script[sh])
Undo!
```

6. Теперь научимся использовать команды по перемещению курсора.

6.1. Переместим курсор в начало строки - C-a (рисунок 14). Изначальное положение курсора было как на рисунке 13.

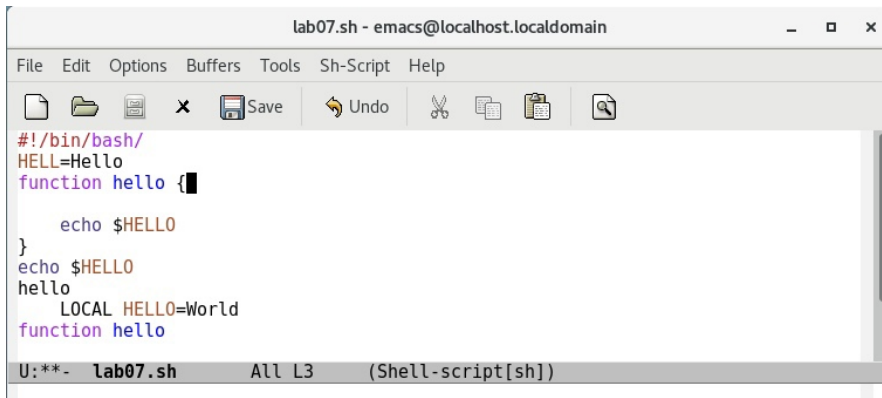
Рисунок 14: перемещение курсора в начало строки:



```
lab07.sh - emacs@localhost.localdomain
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
    LOCAL HELLO=World
function hello
U:***- lab07.sh All L3 (Shell-script[sh])
```

6.2. Переместим курсор в конец строки - C-e (рисунок 15).

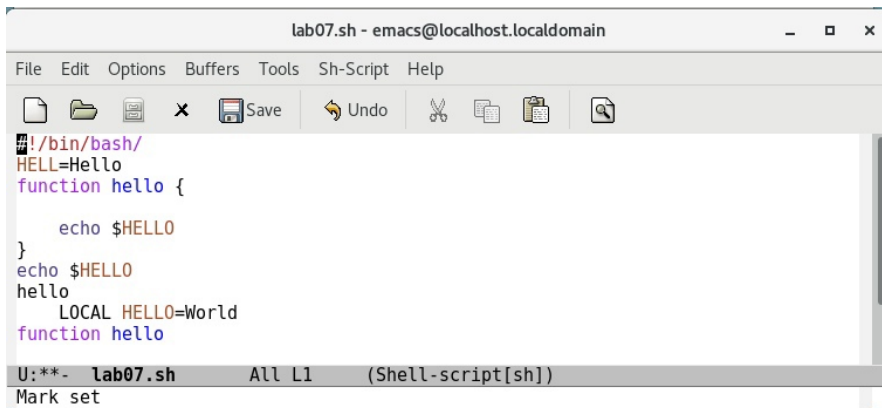
Рисунок 15: перемещение курсора в конец строки:



```
lab07.sh - emacs@localhost.localdomain
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
    LOCAL HELLO=World
function hello
U:***- lab07.sh All L3 (Shell-script[sh])
```

6.3. Переместим курсор в начало буфера - Esc - < (рисунок 16).

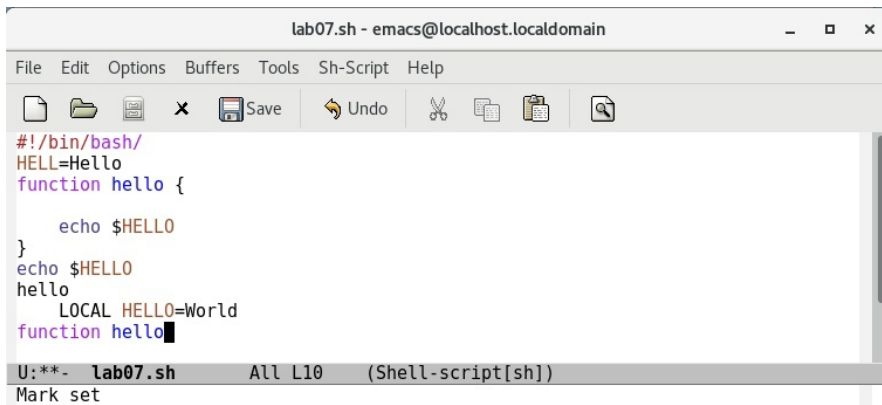
Рисунок 16: перемещение курсора в начало буфера:



```
lab07.sh - emacs@localhost.localdomain
File Edit Options Buffers Tools Sh-Script Help
Save Undo
#!/bin/bash/
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
    LOCAL HELLO=World
function hello
U:***- lab07.sh All L1 (Shell-script[sh])
Mark set
```

6.4. Переместим курсор в конец буфера - Esc - > (рисунок 17).

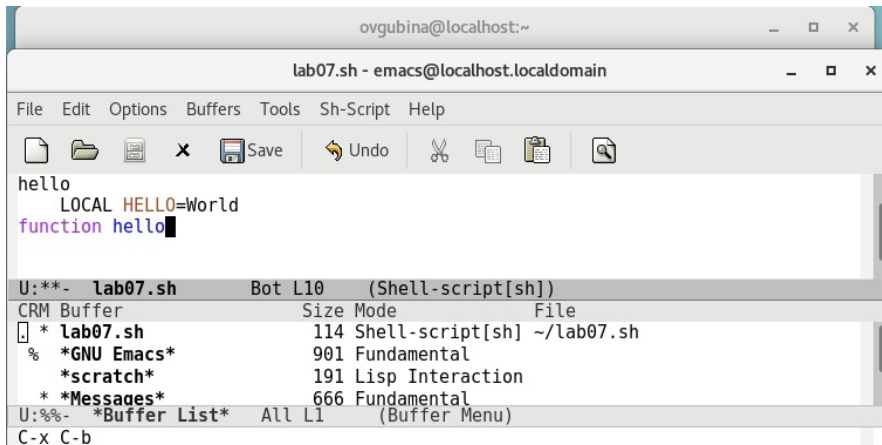
Рисунок 17: перемещение курсора в конец буфера:



7. Управление буферами.

7.1. Выводим список активных буферов на экран командой C-x C-b (рисунк 18).

Рисунок 18: вывод списка активных буферов:



7.2. Переместимся во вновь открытое окно со списком открытых буферов командой C-x o (рисунк 19) и переключимся на другой буфер командой C-x b. Высветивается фрейм, в который нам нужно ввести название буфера, на который мы хотим переключиться (рисунк 20).

Рисунок 19: перемещение в окно со списком открытых буферов:

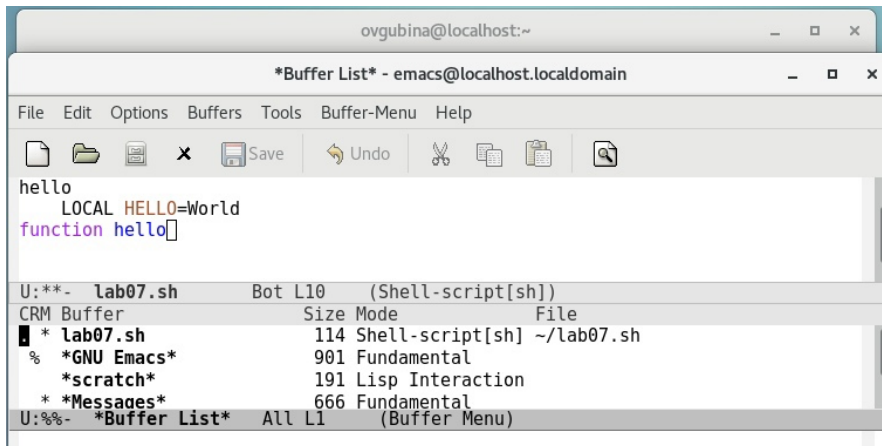
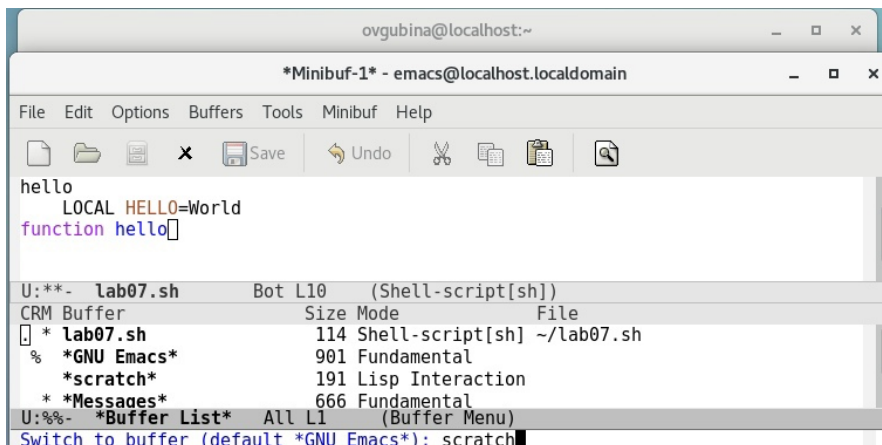
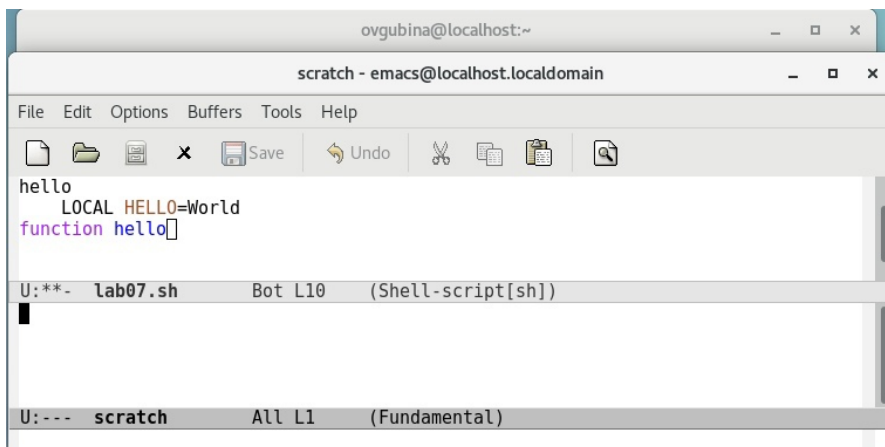


Рисунок 20: переключение на буфер scratch:



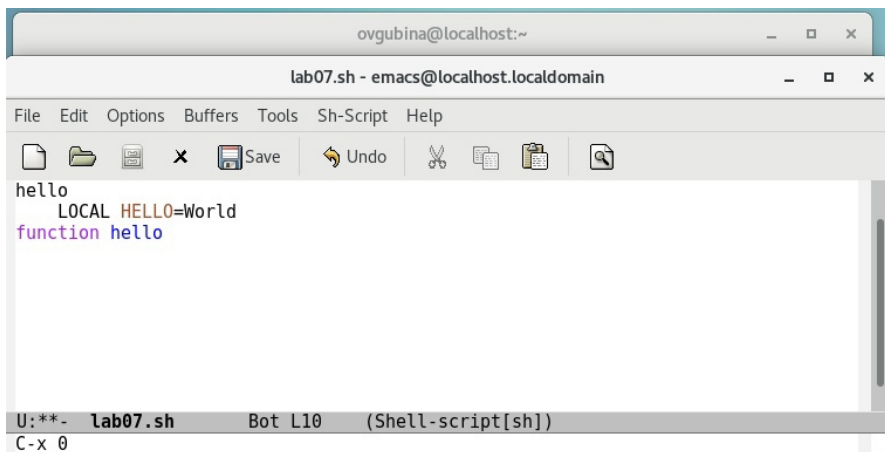
Видим, что действительно переключились на буфер scratch (рисунк 21).

Рисунок 21: переключение на буфер scratch (2):



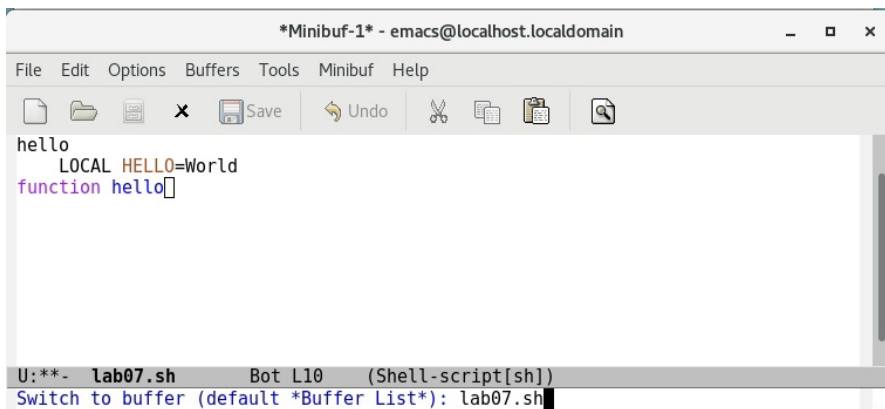
7.3. Закроем это окно - C-x 0 (рисунок 22).

Рисунок 22: закрыли окно со списком буферов:



7.4. Теперь вновь переключимся между буферами, но уже без вывода их списка на экран - C-x b (рисунок 23).

Рисунок 23: переключение на буфер lab07.sh:



8. Управление окнами.

8.1. Поделим фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3) (рисунок 24), а затем каждое из этих окон на две части по горизонтали (C-x 2) (рисунок 25).

Рисунок 24: деление окна по вертикали:

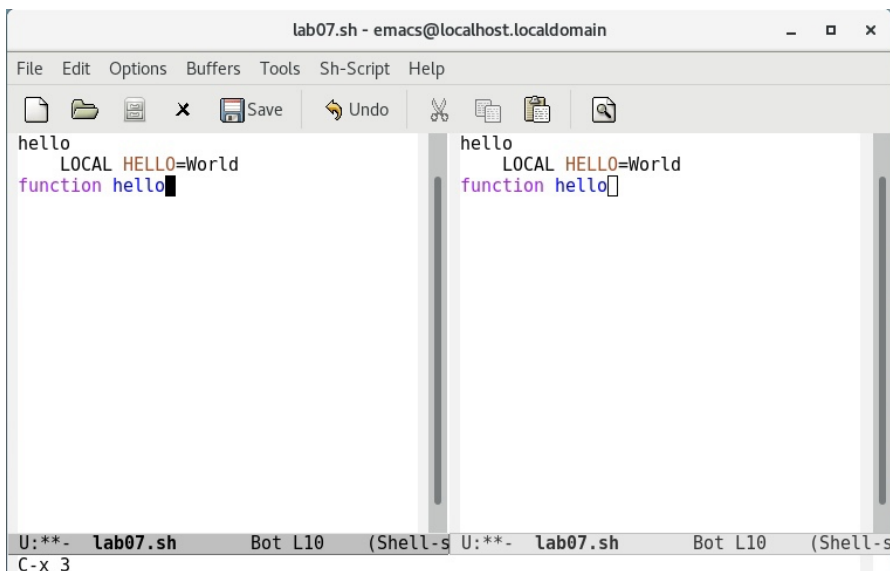
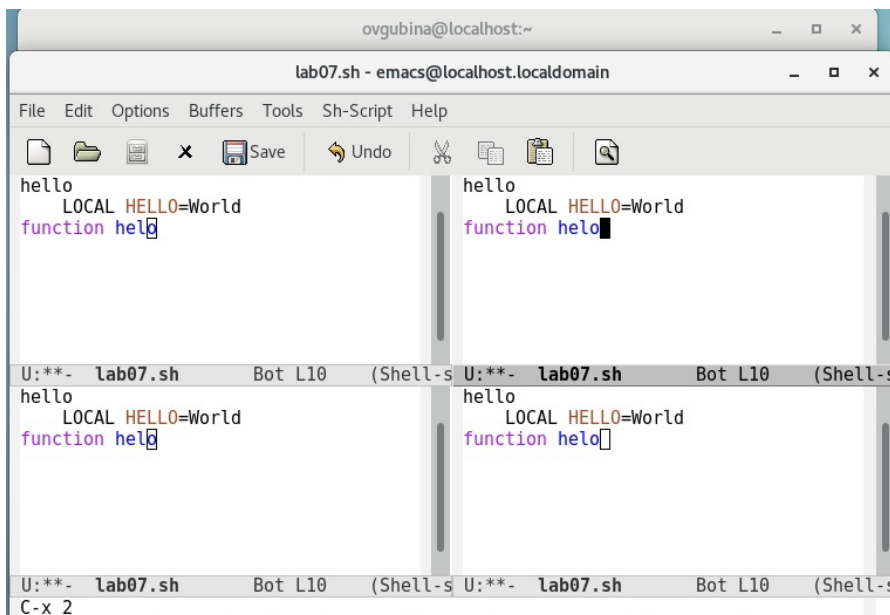


Рисунок 25: деление по горизонтали:



8.2. В каждом из четырёх созданных окон откроем новый буфер (файл) new_file.txt командой `C-x C-f` (рисунок 26-28) (между окнами переключаемся командой `C-x 0`) и введем несколько строк текста (рисунок 29).

Рисунок 26: открытие файла в одном окне:

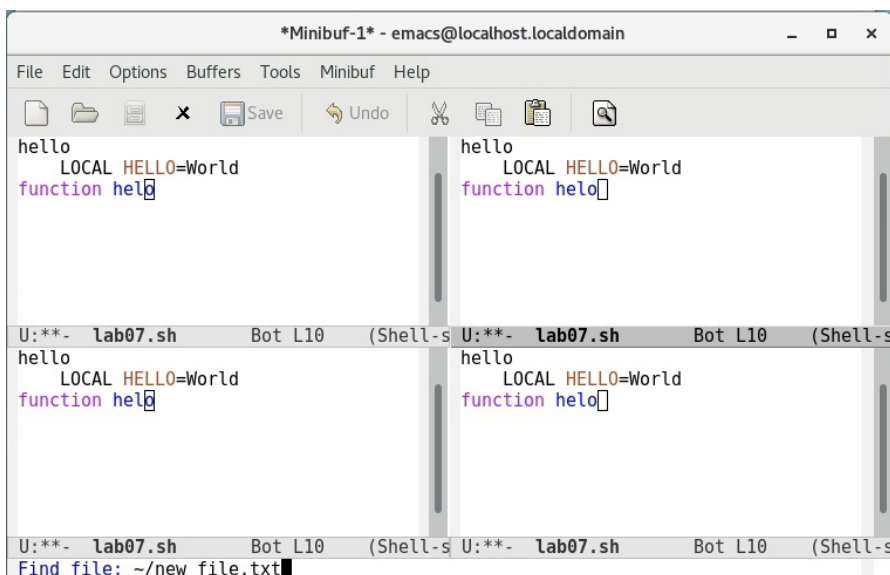


Рисунок 27: открытие файла в одном окне (2):

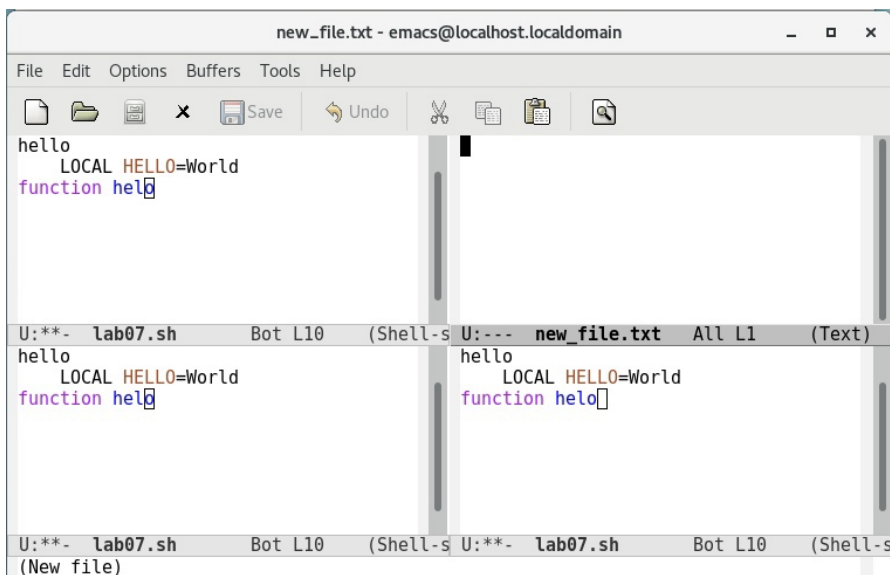


Рисунок 28: открытие файла во всех окнах:

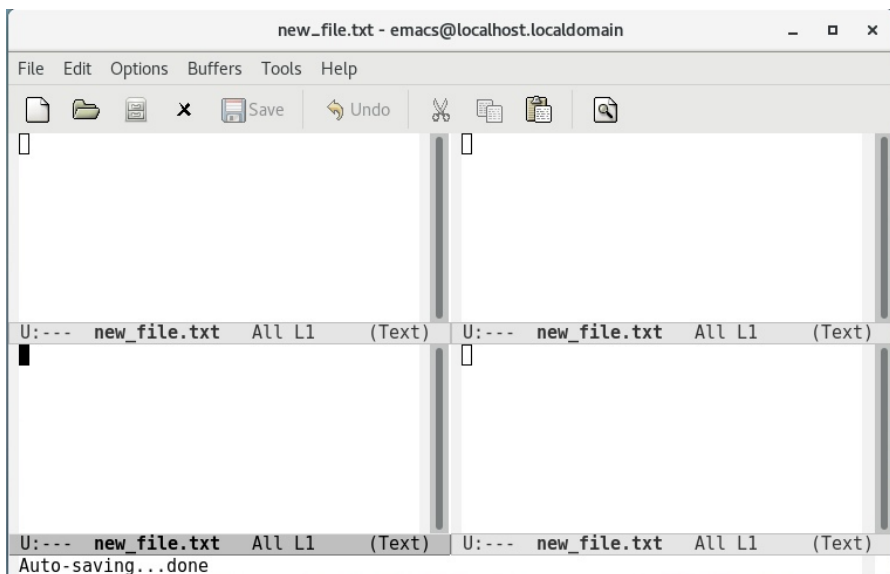
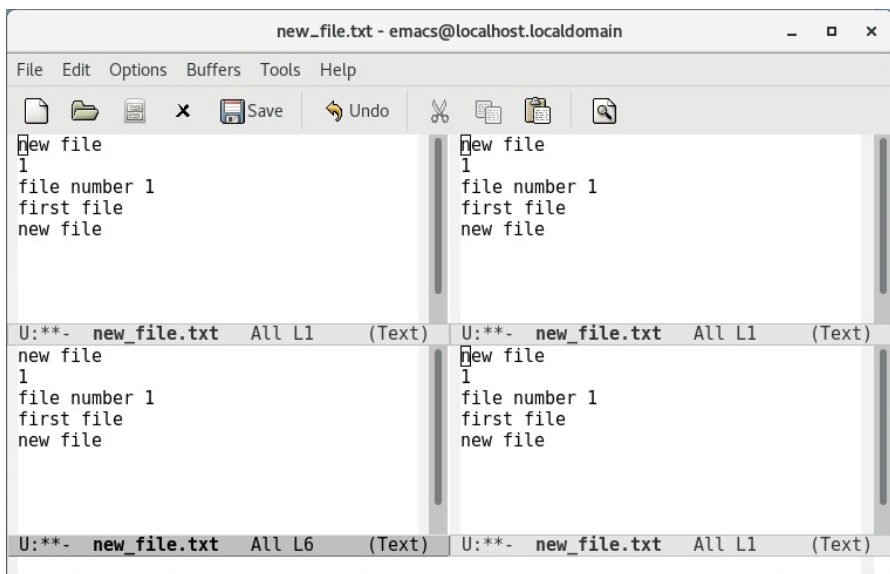


Рисунок 29: ввод текста:



9. Режим поиска

9.1. Переключимся в режим поиска C-s и найдем несколько слов, присутствующих в тексте, а именно слова 1 (рисунок 30) и file (рисунок 31) (найденные слова в тексте подсвечены голубым).

Рисунок 30: поиск слова 1:

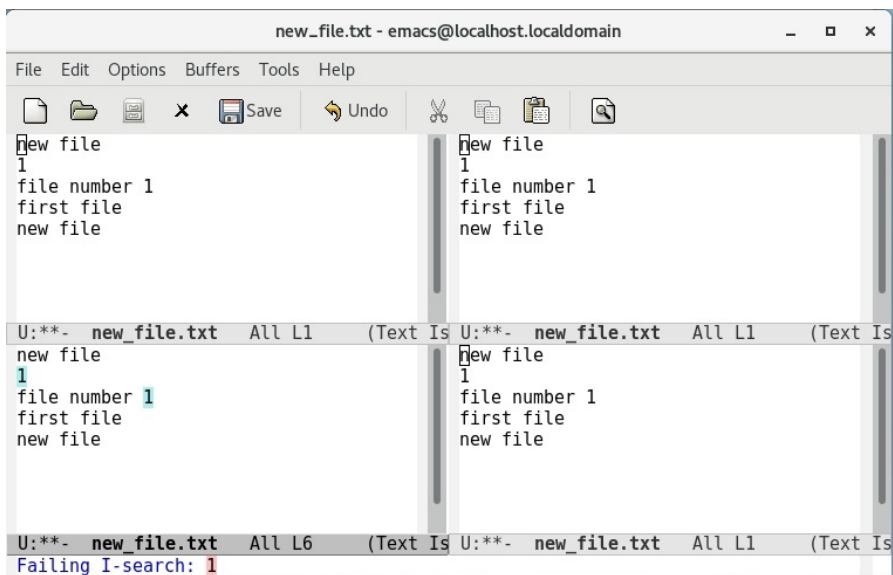
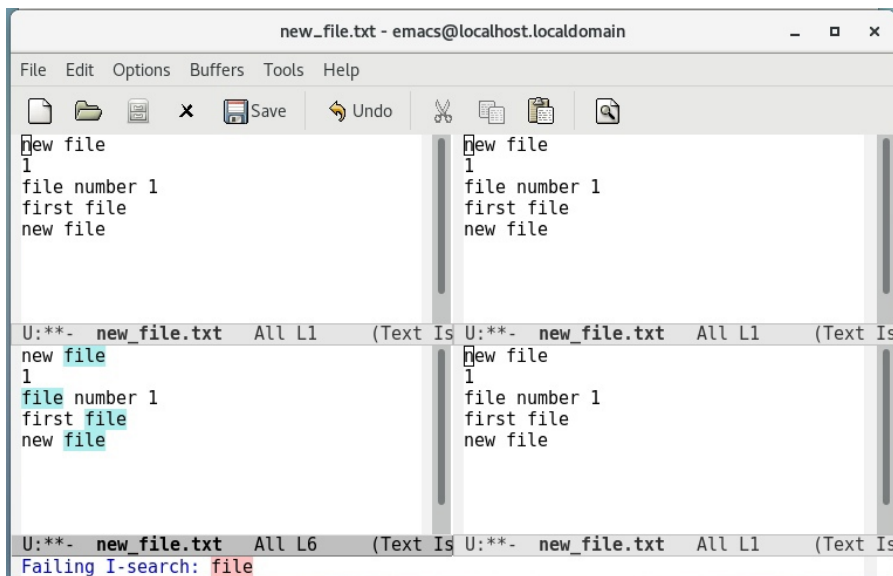


Рисунок 31: поиск слова file:



9.2. Теперь будем переключаться между результатами поиска, нажимая C-s (выбранные поиском слова подсвечены фиолеторым) (рисунки 32-33).

Рисунок 32: переключение между результатами поиска:

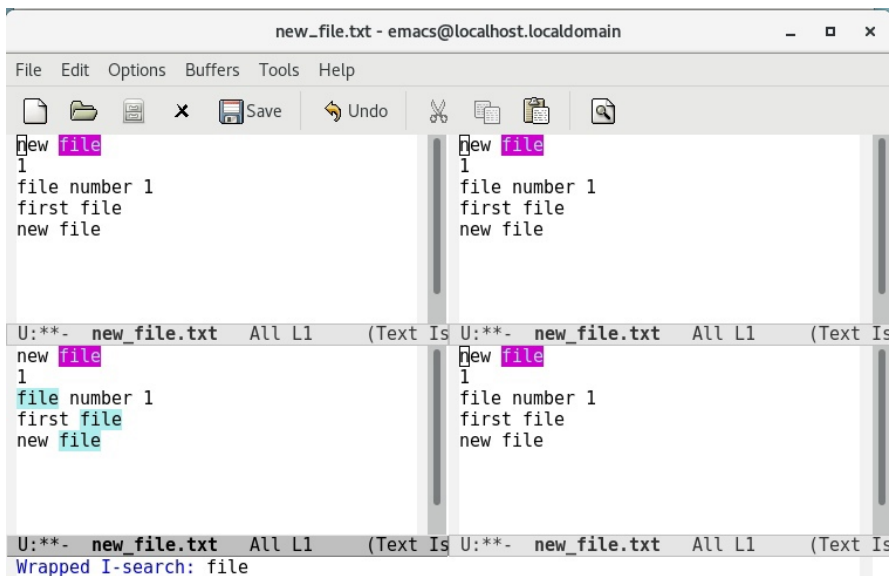
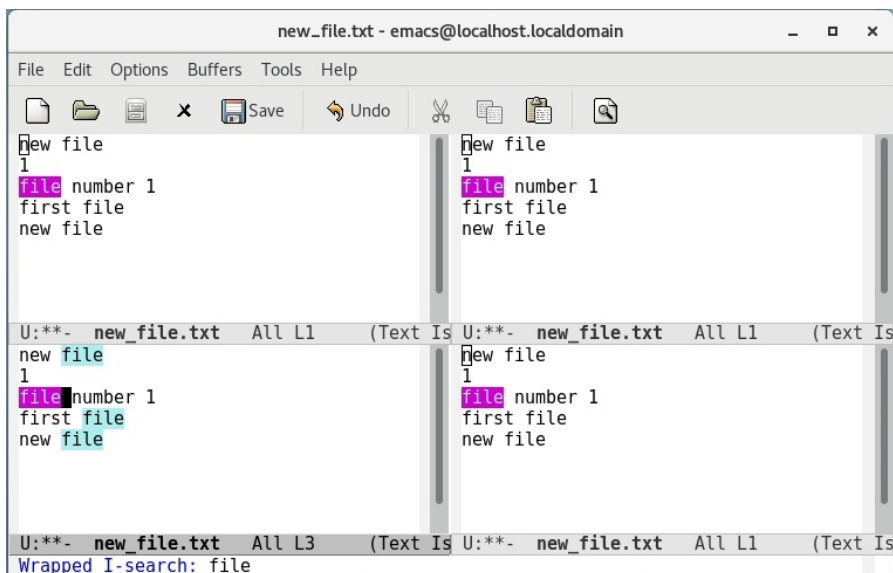
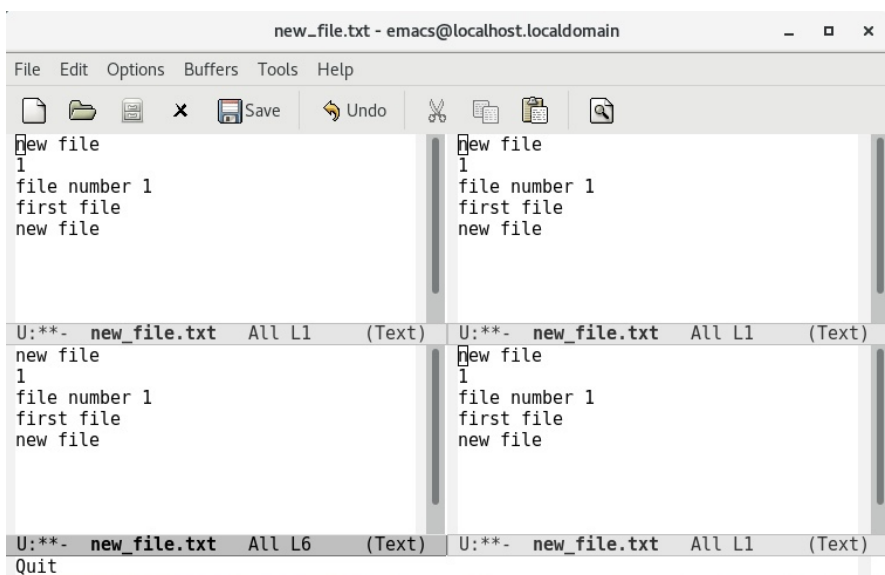


Рисунок 33: переключение между результатами поиска:



9.3. Выйдем из режима поиска, нажав C-g (рисунок 34).

Рисунок 34: выход из режима поиска:



9.4. Перейдем в режим поиска и замены (Esc-%), введем текст, который следует найти и заменить - слово file (рисунок 35), нажмем Enter, затем введем текст для замены - пате (рисунок 36). После того как будут подсвечены результаты поиска (рисунок 37), нажмем ! для подтверждения замены. Видим, что все слова file были заменены на пате, начиная с выделенного на рисунке 37 слова file (рисунок 38).

Рисунок 35: выбор заменяемого слова:

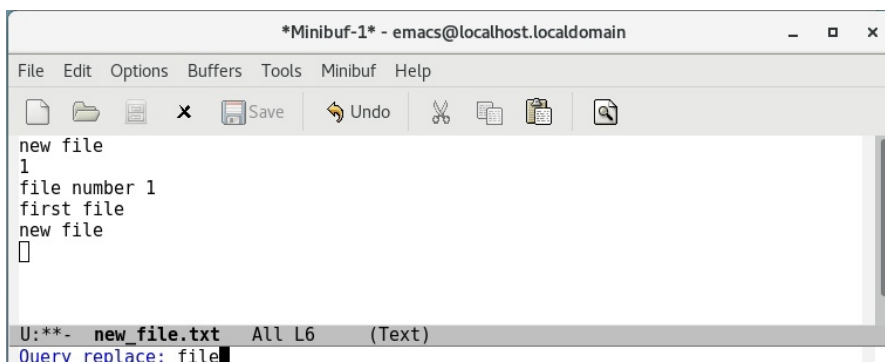


Рисунок 36: выбор слова, на которое хотим заменить:

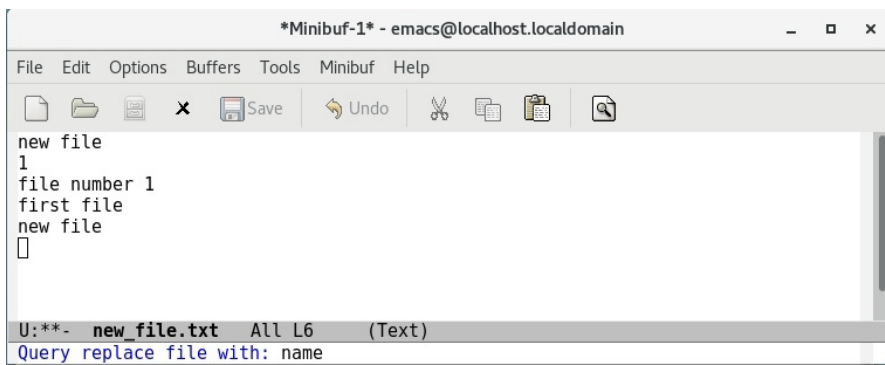


Рисунок 37: результаты поиска:

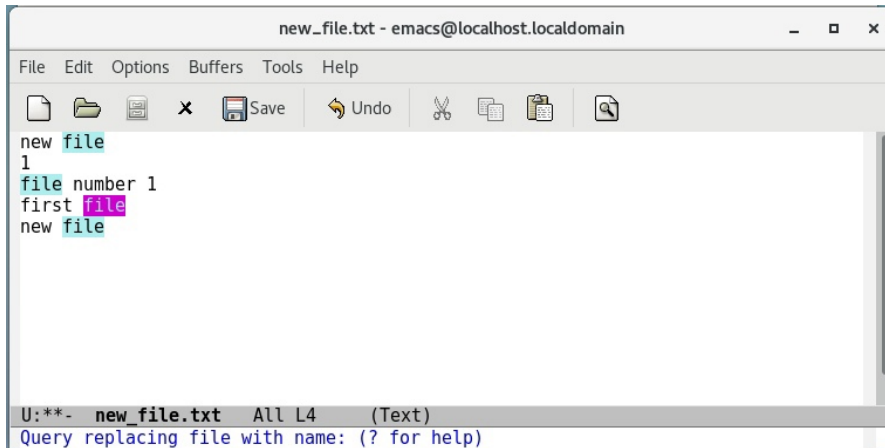
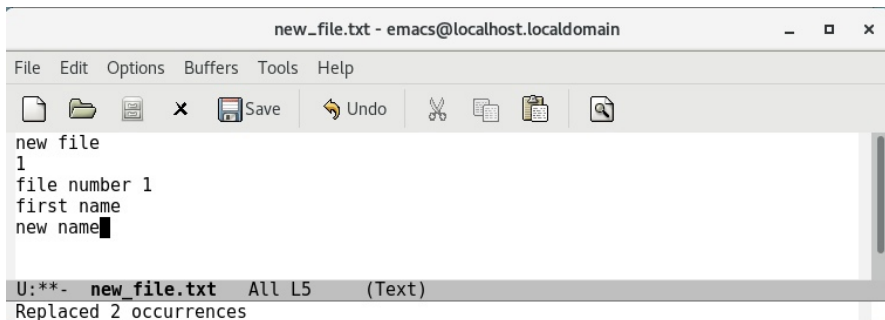


Рисунок 38: замена:



9.5. Испробуем другой режим поиска, нажав `Alt-s` и найдем слово "1" (рисунок 39). Смотрим на вывод поиска (рисунок 40) и видим, что данный режим поиска отличается от предыдущих тем, что он ищет не введенное слово, а строки, в которых оно встречается.

Рисунок 39: поиск слова "1":

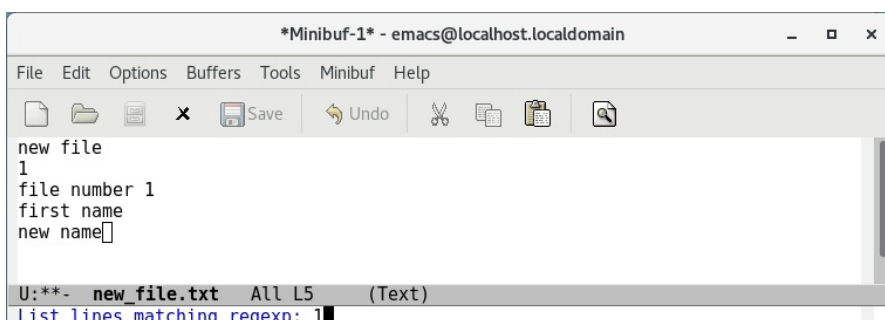
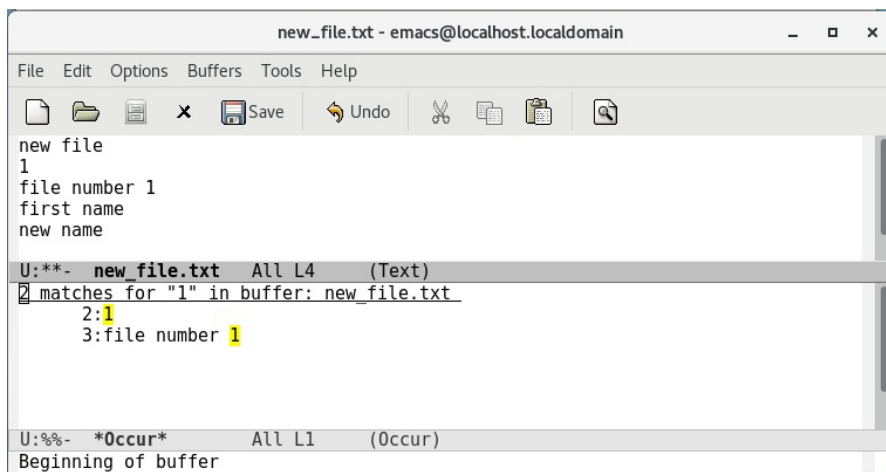


Рисунок 40: вывод поиска:



Вывод:

Познакомилась с операционной системой Linux. Получила практические навыки работы с редактором Emacs. Научилась проделывать с текстом стандартные процедуры редактирования, использовать команды по перемещению курсора, научилась управлять буферами, окнами и режимом поиска.

Библиография:

[1]: [Хабр - Почему Emacs?](#)

[2]: [Справочник команд GNU Emacs](#)