

Лабораторная работа №10

Российский университет дружбы народов

Андреев Владислав Владимирович

Содержание

Цель работы	1
Задание	1
Выполнение лабораторной работы.....	1
Выводы	11
Ответы на контрольные вопросы	11

Цель работы

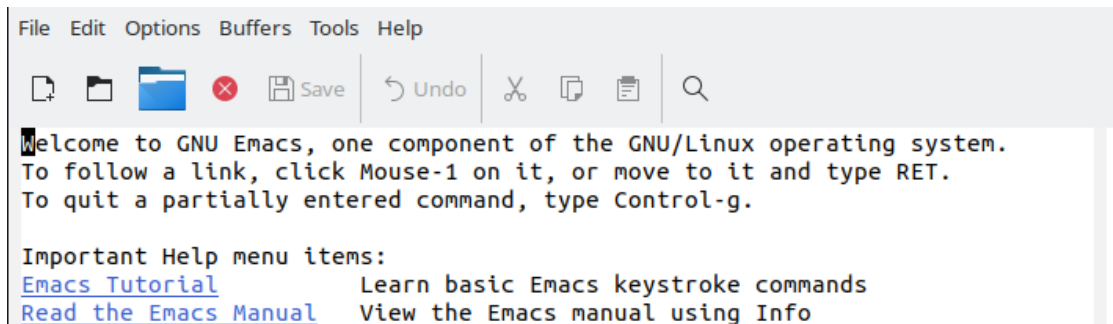
Познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

Задание

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

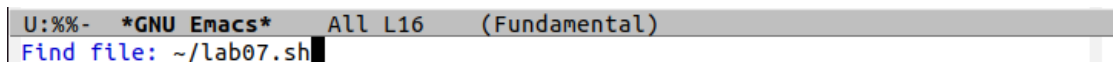
Выполнение лабораторной работы

- 1.Открываем emacs(Рисунок 3.1).



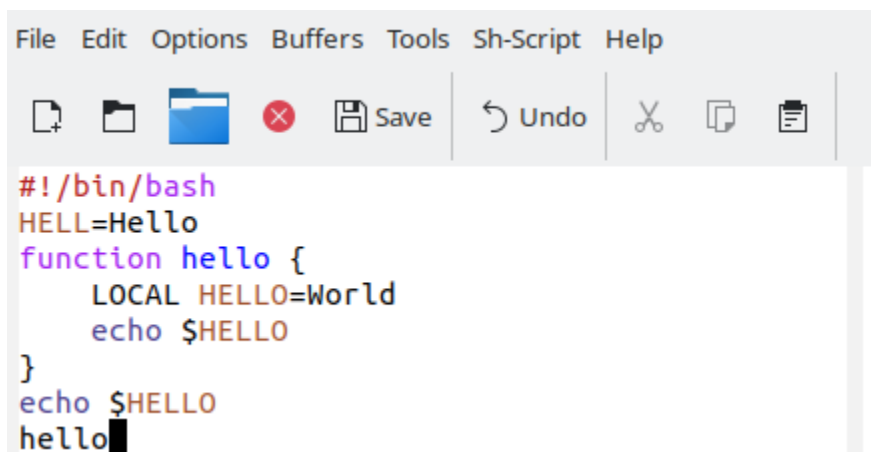
Открытие

2. Создаём файл lab07.sh с помощью комбинации Ctrl-x и Ctrl-f (Рисунок 3.2).



Создание файла

3. Набираем текст (Рисунок 3.3).



Текст

4. Сохраняем файл с помощью комбинации Ctrl-x Ctrl-s.

5. Проделываем с текстом стандартные процедуры редактирования. (Примечание: клавиша Ctrl обозначена как C, а клавиша Alt обозначена как A).

5.1. Вырезаем одной командой целую строку (C-k) (Рисунок 3.4).

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

Вырезание

5.2. Вставляем эту строку в конец файла (С-у) (Рисунок 3.5).

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

Вставка

5.3. Выделяем область текста (С-space) (Рисунок 3.6).

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

Выделение

5.4.Копируем область в буфер обмена (A-w).

5.5.Вставляем область в конец файла(Рисунок 3.7).

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

Вставка в конец

5.6.Выделяем эту область второй раз и на этот раз вырезаем её (C-w)(Рисунок 3.8).

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

    echo $HELLO
```

Вырезание области

5.7.Отменяем последнее действие (C-/)(Рисунок 3.9).

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

    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
function hello {

    echo $HELLO
```

Отмена действия

6.Учимся использовать команды по перемещению курсора.

6.1.Перемещаем курсор в начало строки (C-a)(Рисунок 3.10).

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

Перемещение в начало

6.2.Перемещаем курсор в конец строки (C-e)(Рисунок 3.11).

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

Перемещение в конец

6.3.Перемещаем курсор в начало буфера (A-<)(Рисунок 3.12).

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

Перемещение в начало буфера

6.4.Перемещаем курсор в конец буфера (A->)(Рисунок 3.13).

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

Перемещение в конец буфера

7.Учимся управлять буферами.

7.1. Выводим список активных буферов на экран (C-x C-b) (Рисунок 3.14).

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

-:--- lab07.sh		Top L1	(Shell-script[bash])
CRM	Buffer	Size	Mode
	lab07.sh	131	Shell-script[... ~/lab07.sh
%	*GNU Emacs*	898	Fundamental
	scratch	145	Lisp Interaction
%*	*Messages*	1076	Messages

Активные буферы

7.2. Перемещаемся во вновь открытое окно (C-x o) со списком открытых буферов и переключаемся на другой буфер (Рисунок 3.15, 3.16).

-:--- lab07.sh		Top L1	(Shell-script[bash])
CRM	Buffer	Size	Mode
.	lab07.sh	131	Shell-script[... ~/lab07.sh
%	*GNU Emacs*	898	Fundamental
	scratch	145	Lisp Interaction
%*	*Messages*	1076	Messages

Перемещение в окно

-:--- lab07.sh		Top L4	(Shell-script[bash])
Loading /etc/emacs/site-start.d/50latexmk.el (source)...done			
Loading /etc/emacs/site-start.d/50texlive-lang-english.el (source)...done			
For information about GNU Emacs and the GNU system, type C-h C-a.			
Setting up indent for shell type bash			
Indentation variables are now local.			
Indentation setup for shell type bash			
U:%*- *Messages* Bot L19 (Messages)			

Перемещение в другой буфер

7.4.Теперь вновь переключаемся между буферами, но уже без вывода их списка на экран (С-х b)(Рисунок 3.17, 3.18).

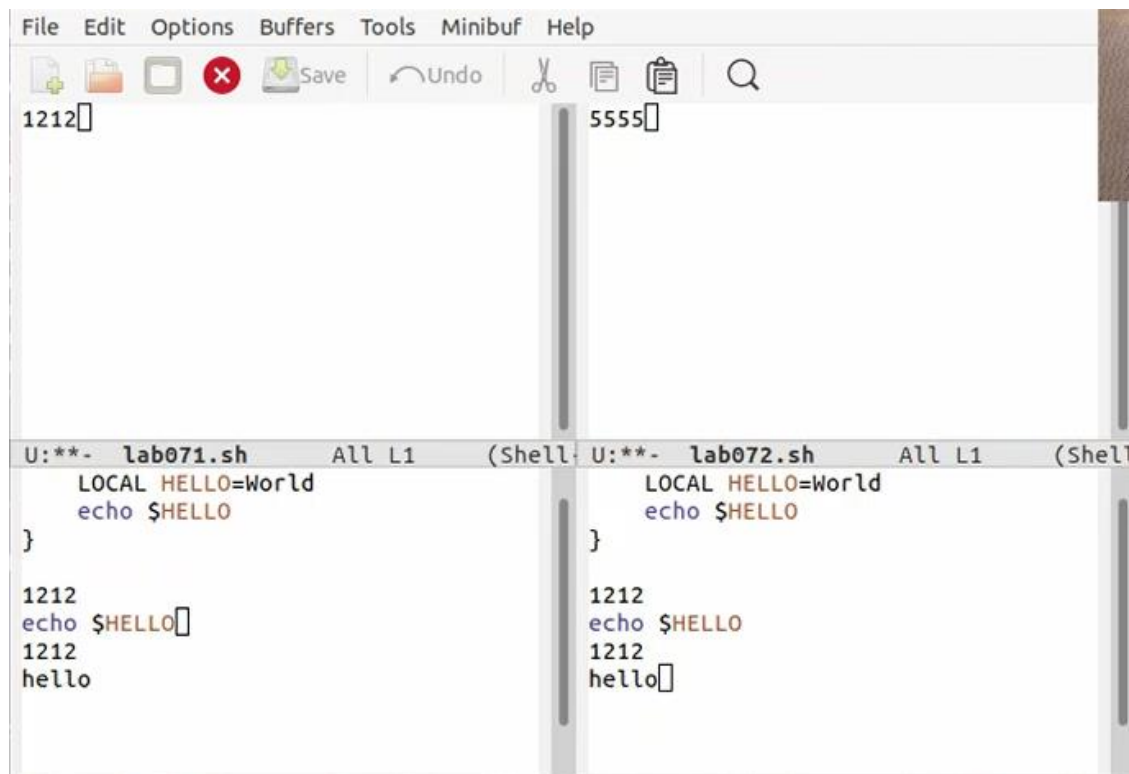
Переключение без списка

Переключение

8.1. Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2) (Рисунок 3.19).

Фрейм на 4 части

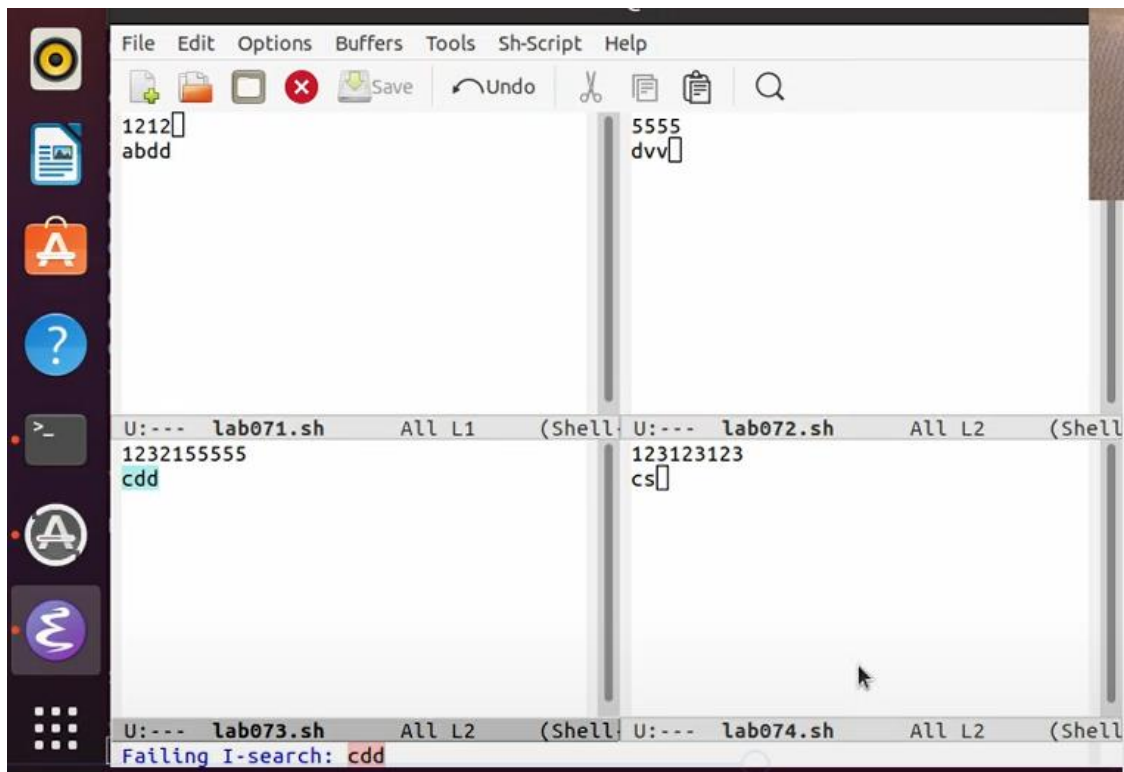
8.2. В каждом из четырёх созданных окон откроем новый буфер (файл) и введём несколько строк текста (Рисунок 3.20).



Новые буферы

9.Учимся режиму поиска.

9.1.Переключаемся в режим поиска (C-s) и находим несколько слов, присутствующих в тексте(Рисунок 3.21).



Режим поиска

Выводы

Я познакомился с операционной системой Linux. Я получил практические навыки работы с редактором Emacs.

Ответы на 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 XWindow – разные вещи. Одно окно XWindow 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); «GNUEmacs»(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 «←»(стрелочка) удаляет символ перед курсором, но в редакторе её можно переназначить. Для этого необходимо изменить конфигурацию файла .emacs.

10.Более удобным я считаю редактор emacs, нежели vi, 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.