

Операционные системы

Лабораторная работа №9

Матюшкин Денис Владимирович (НПИбд-02-21)

Содержание

1	Цель работы	3
2	Ход работы	4
3	Контрольные вопросы	14
4	Вывод	16

1 Цель работы

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

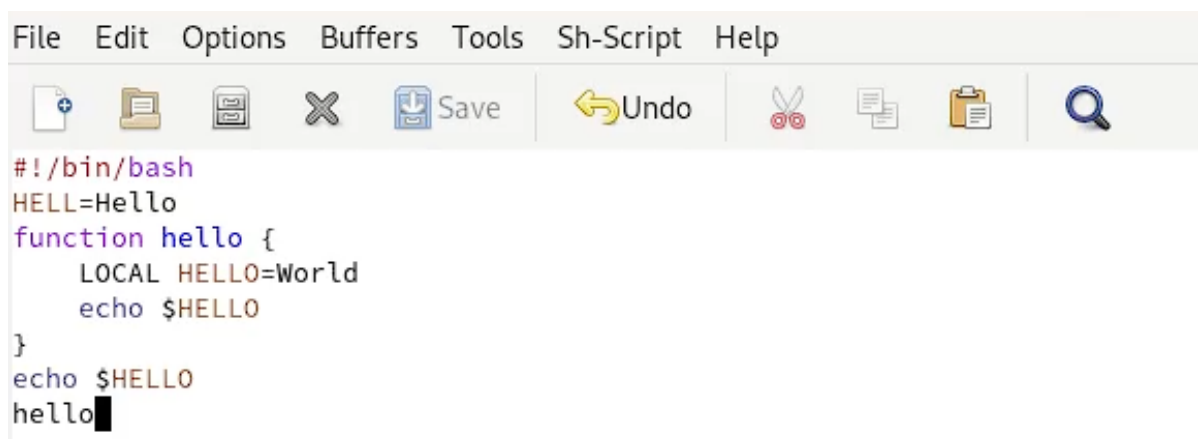
2 Ход работы

1. Откроем *emacs*, предварительно его скачав (рис. 2.1).

```
[dvmatyushkin@dvmatyushkin ~]$ emacs
```

Рис. 2.1: Запуск emacs

2. Создадим файл *lab07.sh* с помощью комбинации Ctrl-x Ctrl-f и наберем текст (рис. 2.2).



The screenshot shows the Emacs editor window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for file operations. The main text area contains the following shell script:

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

Рис. 2.2: Создание файла

3. Сохраним файл с помощью комбинации Ctrl-x Ctrl-s.
4. Прделаем с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш:
 - 4.1) Вырезаем одной командой целую строку (C-k) (рис. 2.3).

4.2) Вставим эту строку в конец файла (C-y) (рис. 2.3).

```
#!/bin/bash

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

Рис. 2.3: Вырезка и вставка строки

4.3) Выделим область текста (C-space) (рис. 2.4).

```
#!/bin/bash

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

Рис. 2.4: Выделение области текста

4.4) Скопируем область в буфер обмена (M-w) и вставим область в конец файла (рис. 2.5).

```
#!/bin/bash

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

Рис. 2.5: Копирование и вставка в конец файла

4.5) Вновь выделим эту область и на этот раз вырежем её (C-w) (рис. 2.6).

```
#!/bin/bash

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

Рис. 2.6: Вырезка выделенной области

4.6) Отменим последнее действие (C-/) (рис. 2.7).

```
#!/bin/bash

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

Рис. 2.7: Отмена действий

5. Научимся использовать команды по перемещению курсора:

5.1 Переместим курсор в начало строки (C-a) (рис. 2.8).

```
#!/bin/bash

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

Рис. 2.8: Курсор в начале строки

5.2 Переместим курсор в конец строки (C-e) (рис. 2.9).

```
#!/bin/bash

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

Рис. 2.9: Курсор в конце строки

5.3 Переместим курсор в начало буфера (M-<) (рис. 2.10).

```
#!/bin/bash

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

Рис. 2.10: Курсор в начале буфера

5.4 Переместим курсор в конец буфера (M->) (рис. 2.11).

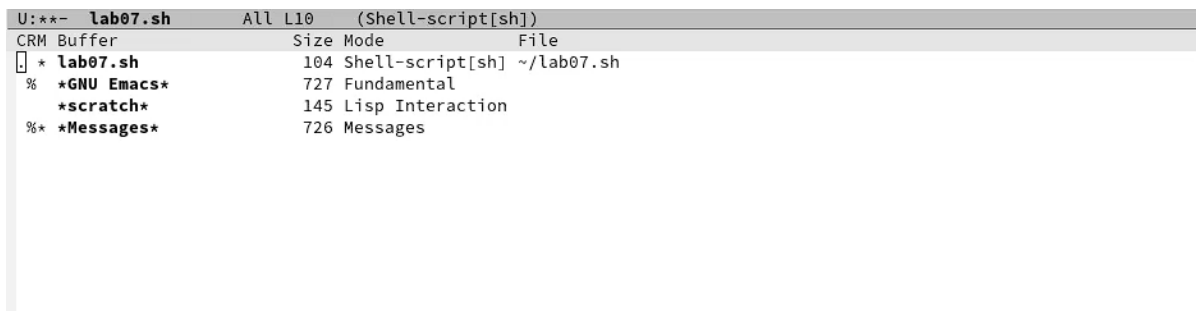
```
#!/bin/bash

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

Рис. 2.11: Курсор в конце буфера

6. Управление буферами:

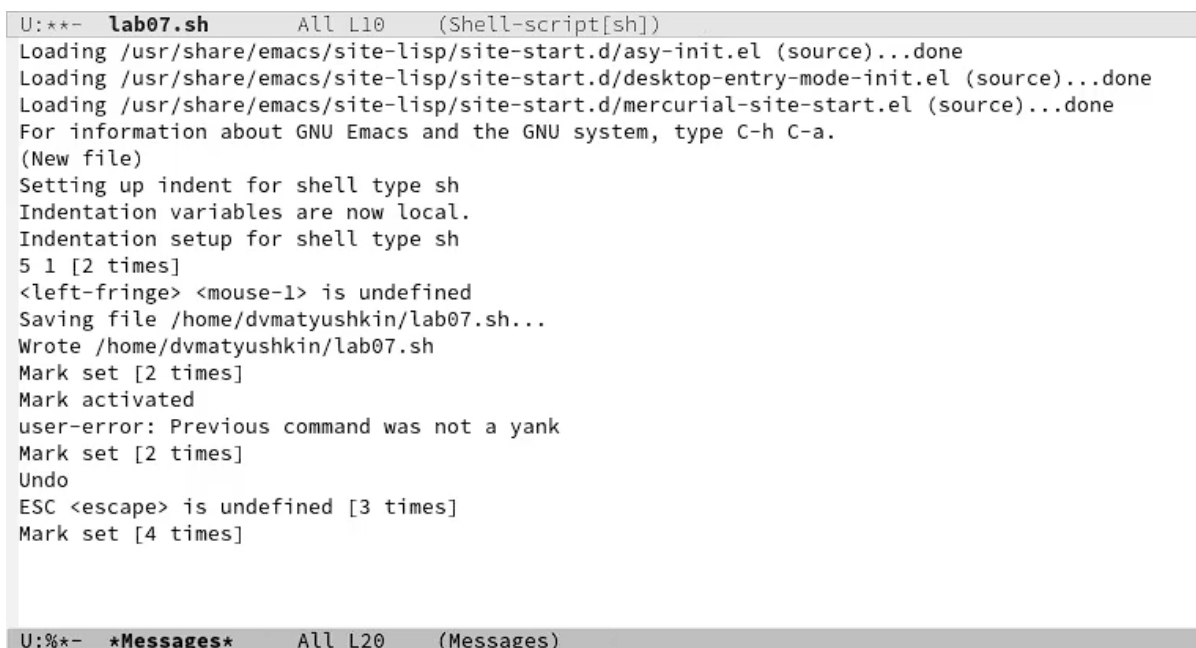
6.1 Выведем список активных буферов на экран (C-x C-b) (рис. 2.12).

A screenshot of the Emacs buffer list window. The title bar shows 'U:***- lab07.sh All L10 (Shell-script[sh])'. The table lists buffers: 'lab07.sh' (104 characters, Shell-script[sh]), '*GNU Emacs*' (727 characters, Fundamental), '*scratch*' (145 characters, Lisp Interaction), and '*Messages*' (726 characters, Messages).

CRM Buffer	Size	Mode	File
* lab07.sh	104	Shell-script[sh]	~/lab07.sh
% *GNU Emacs*	727	Fundamental	
scratch	145	Lisp Interaction	
%* *Messages*	726	Messages	

Рис. 2.12: Список активных буферов

6.2 Переместимся во вновь открытое окно (C-x) о со списком открытых буферов и переключимся на буфер *Messages* (рис. 2.13).

A screenshot of the Emacs Messages buffer. The title bar shows 'U:***- lab07.sh All L10 (Shell-script[sh])'. The buffer contains a log of Emacs startup messages, including loading site files, setting up indentation, and saving the file. The bottom status bar shows 'U:%*- *Messages* All L20 (Messages)'.

U:***- lab07.sh	All L10	(Shell-script[sh])
Loading /usr/share/emacs/site-lisp/site-start.d/asy-init.el (source)...done		
Loading /usr/share/emacs/site-lisp/site-start.d/desktop-entry-mode-init.el (source)...done		
Loading /usr/share/emacs/site-lisp/site-start.d/mercurial-site-start.el (source)...done		
For information about GNU Emacs and the GNU system, type C-h C-a.		
(New file)		
Setting up indent for shell type sh		
Indentation variables are now local.		
Indentation setup for shell type sh		
5 1 [2 times]		
<left-fringe> <mouse-1> is undefined		
Saving file /home/dvmatyushkin/lab07.sh...		
Wrote /home/dvmatyushkin/lab07.sh		
Mark set [2 times]		
Mark activated		
user-error: Previous command was not a yank		
Mark set [2 times]		
Undo		
ESC <escape> is undefined [3 times]		
Mark set [4 times]		

U:%*- *Messages*	All L20	(Messages)
------------------	---------	------------

Рис. 2.13: Перемещение между буферами

6.3 Закроем это окно (C-x 0).

6.4 Теперь вновь переключимся между буферами, но уже без вывода их списка на экран (C-x b). Откроем тот же буфер *Messages* (рис. 2.14).

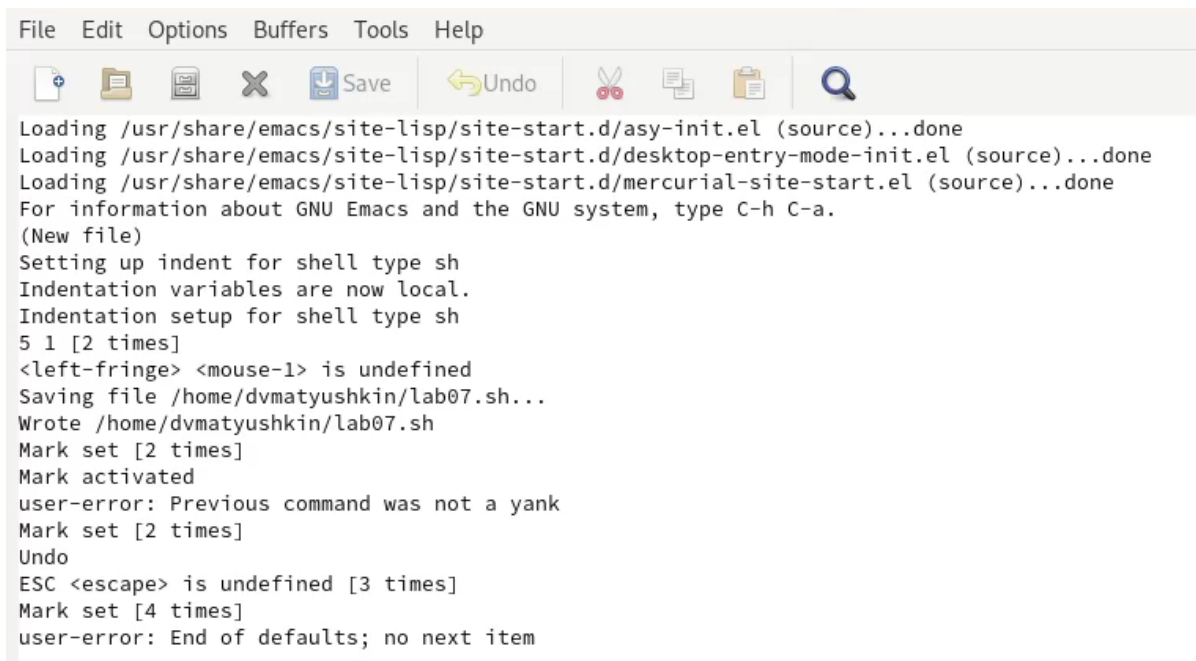


Рис. 2.14: Перемещение между буферами без вывода списка

7. Управление окнами:

7.1 Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2) (рис. 2.15).



Рис. 2.15: Фрейм emacs, разделенный на 4 окна

7.2 В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста (рис. 2.15).

8. Режим поиска:

8.1 Переключимся в режим поиска (C-s) и найдем несколько слов, присутствующих в тексте (рис. 2.16).

```
#!/bin/bash

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

Рис. 2.16: Режим поиска

8.2 Переключаемся между результатами поиска, нажимая C-s.

8.3 Выйдем из режима поиска, нажав C-g.

8.4 Перейдем в режим поиска и замены (M-%), введем текст, который следует найти и заменить, нажмем Enter , затем введем текст для замены. После того как будут подсвечены результаты поиска, нажмем ! для подтверждения замены (рис. 2.17 и рис. 2.18).

```
#!/bin/bash

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

U:**~ **lab07.sh** All L3 (Shell-script[sh])

```
int main() {
    cout<<"Hello world!";
    return 0;
}
```

U:**~ **text2** All L2 (Fundamental)

Query replacing hello with hi: (? for help)

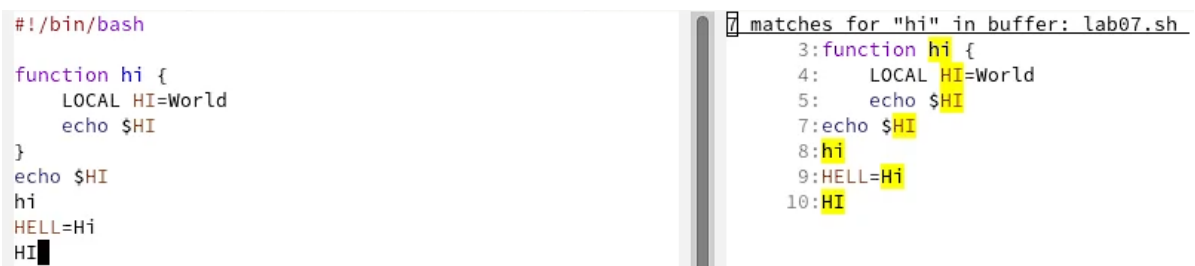
Рис. 2.17: Замена слов

```
#!/bin/bash

function hi {
    LOCAL HI=World
    echo $HI
}
echo $HI
hi
HELL=Hi
HI
```

Рис. 2.18: Замена слов

8.5 Испробуем другой режим поиска, нажав *M-s o* (рис. 2.19). В отличие от обычного режима создается отдельное окно со строками, где есть слово, которое мы ввели для поиска. Также можно быстро переходить по строкам через клик.



```
#!/bin/bash

function hi {
    LOCAL HI=World
    echo $HI
}
echo $HI
hi
HELL=Hi
HI
```

7 matches for "hi" in buffer: lab07.sh

```
3: function hi {
4:     LOCAL HI=World
5:     echo $HI
7: echo $HI
8: hi
9: HELL=Hi
10: HI
```

Рис. 2.19: Другой режим поиска

3 Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

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

2. Какие особенности данного редактора могут сделать его сложным для освоения новичком?

Для новичка может быть трудно сразу привыкнуть к тому, что в редакторе в основном используются комбинации клавиш.

3. Своими словами опишите, что такое буфер и окно в терминологии emacs'а.

Буфер - объект, который представляет какой-то текст.

Окно - прямоугольная область фрейма, отображающая один из буферов.

4. Можно ли открыть больше 10 буферов в одном окне?

Да, можно.

5. Какие буферы создаются по умолчанию при запуске emacs?

Буферы по умолчанию: GNU Emacs, scratch, Messages, Compile-Log, Quail Completions

6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?

Ответ: ctrl-f, ctrl-x

7. Как поделить текущее окно на две части?

Ответ: ctrl-x 3

8. В каком файле хранятся настройки редактора emacs?

Ответ: (~/.emacs)

9. Какую функцию выполняет клавиша (<-) и можно ли её переназначить?

Удаление текста. Да, можно, но для новичка это может показаться трудным.

10. Какой редактор вам показался удобнее в работе vi или emacs? Поясните почему.

Emacs показался мне удобнее. Он более современен и функционален. Так же он гораздо более удобен в использовании. Нет случайных удалений текста как было с vi (из-за незнания). Интерфейс emacs приятнее и интуитивнее, команды более дружелюбны к пользователю.

4 Вывод

- В ходе этой лабораторной работы мы познакомились с операционной системой Linux. Получили практические навыки работы с редактором Emacs.