

Редактор Emacs

Олейников Артём Игоревич¹

16 июня, 2024, 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.

Задачи лабораторной работы

1 Изучить возможности редактора Emacs

Процесс выполнения лабораторной работы

Выполнение работы

1. Откроем Emacs.

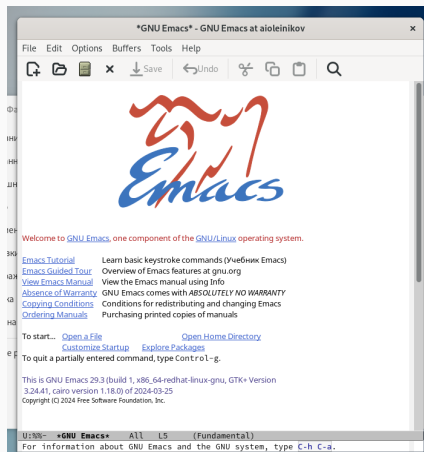


Рис. 1: Запуск Emacs

Выполнение работы

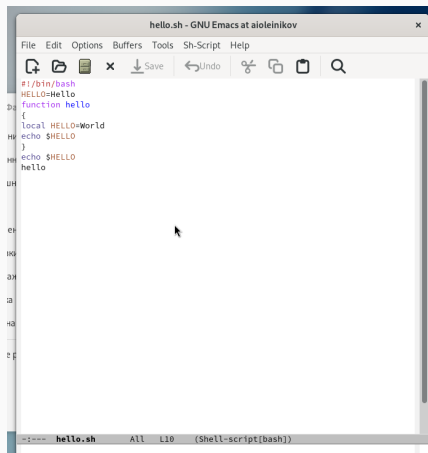
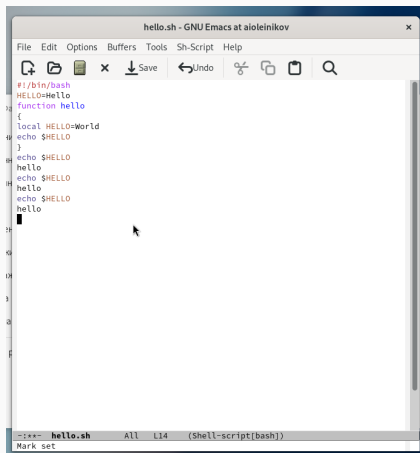


Рис. 2: Новый файл

Выполнение работы



The image shows a GNU Emacs editor window titled "hello.sh - GNU Emacs at aioleinikov". The window displays a shell script with the following content:

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

The script defines a function named "hello" that sets a local variable "HELLO" to "World" and prints it. The function is called multiple times, each time printing the value of "HELLO". The editor interface includes a menu bar (File, Edit, Options, Buffers, Tools, Sh-Script, Help) and a toolbar with icons for file operations, undo, redo, and search. The status bar at the bottom shows the current file is "hello.sh", the buffer is "All", the line is "L14", and the mode is "(Shell-script(bash))".

Рис. 3: Операция вставка

Выполнение работы

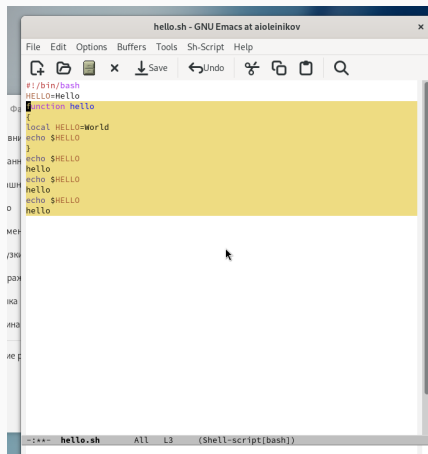
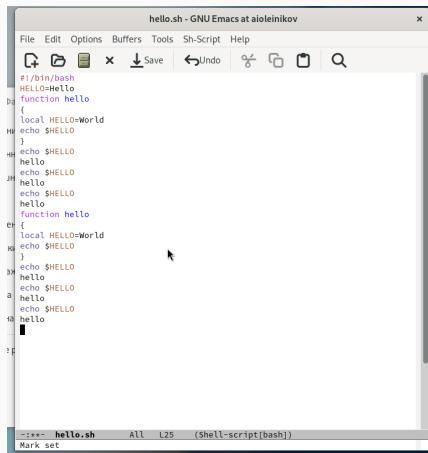


Рис. 4: Выделение блока

Выполнение работы



```
hello.sh - GNU Emacs at aioleinikov
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Close, Undo, Cut, Copy, Paste, Find]

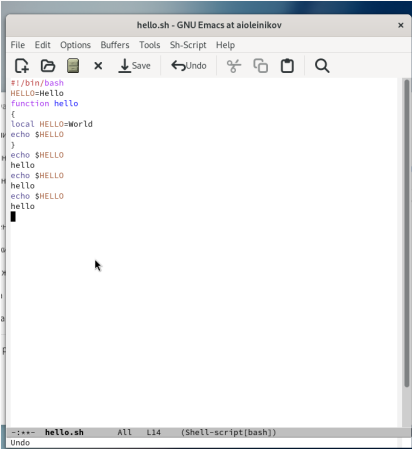
#!/bin/bash
HELLO=Hello
function hello
{
  local HELLO=World
  echo $HELLO
}
echo $HELLO
hello
echo $HELLO
hello
echo $HELLO
hello
function hello
{
  local HELLO=World
  echo $HELLO
}
echo $HELLO
hello
echo $HELLO
hello
echo $HELLO
hello

```

--- hello.sh All L25 (Shell-script[bash])
Mark set

Рис. 5: Копирование блока

Выполнение работы



```
hello.sh - GNU Emacs at aiolelnikov
File Edit Options Buffers Tools Sh-Script Help
[Icons: New, Open, Save, Close, Undo, Cut, Copy, Paste, Find]

#!/bin/bash
HELLO=Hello
function hello
{
  local HELLO=World
  echo $HELLO
}
echo $HELLO
hello
echo $HELLO
hello
echo $HELLO
hello

--:-- hello.sh All L14 (shell-script)(bash)
Undo
```

Рис. 6: Удаление блока

Выполнение работы

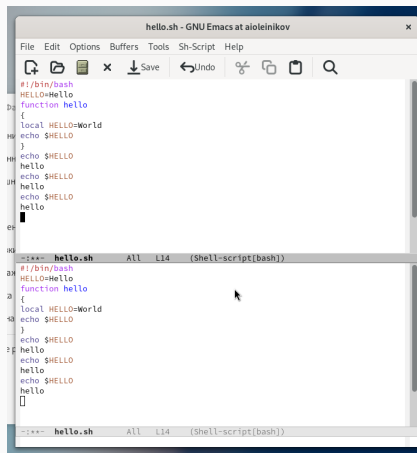
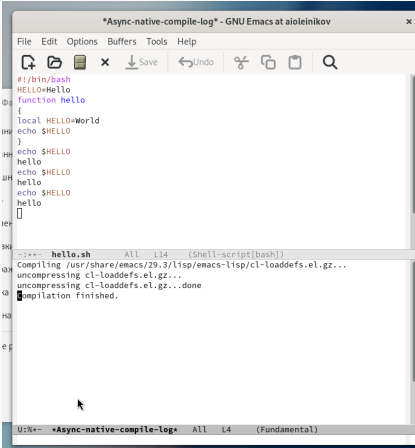


Рис. 7: Горизонтальное разделение

Выполнение работы



The screenshot shows the GNU Emacs editor window titled "*Async-native-compile-log* - GNU Emacs at aioleinikov". The menu bar includes File, Edit, Options, Buffers, Tools, and Help. The toolbar contains icons for opening, saving, undo, redo, and search. The main text area displays a shell script named "hello.sh" with the following content:

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

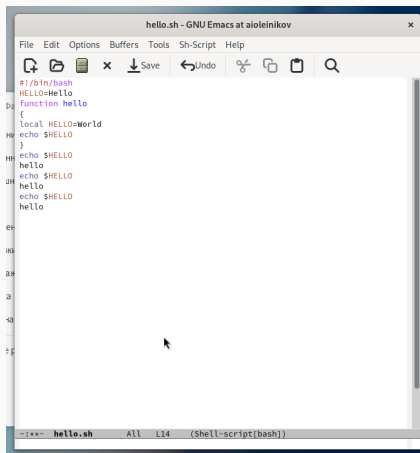
Below the script, the output of the compilation process is shown:

```
---+--- hello.sh All L1d (Shell-script(bash))
Compiling /usr/share/emacs/29.3/lisp/emacs-lisp/cl-loaddefs.el.gz...
uncompressing cl-loaddefs.el.gz...done
uncompressing cl-loaddefs.el.gz...done
Compilation finished.
```

The status bar at the bottom indicates the current buffer is "*Async-native-compile-log*", and the mode is "(Fundamental)".

Рис. 8: Переключение буфера

Выполнение работы



The image shows a screenshot of the GNU Emacs editor window titled "hello.sh - GNU Emacs at aioleinikov". The window displays a shell script with the following content:

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

The script defines a function named "hello" that sets a local variable "HELLO" to "World" and prints the value of "HELLO". The function is then called multiple times, each time printing the value of "HELLO" (which is "World" inside the function and "Hello" outside). The status bar at the bottom of the window shows "hello.sh", "All", "L14", and "(Shell-script(bash))".

Рис. 9: Закроем буфер

Выполнение работы

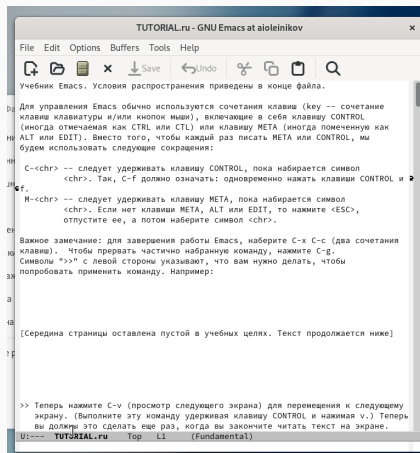


Рис. 10: Переключение буфера

Выполнение работы

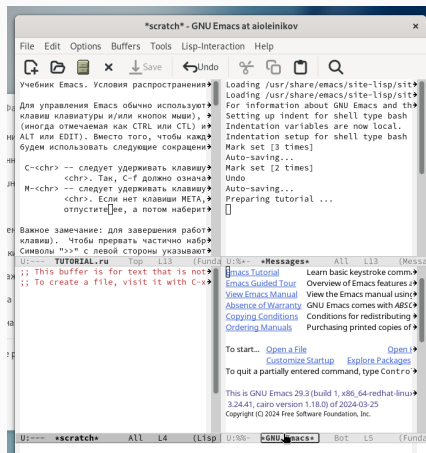


Рис. 11: Четыре буфера

Выполнение работы

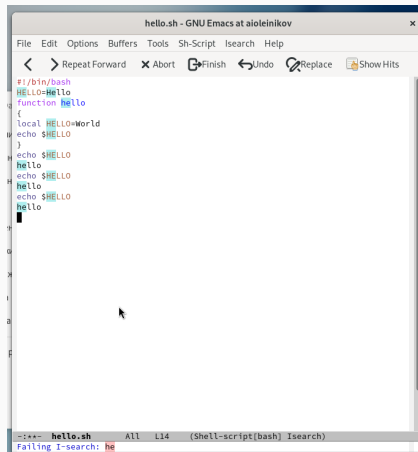


Рис. 12: Режим поиска

Выводы по проделанной работе

В данной работе мы познакомились с еще одним редактором операционной системой Linux. Получили практические навыки работы с редактором Emacs.