

Текстовой редактор emacs

Доленко Дарья Васильевна НБИбд-01-21¹

19 мая, 2022, Москва, Россия

¹Российский Университет Дружбы Народов

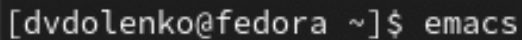
Цели и задачи работы

Цель лабораторной работы

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

Процесс выполнения лабораторной работы

Открываю emacs.(рис. 1)

A terminal window with a dark background. The prompt is [dvdolenko@fedora ~]\$ and the command emacs has been entered.

```
[dvdolenko@fedora ~]$ emacs
```

Figure 1: открытие emacs

Создаю файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f. Набираю приведенный текст. Сохраняю файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s). (рис. 2)

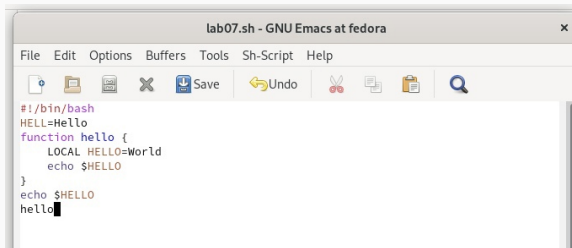


Figure 2: Создание файла, набор текста, сохранение файла

**Проделываю с текстом
стандартные процедуры
редактирования, каждое
действие должно осуществляться
комбинацией клавиш.**

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



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

The image shows a terminal window with a light gray background. The first four lines of the script are color-coded: the shebang is red, the variable assignment is brown, the function keyword is purple, and the function name is blue. The function body contains a local variable assignment (brown) and an echo command (blue). After the closing brace of the function, there is a solid black rectangular block. The final line of the script, 'hello', is not color-coded.

Figure 3: Вырезание строки

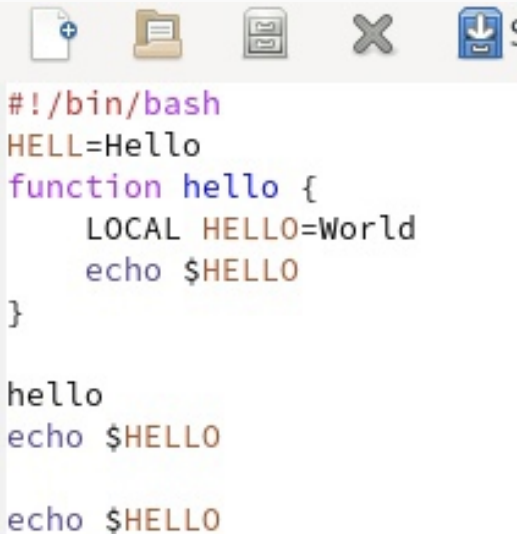
Вставляю эту строку в конец файла (С-у). (рис. 4)

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

hello
echo $HELLO
```

Figure 4: Вставка строки в конец файла

Выделяю область текста (C-space). Копирую область в буфер обмена (M-w). Вставляю область в конец файла.(рис. 5)



The image shows a terminal window with a light gray title bar containing icons for file operations (add, folder, save, close, download). The terminal content is as follows:

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

hello
echo $HELLO

echo $HELLO
```

Выделяю эту область и на этот раз вырезаю её (C-w), отменяю последнее действие (C-/).

**Учучь использовать команды по
перемещению курсора**

Перемещаю курсор в конец строки (C-e). (рис. 6)



The image shows a terminal window with a light gray title bar containing icons for file operations (add, folder, save, close, download) and a partial 'Sa' label. The terminal content is as follows:

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

hello
echo $HELLO

echo $HELLO
```

A black cursor is positioned at the end of the line containing the function definition: `function hello {`.

Figure 6: Курсор в конце строки

Перемещаю курсор в конец строки (C-a). (рис. 7)



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

hello
echo $HELLO
```

The image shows a terminal window with a Bash script. The script consists of several lines: a shebang line, a variable assignment, a function definition, and two calls to the function. The cursor is positioned at the beginning of the second line, before the variable assignment.

Figure 7: Курсор в начале строки

Затем перемещаю курсор в начало буфера (M-<), перемещаю курсор в конец буфера (M->).

Управляю буферами: вывожу список активных буферов на экран (C-x C-b) (рис. 8)

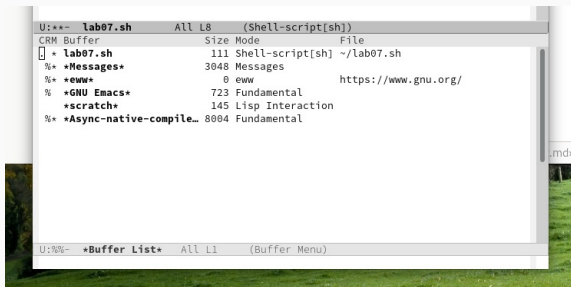


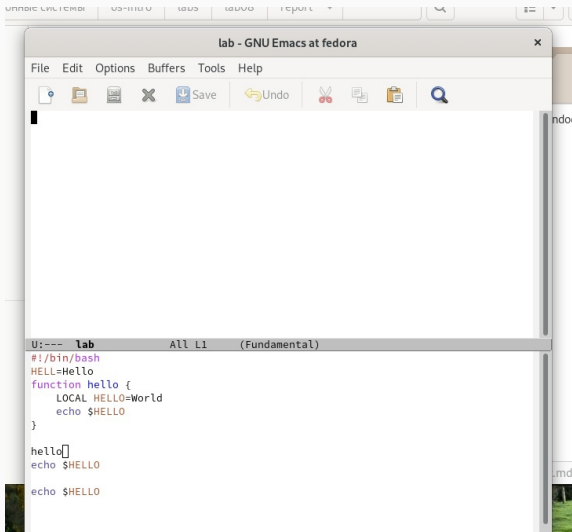
Figure 8: Список активных буферов

**Перемещаюсь во вновь открытое
окно (С-х) о со списком
открытых буферов и
перемещаюсь на другой буфер.**

Закрываю это окно (С-х 0).

Теперь вновь переключаюсь между буферами, но уже без вывода их списка на

экран (C-x b). (рис. 9)



Делю фрейм на 4 части: на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2) В каждом из четырёх созданных окон открываю новый буфер (файл) и ввожу

несколько строк текста. (рис. 10)



The screenshot shows the GNU Emacs editor interface with a 2x2 grid of buffers. The title bar at the top reads "lab07.sh - GNU Emacs at fedora". The menu bar includes "File", "Edit", "Options", "Buffers", "Tools", "Sh-Script", and "Help". The toolbar contains icons for opening files, saving, undo, redo, and search. Each of the four buffers displays the following shell script code:

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

hello
echo $HELLO

echo $HELLO
```

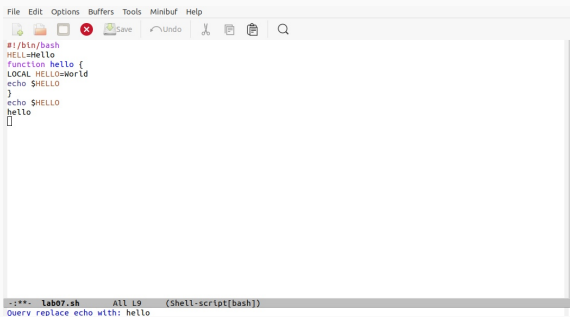
Режим поиска:

Переключаюсь в режим поиска (C-s) и нахожу несколько слов, присутствующих в тексте. Переключаюсь между результатами поиска, нажимая C-s. Выхожу нажав C-g. (рис. 11)

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

hello
echo $HELLO
echo $HELLO
```

Перехожу в режим поиска и замены (M-%), ввожу текст, который следует найти и заменить, нажимаю Enter , затем введите текст для замены. После того как будут подсвечены результаты поиска, нажимаю ! для подтверждения замены. (рис. 12)



The screenshot shows a Vim editor window with a menu bar (File, Edit, Options, Buffers, Tools, Minibuf, Help) and a toolbar. The main text area contains a shell script snippet:

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

The word "HELLO" in the function definition is highlighted in red. The status bar at the bottom shows the file name "Lab07.sh", line "All L9", and the command "(Shell-script[bash])". Below the status bar, a command prompt shows the command "Ouvr replace echo wth: hello".

Figure 12: Поиск и замена слов

Пробую другой режим поиска, нажав M-s o. (рис. 13)

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


Выводы по проделанной работе

В ходе данной лабораторной работы я ознакомилась с операционной системой Linux. Получила практические навыки работы с редактором Emacs.