

Лабораторная работа №10

Бондарь Алексей Олегович НПМбд-01-20¹

10 мая, 2021, Москва, Россия

¹RUDN University, Moscow, Russian Federation

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

Выполнение лабораторной работы

1. Откроем редактор Emacsc помощью команды «emacs&». (рис. 1)



Figure 1: Emacsc

- Создадим файл lab10.sh с помощью комбинации «Ctrl-x» «Ctrl-f».(рис. 2)

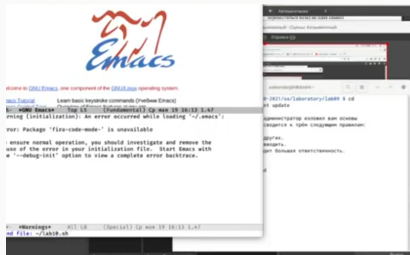


Figure 2: Файл lab07.sh

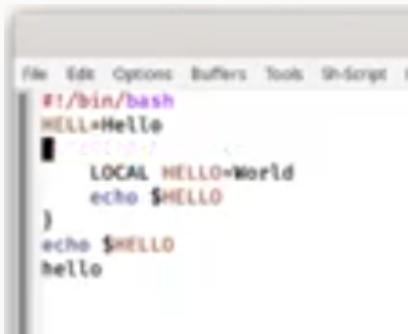
3. В открывшемся буфере наберем необходимый текст (рис. 3)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'SH-Script', and 'Help'. The terminal displays a Bash script with the following content:

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

Figure 3: Текст

Сохраним файл с помощью комбинации «Ctrl-x»«Ctrl-s». 4. 1) Вырежем одной командой целую строку («Ctrl-k»).(рис. 4)



```
File Edit Options Buffers Tools Sh-Script H
#!/bin/bash
HELL=Helle
f() {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
helle
```

Figure 4: Ctrl-k

2) Вставим эту строку в конец файла («Ctrl-y»).(рис. 5)



```
File Edit Options Buffers Tools Shell-Script
#!/bin/bash
HELLO=Hello

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

Figure 5: Ctrl-y

3) Выделим область текста («Ctrl-space»).(рис. 6)



The image shows a terminal window with a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a text area containing a Bash script. The script is as follows:

```
#!/bin/bash
HELL=Hello

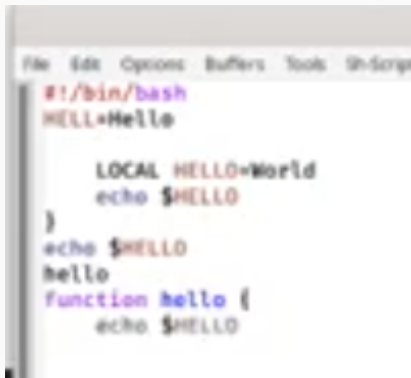
    LOCAL HELLO=World
    echo $HELLO
)
echo $HELLO
hello
function hello {
```

The text is color-coded: blue for shebangs and function definitions, red for variable assignments, and black for other text. A black cursor is positioned at the end of the line `echo $HELLO`. A small black square, representing a Ctrl-space character, is located at the end of the line `echo $HELLO`.

Figure 6: Ctrl-space

Выполнение лабораторной работы

- 4) Скопируем область в буфер обмена («Alt-w»).
- 5) Вставим область в конец файла («Ctrl-y»).(рис. 7)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Shell-Script'. The terminal displays a script with the following content:

```
#!/bin/bash
HELL=Hello

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

Figure 7: Ctrl-y

- 6) Вновь выделим эту область («Ctrl-space») и на этот раз вырежем её («Ctrl-w»).(рис. 8)

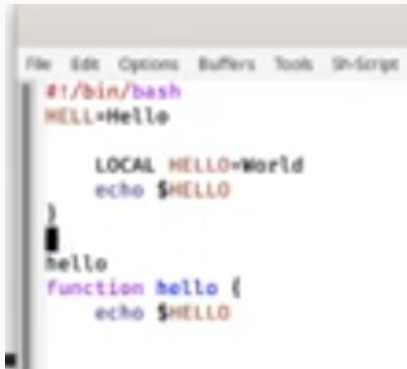


```
File Edit Options Buffers Tools Sh-script Help
#!/bin/bash
HELLO=Hello

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

Figure 8: Ctrl-w

7) Отменим последнее действие («Ctrl-/»).(рис. 9)



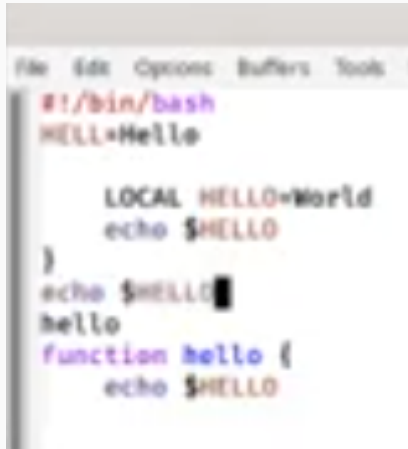
```
File Edit Options Buffers Tools SH-script
#!/bin/bash
HELL=Hello

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

Figure 9: Ctrl-/

5.

1) Переместим курсор в начало строки («Ctrl-a»).(рис. 10) и (рис. 11)

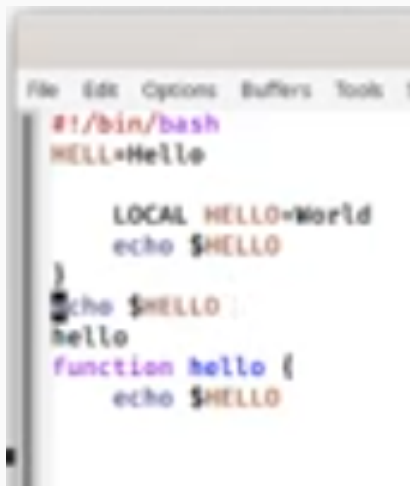
A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', and 'Tools'. The terminal displays a script with the following content:

```
#!/bin/bash
HELL=Hello

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

 A black cursor is positioned at the start of the line containing 'echo \$HELLO'.

Figure 10: Ctrl-a

A screenshot of a terminal window with a menu bar (File, Edit, Options, Buffers, Tools, ?) and a light gray background. The terminal shows a bash script with the following content:

```
#!/bin/bash
HELLO=Hello

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

 A cursor is positioned at the end of the line 'echo \$HELLO', and the text 'hello' has been inserted at that position, demonstrating the effect of the Ctrl+A command.

Figure 11: Ctrl a

2) Переместим курсор в конец строки («Ctrl-e»).(рис. 12)

A screenshot of a terminal window with a menu bar at the top containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Shell-script'. The terminal displays a script with the following content:

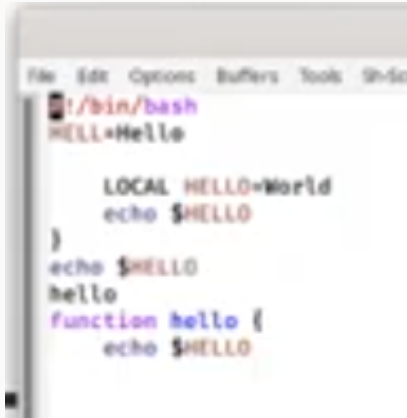
```
#!/bin/bash
HELL=Hello

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

 The cursor is positioned at the end of the line 'echo \$HELLO'.

Figure 12: Ctrl-e

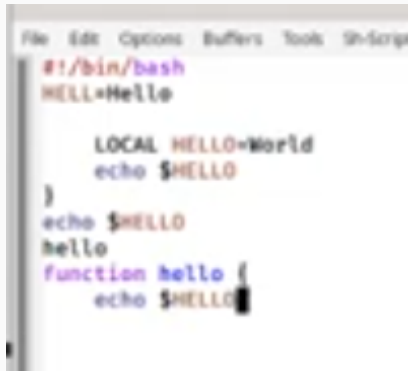
3) Переместим курсор в началобуфера («Alt-<»).(рис. 13)



```
File Edit Options Buffers Tools Shell  
#!/bin/bash  
HELL=Hello  
  
    LOCAL HELLO=World  
    echo $HELLO  
}  
echo $HELLO  
hello  
function hello {  
    echo $HELLO
```

Figure 13: Alt-<

4) Переместим курсор в конец буфера («Alt->»). (рис. 14)

A screenshot of a terminal window with a menu bar containing 'File', 'Edit', 'Options', 'Buffers', 'Tools', and 'Sh-script'. The terminal displays a bash script with the following content:

```
#!/bin/bash
HELL=Hello

    LOCAL HELLO=World
    echo $HELLO
)
echo $HELLO
hello
function hello {
    echo $HELLO
```

 A black cursor is positioned at the end of the last line, 'echo \$HELLO'.

Figure 14: Alt->

Выполнение лабораторной работы

6.

- 1) Выведем список активных буферов на экран («Ctrl-x»«Ctrl-b»).(рис. 15)



The screenshot shows the Emacs editor window titled 'emacs@08464'. The main editing area contains a shell script with the following content:

```
#!/bin/bash
HELLO=Hello

LOCAL_HELLO=World
echo $HELLO
}
echo $HELLO
hello
function hello {
  echo $HELLO
}
```

Below the editing area, the command line shows 'U++- lab10.sh All List (Shell-script[sh]) Cp max 19 16'. Below the command line, a table lists the active buffers:

Buffer	Size	Mode	File
* lab10.sh	114	Shell-script[sh]	~/lab10.sh
* GNU Emacs	723	Fundamental	
* scratch	145	Lisp Interaction	
* Messages	31159	Messages	
* Compile-Log	257	elisp-compile	
* Warnings	314	Special	
* Quail Completions	@	Fundamental	

Figure 15: «Ctrl-x»«Ctrl-b»

- 2) Переместимся во вновь открытое окно («Ctrl-x о») со списком открытых буферов и переключимся на другой буфер (для этого необходимо нажать на «enter» после выбора необходимого буфера) .(рис. 16)

The image shows the Emacs buffer list window, which appears after pressing Ctrl-x o. The title bar at the top says "emacs@d6b6d4". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", and "Help". The main area lists several buffers: a shell buffer with the prompt "#!/bin/bash" and the text "HELL=Hello"; a local buffer named "LOCAL_HELLO=World" with the text "echo \$HELLO"; a buffer named "echo \$HELLO" with the text "hello"; and a function buffer named "function hello {" with the text "echo \$HELLO". At the bottom, a status bar shows "U++: tab10.sh All L10 (Shell-script[sh]) Co Mar 19 16:18". Below the buffer list, a scrollback area displays a series of messages: "Mark set [2 times]", "Mark deactivated", "Mark activated", "funcall-interactively: End of buffer [10 times]", "Mark set [2 times]", "Undo", "user-error: Marker stack is empty [2 times]", and "Mark set [2 times]".

```
emacs@d6b6d4
File Edit Options Buffers Tools Help

#!/bin/bash
HELL=Hello

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

U++: tab10.sh All L10 (Shell-script[sh]) Co Mar 19 16:18
Mark set [2 times]
Mark deactivated
Mark activated
funcall-interactively: End of buffer [10 times]
Mark set [2 times]
Undo
user-error: Marker stack is empty [2 times]
Mark set [2 times]
```

Figure 16: Выбор буфера

3) Закроем это окно («Ctrl-x 0»).(рис. 17)

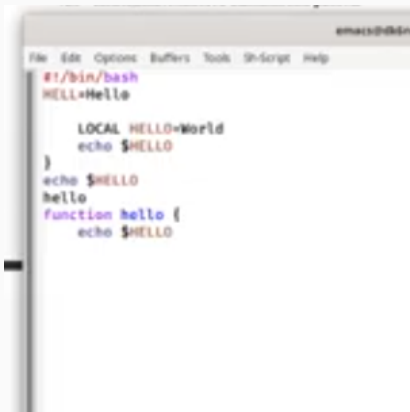


Figure 17: Ctrl-x 0

- 4) Теперь вновь переключимся между буферами, но уже без вывода их списка на экран («Ctrl-x b»).(рис. 18) и (рис. 19)



Figure 18: Переключение



Figure 19: Другой буфер

7.

- 1) Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали («Ctrl-x 3»), а затем каждое из этих окон на две части по горизонтали («Ctrl-x 2»).(рис. 20) , (рис. 21) , (рис. 22)

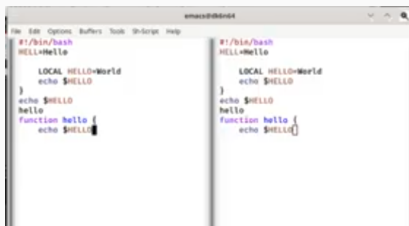


Figure 20: Два окна

Выполнение лабораторной работы



Figure 21: Три окна

Выполнение лабораторной работы

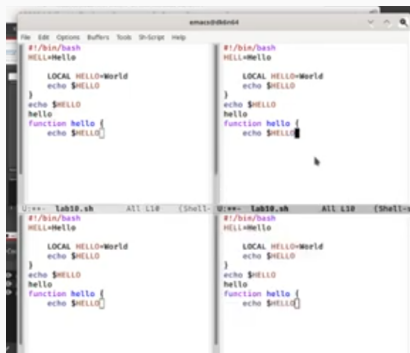
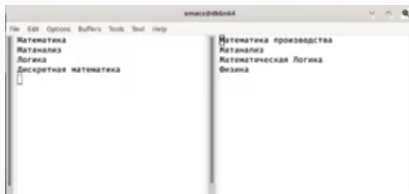


Figure 22: Четыре окна

- 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 example1.txt», «touch example2.txt», «touch example3.txt», «touch example4.txt». (рис. 23) и (рис. 24)

```
labondarj@k6n64 ~$ touch example1.txt
labondarj@k6n64 ~$ touch example2.txt
labondarj@k6n64 ~$ touch example3.txt
labondarj@k6n64 ~$ touch example4.txt
labondarj@k6n64 ~$
```

Figure 23: Создание файлов



8.

- 1) Переключимся в режим поиска («Ctrl-s») и найдем несколько слов, присутствующих в тексте.(рис. 25)



Figure 25: Режим поиска

Выполнение лабораторной работы

- 2) Переключимся между результатами поиска, нажимая «Ctrl-s».
- 3) Выйдем из режима поиска, нажав «Ctrl-g»
- 4) Перейдем в режим поиска и замены («Alt-%»), 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. 26) и (рис. 27)

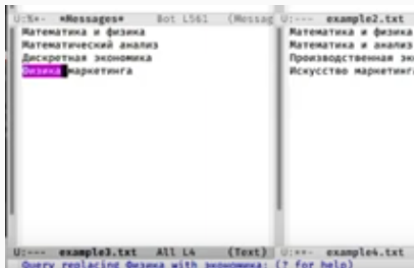


Figure 26: Замена результата - начальное значение

Выполнение лабораторной работы



Figure 27: Замена результата - конечное значение

- 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.

В ходе выполнения данной лабораторной работы я познакомился с операционной системой Linux и получил практические навыки работы с редактором Emacs.