

Презентация по лабораторной работе No11

Операционные системы

Нджову Н.

18 Апреля 2024

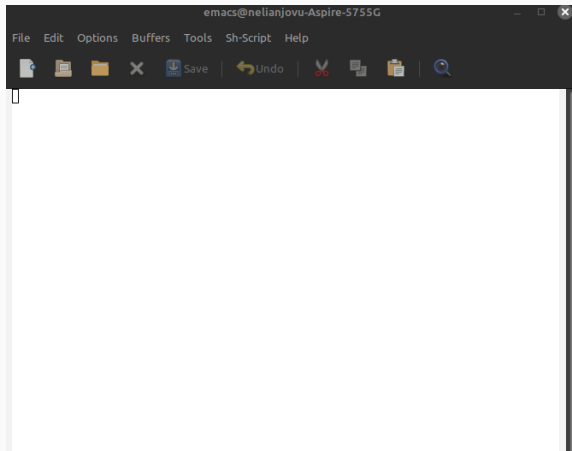
Российский университет дружбы народов, Москва, Россия

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

1. Основные команды emacs
2. Научиться использовать команды по перемещению курсора.
3. Управление буферами
4. Управление окнами.
5. Режим поиска

1. Основные команды emacs

Я открываю emacs и создаю файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f)(рис.1). Emacs представляет собой мощный экраннй редактор текста, написанный на языке высокого уровня Elisp.



1. Основные команды emacs

Я записываю текст в только созданный файл. Потом я сохраняю файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s)(рис.2)

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

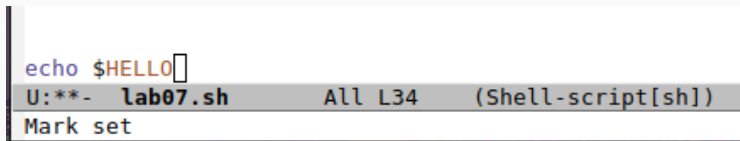
1. Основные команды emacs

Я вырезаю одной командой целую строку(C-k)(рис.3)

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

1. Основные команды emacs

Я вставляю эту строку в конец файла с помощью C-y(рис.4)

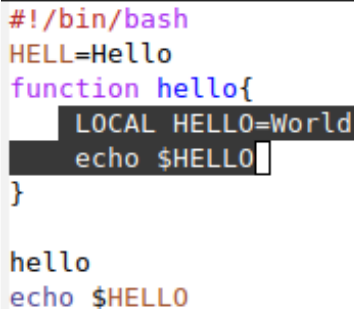


```
echo $HELLO
U:**- lab07.sh All L34 (Shell-script[sh])
Mark set
```

Рис. 4: вставление строки

1. Основные команды emacs

Я выделяю область текста с помощью C-space(рис.5)



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

hello
echo $HELLO
```

Рис. 5: выделение текста

1. Основные команды emacs

Я скопирую область в буфер обмена (M-w) и вставляю область в конец файла(рис.6).
Буфер — объект, представляющий какой-либо текст. Область вывода — одна или несколько строк внизу фрейма, в которой Emacs выводит различные сообщения, а также запрашивает подтверждения и дополнительную информацию от пользователя.

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

hello
echo $HELLO
LOCAL HELLO=World
```

1. Основные команды emacs

Я ещё раз выделяю эту область и на этот раз вырезаю её(C-w)(рис.7)

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

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

1. Основные команды emacs

Потом я отменяю последнее действие(рис.8)

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

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

2. Научиться использовать команды по перемещению курсора.

Сначала я перемещаю курсор в начало строки с помощью C-a(рис.9)

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

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

2. Научиться использовать команды по перемещению курсора.


После этого я перемещаю курсор в конец строки(C-e)(рис.10)

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

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

2. Научиться использовать команды по перемещению курсора.

Затем я перемещаю курсор в начало буфера с помощью M(alt)-<(рис.11)



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

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

2. Научиться использовать команды по перемещению курсора.

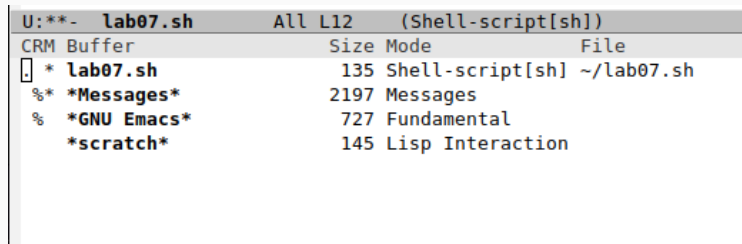
Потом я перемещаю курсор в конец буфера(M->)(рис.12)

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

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

3. Управление буферами

Я выведу список активных буферов на экран (C-x C-b)(рис.13)

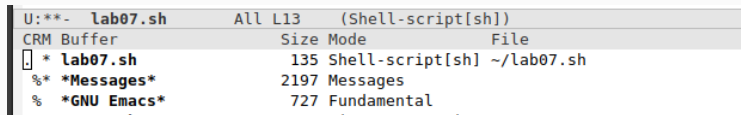
A screenshot of the Emacs buffer list window, which is titled 'U:**- lab07.sh All L12 (Shell-script[sh])'. The window shows a table of active buffers. The first buffer is '* lab07.sh' with a size of 135, mode 'Shell-script[sh]', and file path '~/lab07.sh'. It is selected with a cursor. Below it are three other buffers: '*Messages*' (size 2197, mode 'Messages'), '*GNU Emacs*' (size 727, mode 'Fundamental'), and '*scratch*' (size 145, mode 'Lisp Interaction').

U:**- lab07.sh All L12 (Shell-script[sh])				
CRM Buffer	Size	Mode	File	
<input checked="" type="checkbox"/> * lab07.sh	135	Shell-script[sh]	~/lab07.sh	
%* *Messages*	2197	Messages		
% *GNU Emacs*	727	Fundamental		
scratch	145	Lisp Interaction		

Рис. 13: список активных буферов

3. Управление буферами

Я перемещаю во вновь открытое окно “(С-х) о” со списком открытых буферов и переключаюсь на другой буфер(рис.14). Окно — прямоугольная область фрейма, отображающая один из буферов.

A screenshot of the Emacs buffer list window. The window title is 'U:**- lab07.sh All L13 (Shell-script[sh])'. It contains a table with columns 'CRM Buffer', 'Size', 'Mode', and 'File'. The first row is highlighted with a mouse cursor pointing to the buffer name '* lab07.sh'.

U:**- lab07.sh All L13 (Shell-script[sh])			
CRM Buffer	Size	Mode	File
* lab07.sh	135	Shell-script[sh]	~/lab07.sh
%* *Messages*	2197	Messages	
% *GNU Emacs*	727	Fundamental	

Рис. 14: перемещение на буфером

3. Управление буферами

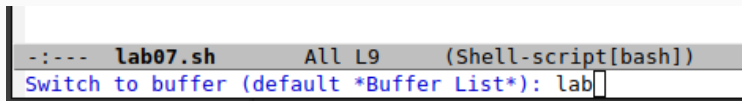
После этого я закрываю это окно (С-х 0)(рис.15)

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

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

3. Управление буферами

Открываю другой буфер без вывода их списка на экран с помощью C-x b(рис.16)

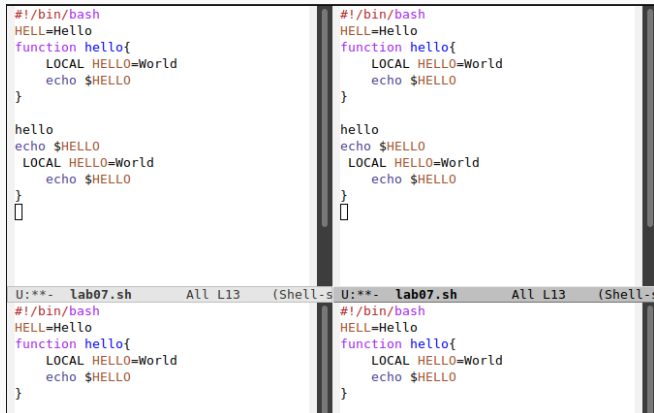


The screenshot shows a terminal window with a dark background. The top line of the terminal is a header bar with the text `-:--- lab07.sh All L9 (Shell-script[bash])`. Below this, the prompt `Switch to buffer (default *Buffer List*):` is displayed in blue. The user has entered the text `lab` and the cursor is positioned at the end of the input.

Рис. 16: Открытие другого буфера

4. Управление окнами.

Я разделяю фрейм на 4 части: разделяю фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2)(рис.17). Фрейм соответствует окну в обычном понимании этого слова. Каждый фрейм содержит область вывода и одно или несколько окон Emacs



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

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

U:**- lab07.sh All L13 (Shell-s

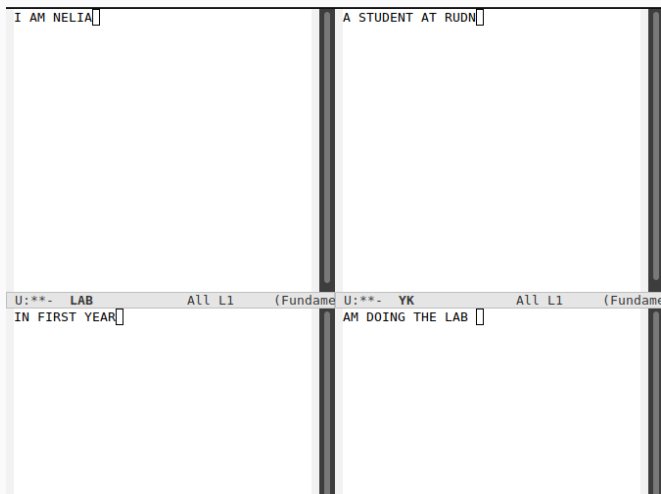
```
#!/bin/bash
HELL=Hello
function hello{
  LOCAL HELLO=World
  echo $HELLO
}
```

U:**- lab07.sh All L13 (Shell-s

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

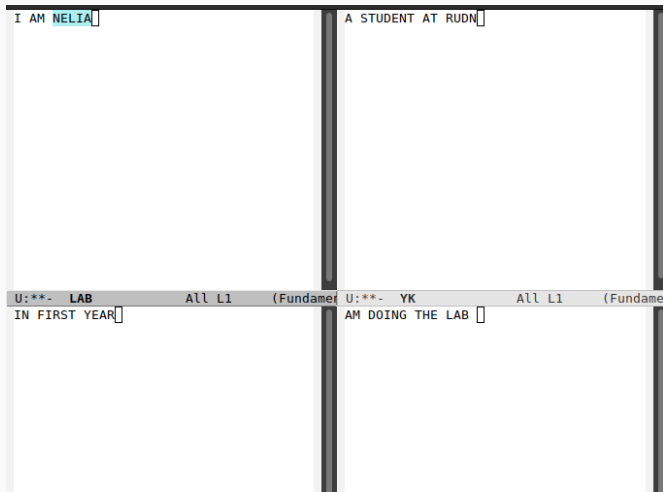
4. Управление окнами.

В каждом из четырёх созданных окон, я открываю новый буфер (файл) и введу несколько строк текста(рис.18)



5. Режим поиска

Я переключаюсь в режим поиска (C-s) и наиду несколько слов, присутствующих в тексте(рис.19)



5. Режим поиска

Я выхожу из режима поиска, нажав C-g(рис.20)

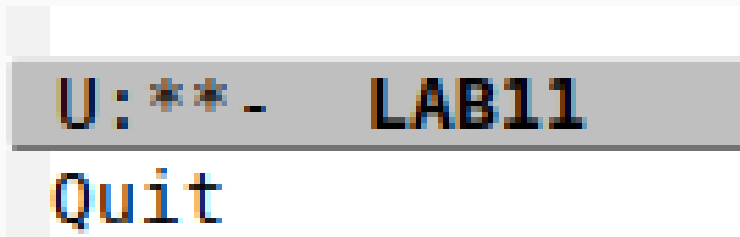
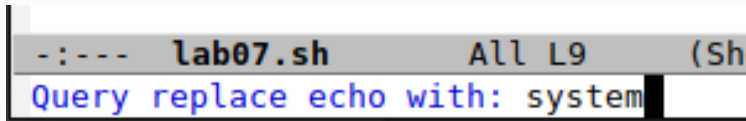


Рис. 20: выход из режима поиска

5. Режим поиска

Перехожу в режим поиска и замены с помощью M-%, ввожу какое слово хочу заменить, затем ввожу на какое хочу заменить(рис.21)



```
-:--- lab07.sh All L9 (Sh  
Query replace echo with: system
```

Рис. 21: Замена слова

5. Режим поиска

С помощью M-s о перехожу в другой режим поиска. Он отличается от предыдущего тем, что выводит результат в отдельном окне от окна буфера и даёт полную строку где слова содержать(рис.22)

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

```
2 matches for "echo" in buffer: lab07.sh
```

```
5: echo $HELLO
```

```
7: echo $HELLO
```

```
U:%%- *0ccur*      All L1      (0ccur)
```

```
Searched 1 buffer; 2 matches for "echo"
```

Выполняя эту лабораторную работу познакомилась с операционной системой Linux и получила практические навыки работы с редактором Emacs