

# **Отчет по лабораторной работе №10**

**Дисциплина**

Филиппова Анна Дмитриевна

# Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выводы	22
4	Контрольные вопросы	23

# Список иллюстраций

2.1	Установка емас	5
2.2	Установка емас	6
2.3	Открытие емас	6
2.4	Создание файла	6
2.5	Набор текста	7
2.6	Вырезаем текст	7
2.7	Вставляем строку	8
2.8	Выделяем область	8
2.9	Копируем и вставляем область	9
2.10	Вырезаем область	10
2.11	Отмена последнего действия	11
2.12	Курсор в начале строки	12
2.13	Курсор в конце строки	13
2.14	Курсор в начале буфера обмена	13
2.15	Курсор в конце буфера обмена	14
2.16	Список активных буферов	14
2.17	Список активных буферов	15
2.18	Переключимся на другой буфер	15
2.19	Закроем окно	15
2.20	Переключаемся между буферами	16
2.21	Делим фрейм на 4 части	16
2.22	Делим фрейм на 4 части	17
2.23	Делим фрейм на 4 части	17
2.24	Создаем файлы	17
2.25	Выбираем буферы	18
2.26	Вводим текст	18
2.27	Режим поиска	18
2.28	Переключаемся между результатами	19
2.29	Переключаемся между результатами	19
2.30	Выходим из режима поиска	20
2.31	Другой режим поиска	21
2.32	Другой режим поиска	21

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Я ознакомилась с теоретическим материалом.
2. Я ознакомилась с редактором emacs.
3. Выполняем упражнения.

### 1 ЗАДАНИЕ

1. Для работы с Emacs необходимо установить данный редактор. (рис. -fig. 2.1)  
(рис. -fig. 2.2)

```
[root@adfilippova adfilippova]# yum install epel-release
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.docker.ru
* epel: mirror.logol.ru
* extras: mirror.docker.ru
* updates: mirror.docker.ru
Разрешение зависимостей
--> Проверка сценария
---> Пакет epel-release.noarch 0:7-11 помечен для обновления
---> Пакет epel-release.noarch 0:7-13 помечен как обновление
--> Проверка зависимостей окончена

Зависимости определены
```

Package	Архитектура	Версия	Репозиторий	Размер
Обновление:				
epel-release	noarch	7-13	epel	15 k

```
Итого по операциям
```

Рис. 2.1: Установка emacs

```
[root@adfilippova adfilippova]# yum install emacs
Загружены модули: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.docker.ru
 * epel: mirror.logol.ru
 * extras: mirror.docker.ru
 * updates: mirror.docker.ru
Разрешение зависимостей
--> Проверка сценария
--> Пакет emacs.x86_64 1:24.3-23.el7 помечен для установки
--> Обработка зависимостей: emacs-common = 1:24.3-23.el7 пакета: 1:emacs-24.3-23.el7.x86_64
--> Обработка зависимостей: libotf.so.0()(64bit) пакета: 1:emacs-24.3-23.el7.x86_64
--> Обработка зависимостей: libMagickWand-6.Q16.so.6()(64bit) пакета: 1:emacs-24.3-23.el7.x86_64
```

Рис. 2.2: Установка emacs

2. Откроем редактор Emacs с помощью команды «emacs &». (рис. -fig. 2.3)

```
[adfilippova@adfilippova ~]$ emacs &
[1] 4502
```

Рис. 2.3: Открытие emacs

3. Создадим файл lab07.sh с помощью комбинации «Ctrl-x» «Ctrl-f». (рис. -fig. 2.4)

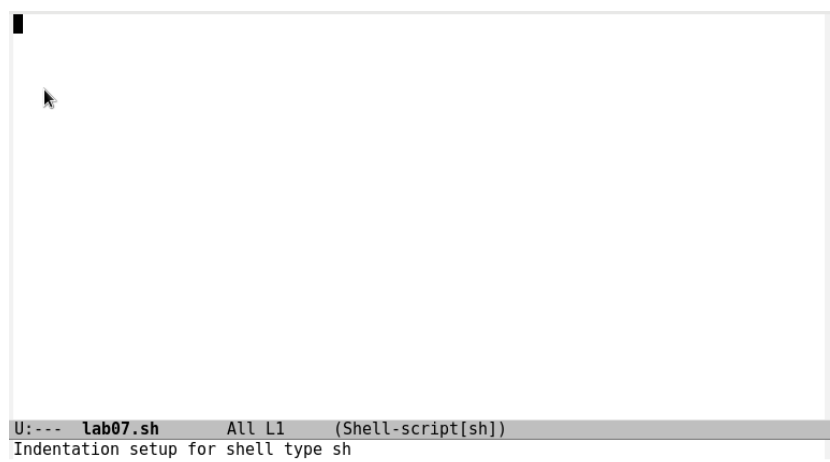


Рис. 2.4: Создание файла

4. Набираем необходимый текст. (рис. -fig. 2.5)

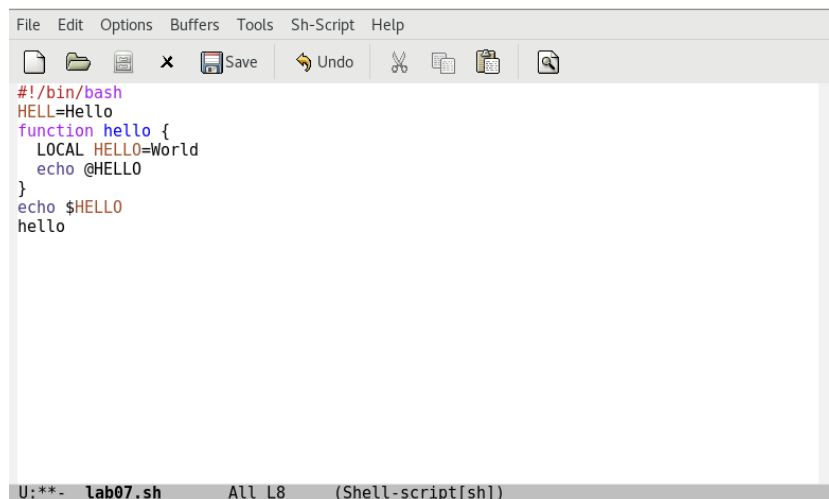


Рис. 2.5: Набор текста

5. 1) Сохраним файл с помощью комбинации «Ctrl-x» «Ctrl-s».
- 2) Вырежем одной командой целую строку «Ctrl-k». (рис. -fig. 2.6)

```
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo @HELLO
}
echo $HELLO
hello
```

Рис. 2.6: Вырезаем текст

- 3) Вставим эту строку в конец файла «Ctrl-y». (рис. -fig. 2.7)

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

Рис. 2.7: Вставляем строку

- 4) Выделим область текста «Ctrl-space». (рис. -fig. 2.8)

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

Рис. 2.8: Выделяем область

- 5) Скопируем область в буфер обмена «Alt-w». Вставляем область в конец файла «Ctrl-y». (рис. -fig. 2.9)



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

Рис. 2.9: Копируем и вставляем область

- 6) Вновь выделим эту область «Ctrl-space» и на этот раз вырежем её «Ctrl-w».  
(рис. -fig. 2.10)

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

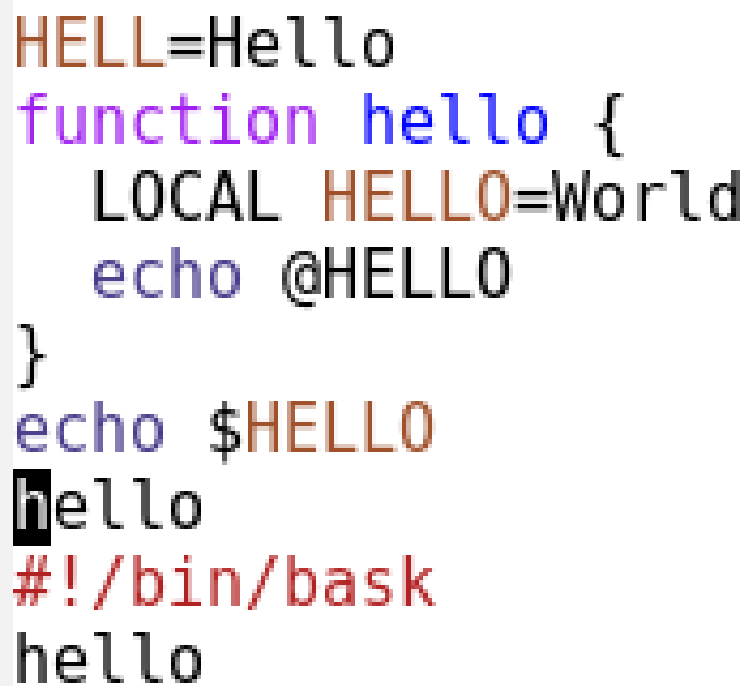
Рис. 2.10: Вырезаем область

7) Отменим последнее действие «Ctrl-/». (рис. -fig. 2.11)

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

Рис. 2.11: Отмена последнего действия

6. 1) Переместим курсор в начало строки «Ctrl-a». (рис. -fig. 2.12)



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

Рис. 2.12: Курсор в начале строки

2) Переместим курсор в конец строки «Ctrl-e». (рис. -@fig:013)

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

Рис. 2.13: Курсор в конце строки

3) Переместим курсор в начало буфера «Alt-<». (рис. -@fig:014)

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

Рис. 2.14: Курсор в начале буфера обмена

4) Переместим курсор в конец буфера «Alt-<». (рис. -@fig:015)

```

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

```

Рис. 2.15: Курсор в конце буфера обмена

7. 1) Выведем список активных буферов на экран «Ctrl-x» «Ctrl-b». (рис. - fig. 2.16)

```

echo @HELLO
}
echo $HELLO
hello
#!/bin/bask
hello

```

U:*	lab07.sh	Bot	L10	(Shell-script[sh])
CRM	Buffer		Size	Mode
	* lab07.sh		100	Shell-script[sh] ~/lab07.sh
	% *GNU Emacs*		901	Fundamental
	*scratch*		191	Lisp Interaction
	*Messages*		1412	Fundamental

Рис. 2.16: Список активных буферов

- 2) Переместимся во вновь открытое окно «Ctrl-x o» со списком открытых буферов и переключимся на другой буфер. (рис. -fig. 2.17) (рис. -fig. 2.18)

```

echo @HELLO
}
echo $HELLO
hello
#!/bin/bask
hello

```

U:***	lab07.sh	Bot	L10	(Shell-script[sh])
CRM	Buffer		Size	Mode
	* lab07.sh		100	Shell-script[sh] ~/lab07.sh
	* GNU Emacs*		901	Fundamental
	* scratch*		191	Lisp Interaction
	* Messages*		1412	Fundamental

Рис. 2.17: Список активных буферов

```

echo @HELLO
}
echo $HELLO
hello
#!/bin/bask
hello

```

U:\*\*\* lab07.sh Bot L10 (Shell-script[sh])

Useful tasks:

- [Visit New File](#) Specify a new file's name, to edit the file
- [Open Home Directory](#) Open your home directory, to operate on its files
- [Customize Startup](#) Change initialization settings including this screen

GNU Emacs 24.3.1 (x86\_64-redhat-linux-gnu, GTK+ Version 3.22.30)  
of 2020-04-03 on x86-01.bsys.centos.org  
Copyright (C) 2013 Free Software Foundation, Inc.

U:\*\*\* \*GNU Emacs\* Bot L16 (Fundamental)  
Find tag (default hello):

Рис. 2.18: Переключимся на другой буфер

3) Закроем это окно «Ctrl-x 0». (рис. -fig. 2.19)

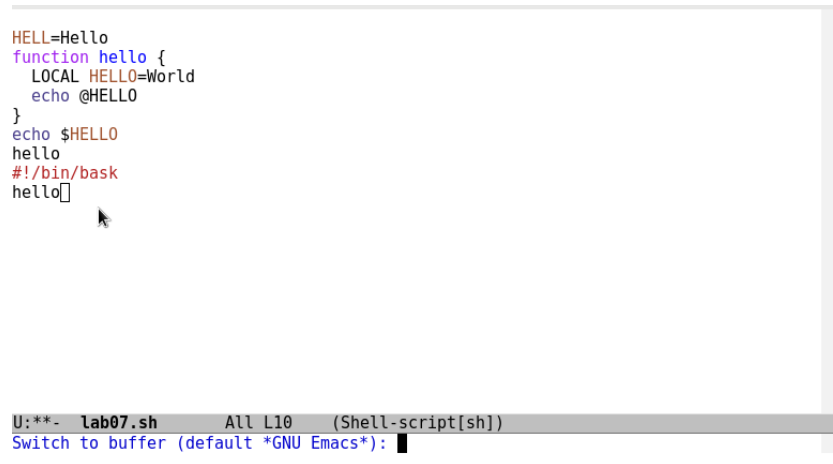
```

echo @HELLO
}
echo $HELLO
hello
#!/bin/bask
hello

```

Рис. 2.19: Закроем окно

- 4) Теперь вновь переключимся между буферами, но уже без вывода их списка на экран «Ctrl-x b». (рис. -fig. 2.20)



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

U:\*\* - lab07.sh All L10 (Shell-script[sh])  
Switch to buffer (default \*GNU Emacs\*):

Рис. 2.20: Переключаемся между буферами

8. 1) Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали, а затем каждое из этих окон на две части по горизонтали. (рис. -fig. 2.21) (рис. -fig. 2.22) (рис. -fig. 2.23)



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

U:\*\* - lab07.sh All L10 (Shell-script[sh])

Рис. 2.21: Делим фрейм на 4 части



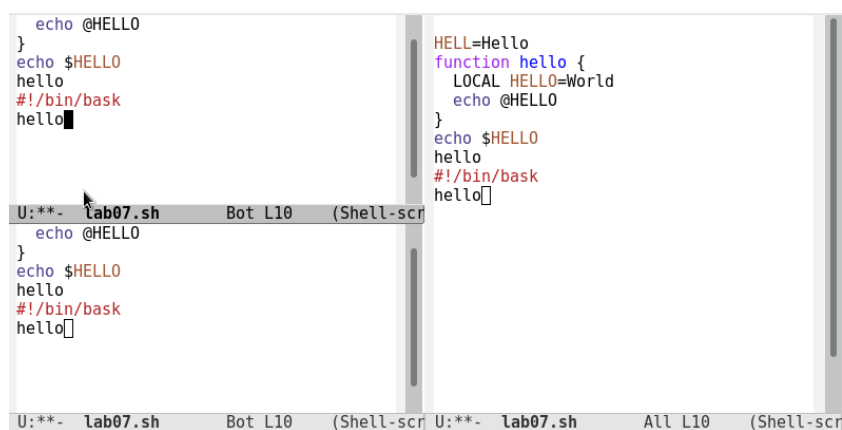


Рис. 2.22: Делим фрейм на 4 части



Рис. 2.23: Делим фрейм на 4 части

- 2) В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста. Для этого предварительно создадим эти файлы с помощью команд «touch e1.txt», «touch e2.txt», «touch e3.txt», «touch e4.txt». (рис. -fig. 2.24) (рис. -fig. 2.25) (рис. -fig. 2.26)

```
[adfilippova@adfilippova ~]$ touch e1.txt
[adfilippova@adfilippova ~]$ touche2.txt
bash: touche2.txt: команда не найдена...
[adfilippova@adfilippova ~]$ touch e2.txt
[adfilippova@adfilippova ~]$ touch e3.txt
[adfilippova@adfilippova ~]$ touch e4.txt
[adfilippova@adfilippova ~]$
```

Рис. 2.24: Создаем файлы

Switch to buffer (default e3.txt): e1.txt

Рис. 2.25: Выбираем буферы

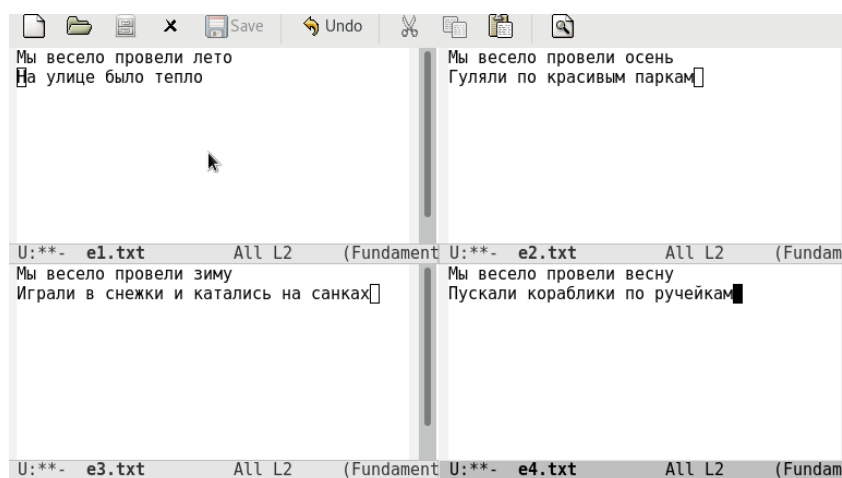


Рис. 2.26: Вводим текст

9. 1) Переключимся в режим поиска «Ctrl-s» и найдем несколько слов, присутствующих в тексте. (рис. -fig. 2.27)

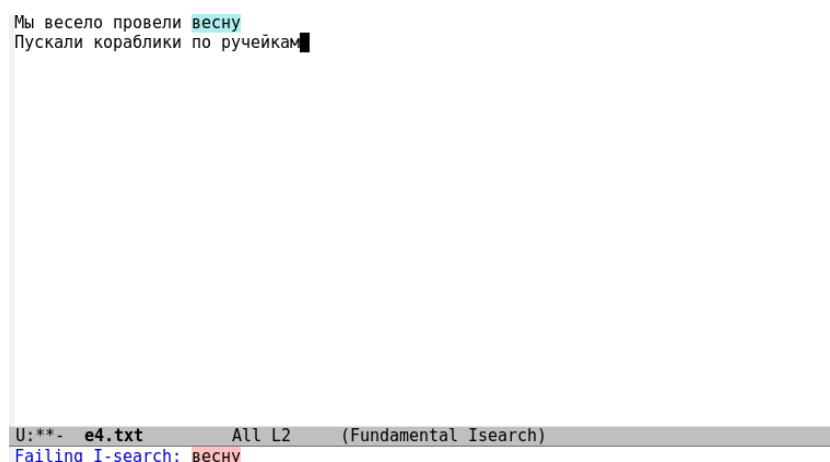


Рис. 2.27: Режим поиска

- 2) Переключимся между результатами поиска, нажимая «Ctrl-s». (рис. -fig. 2.28)  
(рис. -fig. 2.29)

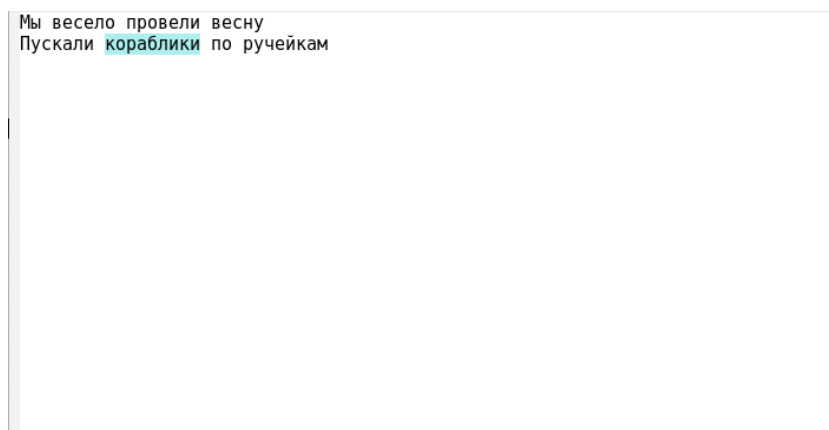


Рис. 2.28: Переключаемся между результатами

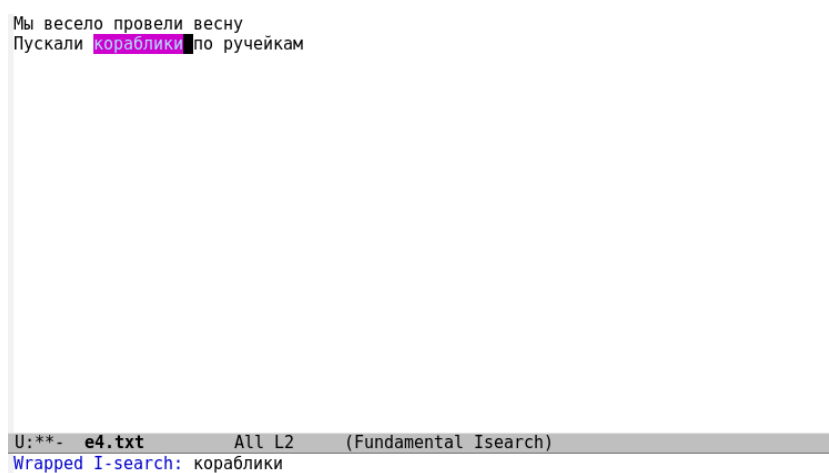


Рис. 2.29: Переключаемся между результатами

3) Выходим из режима поиска, нажав «Ctrl-g». (рис. -fig. 2.30)

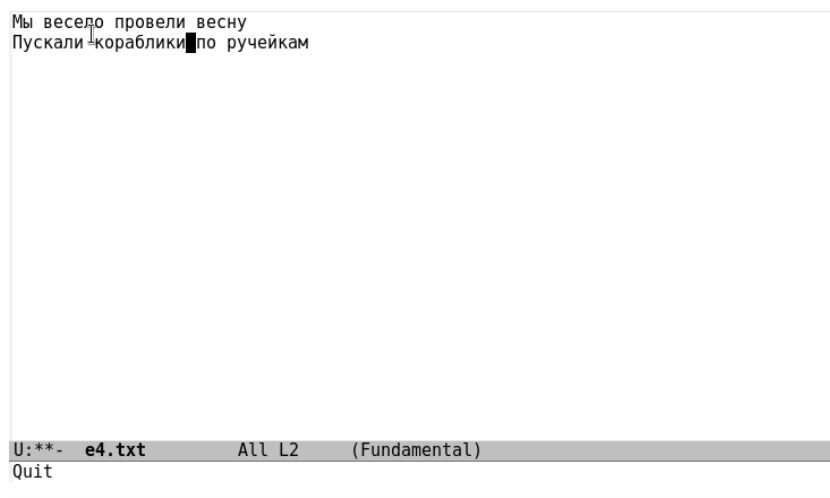


Рис. 2.30: Выходим из режима поиска

- 4) Перейдем в режим поиска и замены «Alt-%», но в связи с тем, что на Cent Os 7, установленном на опреационную систему Mac OS, данное сочетание клавиш не сработало, поэтому данное задание было невозможно выполнить.
- 5) Пробуем другой режим поиска, нажав «Alt-s o». Данный вид поиска отличается от обычного тем, что тут считывается строка поиска, которая трактуется как регулярное выражение, и не осуществляется поиск точного совпадения в тексте буфера. Регулярное выражение – это образец, который обозначает набор строк, возможно, и неограниченный набор. (рис. -fig. 2.31) (рис. -fig. 2.32)

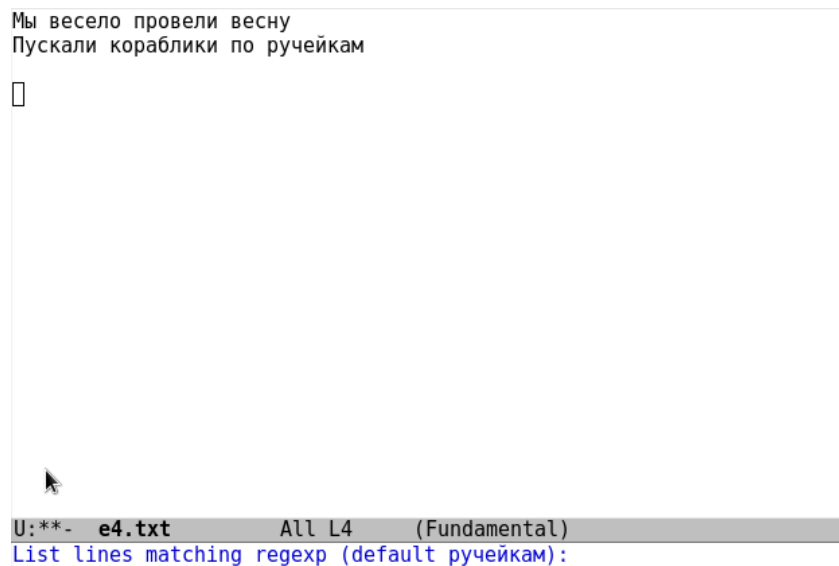


Рис. 2.31: Другой режим поиска

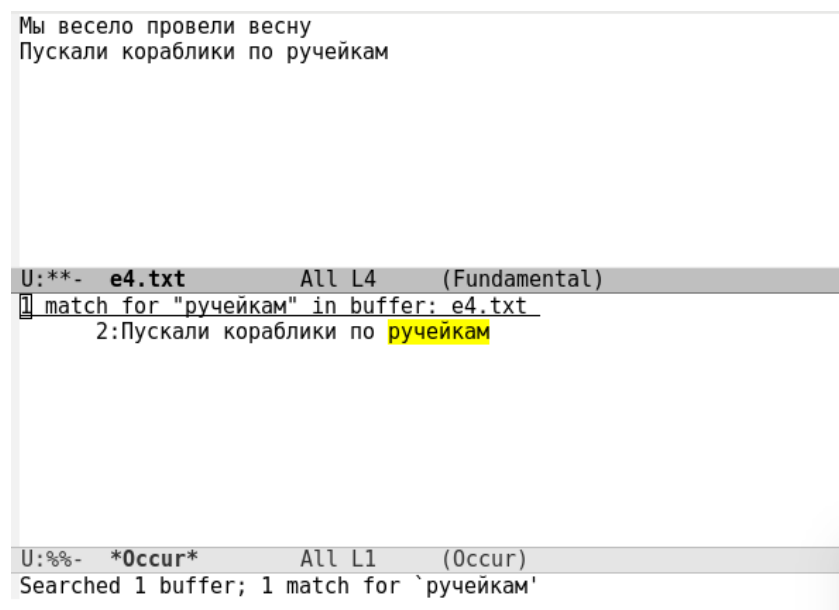


Рис. 2.32: Другой режим поиска


## 3 Выводы

Я познакомилась с операционной системой Linux и получила практические навыки работы с редактором Emacs.

## 4 Контрольные вопросы

1. 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 и т.д. 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.
2. Основную трудность для новичков при освоении данного редактора могут составлять большое количество команд, комбинаций клавиш, которые не получится все запомнить с первого раза и поэтому придется часто обращаться к справочным материалам.
3. Буфер – это объект, представляющий собой текст. Если имеется несколько буферов, то редактировать можно только один. Обычно буфер считывает

данные из файла или записывает в файл данные из буфера. Окно – это область экрана, отображающая буфер. При запуске редактора отображается одно окно, но при обращении к некоторым функциям могут открыться дополнительные окна. Окна Emacs и окна графической среды X Window – разные вещи. Одно окно X Window может быть разбито на несколько окон в смысле Emacs, в каждом из которых отображается отдельный буфер.

4. Да, можно.
5. При запуске Emacs по умолчанию создаются следующие буферы:
  - «scratch» (буфер для несохраненного текста)
  - «Messages» (журнал ошибок, включающий также информацию, которая появляется в области EchoArea)
  - «GNU Emacs» (справочный буфер о редакторе)
6. C-c | сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и нажимаю «|» C-c C-| сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и, удерживая «ctrl», нажимаю «|»
7. Чтобы поделить окно на две части необходимо воспользоваться комбинацией «Ctrl-x 3» (по вертикали) или «Ctrl-x 2» (по горизонтали).
8. Настройки Emacs хранятся в файле .emacs.
9. По умолчанию клавиша «» удаляет символ перед курсором, но в редакторе её можно переназначить. Для этого необходимо изменить конфигурацию файла .emacs.
10. Более удобным я считаю редактор emacs, потому что в нем проще открывать другие файлы, можно использовать сразу несколько окон, нет «Командного режима», «Режима ввода», «Режима командной строки», которые являются немного непривычными и в какой-то степени неудобными.