

Отчёт по лабораторной работе №9

Текстовый редактор Emacs

Дупленских Василий Викторович

Содержание

Цель работы:	6
Выполнение лабораторной работы:	7
1. Открываю emacs:	7
2. Создаю файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f:	8
3. Набираю текст:	8
4. Сохраняю файл с помощью комбинации Ctrl-x Ctrl-s:	9
5. Прodelываю с текстом стандартные процедуры редактирования, каждое действие осуществляю комбинацией клавиш:	9
5.1 Вырезаю одной командой целую строку:	9
5.2. Вставляю эту строку в конец файла:	10
5.3. Выделяю область текста:	10
5.4. Копирую область в буфер обмена:	11
5.5. Вставляю область в конец файла:	11
5.6. Вновь выделяю эту область и на этот раз вырезаю её:	12
5.7. Отменяю последнее действие:	12
6. Использую команды по перемещению курсора:	13
6.1. Перемещаю курсор в начало строки:	13
6.2. Перемещаю курсор в конец строки:	13
6.3. Перемещаю курсор в начало буфера:	14
6.4. Перемещаю курсор в конец буфера:	15
7. Управляю буферами:	16
7.1. Вывожу список активных буферов на экран:	16
7.2. Перемещаюсь во вновь открытое окно со списком открытых буферов	17
7.3. Закрываю это окно:	18
7.4. Теперь вновь переключаюсь между буферами, но уже без вывода их списка на экран:	19
8. Управляю окнами:	20
8.1. Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали,	20
8.2. В каждом из четырёх созданных окон открываю новый буфер (файл) и ввожу несколько строк текста:	21
9. Режим поиска:	22
9.1. Переключаюсь в режим поиска и ищу несколько слов, присутствующих в тексте:	22
9.2. Переключаюсь между результатами поиска:	23
9.3. Выхожу из режима поиска:	24

9.4. Перехожу в режим поиска и замены, ввожу текст, который следует найти	24
9.5. Отprobываю другой режим поиска:	26
Ответы на вопросы:	27
Вывод:	29

Список иллюстраций

0.1	Открытие ешас	7
0.2	Создание файла	8
0.3	Текст	8
0.4	Сохранение	9
0.5	Вырез	9
0.6	Вставка	10
0.7	Выделение	10
0.8	Вставка	11
0.9	Вставка 2	11
0.10	Выделение + вырез	12
0.11	Отмена	12
0.12	Начало строки	13
0.13	Конец строки	13
0.14	Начало буфера	14
0.15	Конец буфера	15
0.16	Список активных буферов	16
0.17	Закрытие	18
0.18	Переключение без вывода	19
0.19	Новый буфер + ввод	21
0.20	Режим поиска	22
0.21	Переключение	23
0.22	Выход из режима Поиска	24
0.23	Другой режим поиска	26

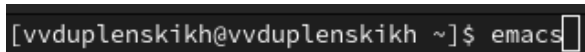
Список таблиц

Цель работы:

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

Выполнение лабораторной работы:

1. Открываю emacs:

A terminal window with a dark background. The prompt is [vvduplenskikh@vvduplenskikh ~]\$ and the command emacs is entered, followed by a cursor.

```
[vvduplenskikh@vvduplenskikh ~]$ emacs
```

Рис. 0.1: Открытие emacs

2. Создаю файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f:

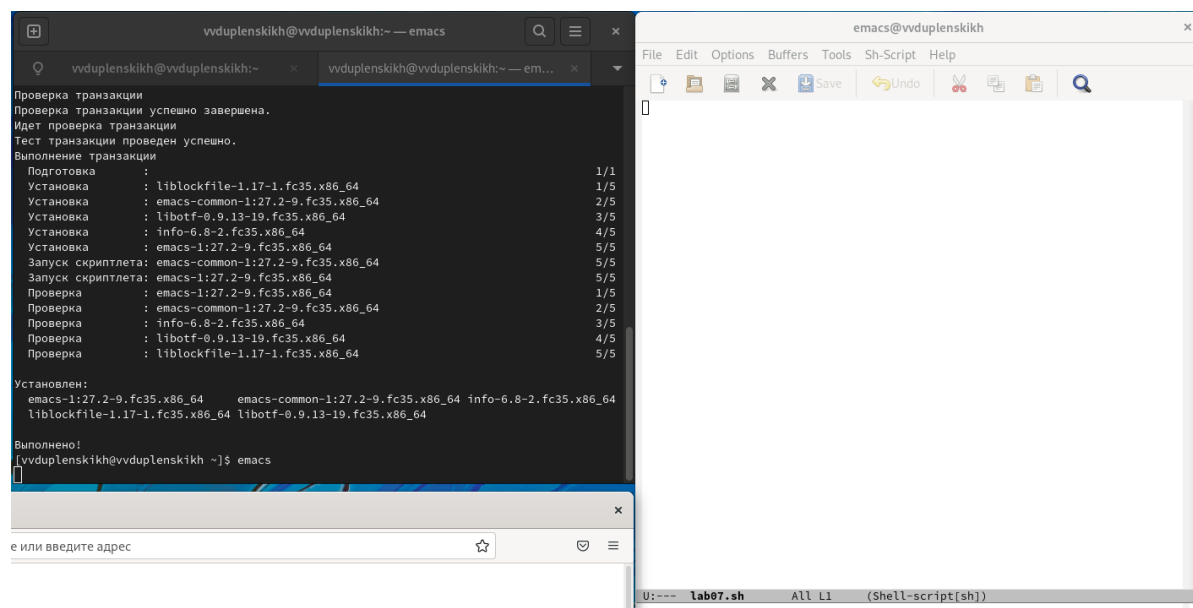


Рис. 0.2: Создание файла

3. Набираю текст:

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

Рис. 0.3: Текст

4. Сохраняю файл с помощью комбинации Ctrl-x Ctrl-s:

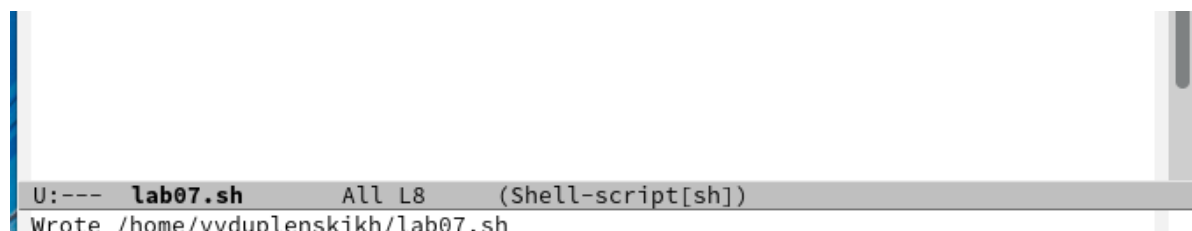


Рис. 0.4: Сохранение

5. Прodelываю с текстом стандартные процедуры редактирования, каждое действие осуществляю комбинацией клавиш.

5.1 Вырезаю одной командой целую строку:

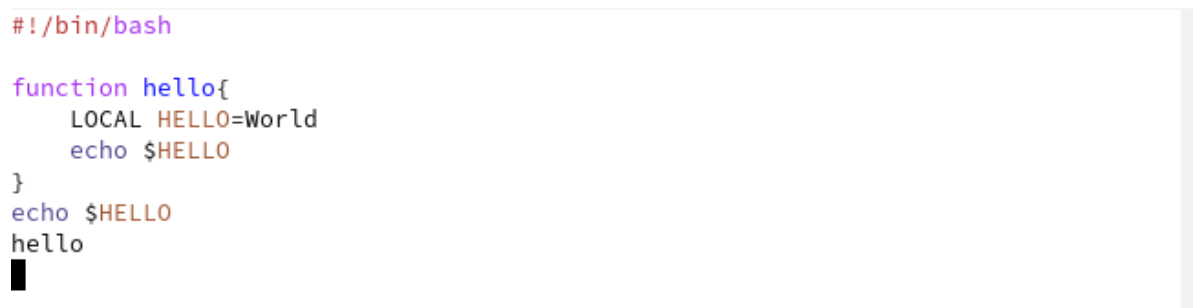


Рис. 0.5: Вырез

5.2. Вставляю эту строку в конец файла:




```
#!/bin/bash

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

Рис. 0.6: Вставка

5.3. Выделяю область текста:



```
#!/bin/bash

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

Рис. 0.7: Выделение

5.4. Копирую область в буфер обмена:

```
#!/bin/bash

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

Рис. 0.8: Вставка

5.5. Вставляю область в конец файла:

```
#!/bin/bash

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

Рис. 0.9: Вставка 2

5.6. Вновь выделяю эту область и на этот раз вырезаю её:

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

Рис. 0.10: Выделение + вырез

5.7. Отменяю последнее действие:

```
#!/bin/bash

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

Рис. 0.11: Отмена

6. Используя команды по перемещению курсора:

6.1. Перемещаю курсор в начало строки:



```
#!/bin/bash

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

The screenshot shows a terminal window with a Bash script. The script defines a function 'hello' that sets a local variable 'HELLO' to 'World' and echoes it. It then calls the function and prints the value of 'HELLO'. After a newline, it sets 'HELL=Hello' and redefines the 'hello' function to echo 'HELLO'. The cursor is positioned at the start of the line containing 'HELL=Hello'.

Рис. 0.12: Начало строки

6.2. Перемещаю курсор в конец строки:



```
#!/bin/bash

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

The screenshot shows the same terminal window and script as in Figure 0.12. The cursor is now positioned at the end of the line containing 'HELL=Hello'.

Рис. 0.13: Конец строки

6.3. Перемещаю курсор в начало буфера:

```
#!/bin/bash

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

U:***- lab07.sh All L9 (Shell-script[sh])
Visit tags table (default TAGS): ~/

Рис. 0.14: Начало буфера

6.4. Перемещаю курсор в конец буфера:

```
#!/bin/bash

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

U:***- lab07.sh All L9 (Shell-script[sh])
Command attempted to use minibuffer while in minibuffer

Рис. 0.15: Конец буфера

7. Управляю буферами:

7.1. Вывожу список активных буферов на экран:

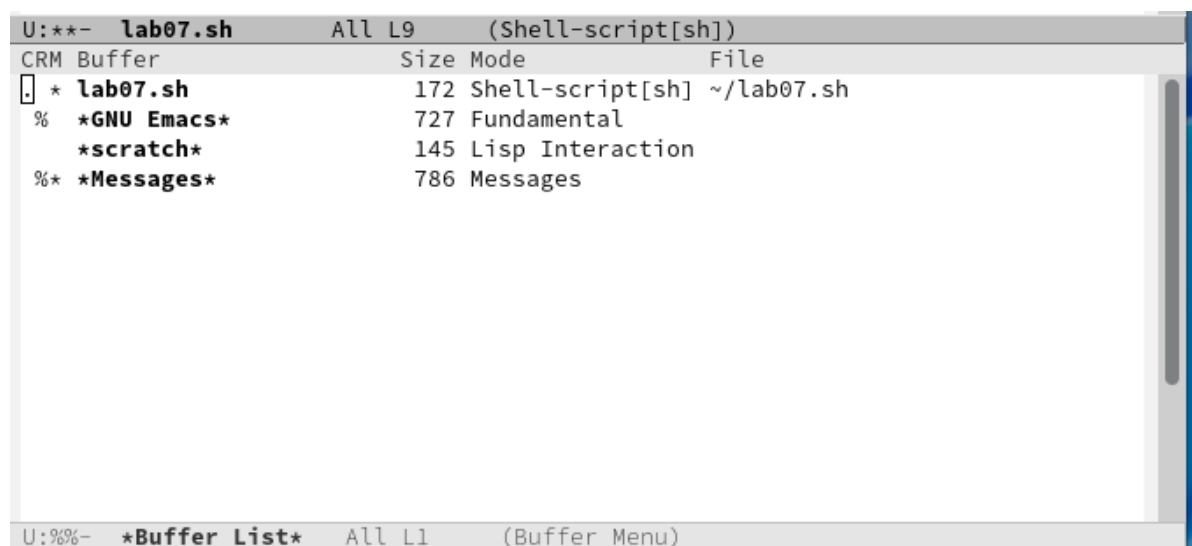


Рис. 0.16: Список активных буферов

7.2. Перемещаюсь во вновь открытое окно со списком открытых буферов

```
#!/bin/bash

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

и переключаюсь на другой буфер:

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

7.3. Закрываю это окно:

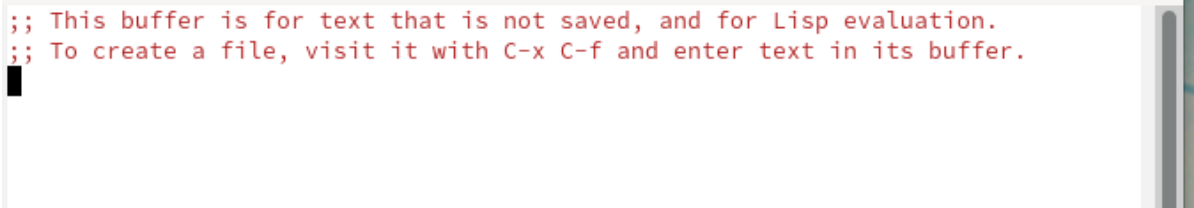
```
#!/bin/bash

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

U:***- lab07.sh All L9 (Shell-script[sh])
Command attempted to use minibuffer while in minibuffer

Рис. 0.17: Закрытие

7.4. Теперь вновь переключаюсь между буферами, но уже без вывода их списка на экран:



```
;; This buffer is for text that is not saved, and for Lisp evaluation.  
;; To create a file, visit it with C-x C-f and enter text in its buffer.
```

Рис. 0.18: Переключение без вывода

8. Управляю окнами:

8.1. Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали,



The image shows a terminal window with a light gray border. The window is split into two vertical panes. Both panes display the same script: a shebang line, a function definition, and two calls to the function. The script is as follows:

```
#!/bin/bash

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

Below the script, the terminal status bar is visible, showing the prompt 'U:***', the filename 'lab07.sh', and the window title 'All L9 (Shell)'. The same status bar is repeated at the bottom of the second pane.

а затем каждое из этих окон на две части по горизонтали:

8.2. В каждом из четырёх созданных окон открываю новый буфер (файл) и ввожу несколько строк текста:



The image displays four terminal windows arranged in a 2x2 grid, each showing the execution of a shell script. The script defines a function named 'hello' that sets a local variable 'HELLO' to 'World' and prints it. The script then calls 'echo \$HELLO', runs the 'hello' function, sets 'HELL=Hello', and calls the 'hello' function again. The terminal windows show the script being executed in a new buffer, with the output 'World' and 'Hello' visible. The status bar of each window indicates the file 'lab07.sh' and the line number (L9 or L15).

```
#!/bin/bash

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

U:*** lab07.sh All L9 (Shell-s

```
#!/bin/bash

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

U:*** lab07.sh All L15 (Shell-s

```
#!/bin/bash

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

U:*** lab07.sh All L9 (Shell-s

```
#!/bin/bash


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

U:*** lab07.sh All L15 (Shell-s

Рис. 0.19: Новый буфер + ввод

9. Режим поиска:

9.1. Переключаюсь в режим поиска и ищу несколько слов, присутствующих в тексте:



```
#!/bin/bash

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

U:*** lab07.sh All L14 (Shell-s U:*** new2 All L3 (Fundame

YESYESYEYES
123
321
NONONONNONNONNOON

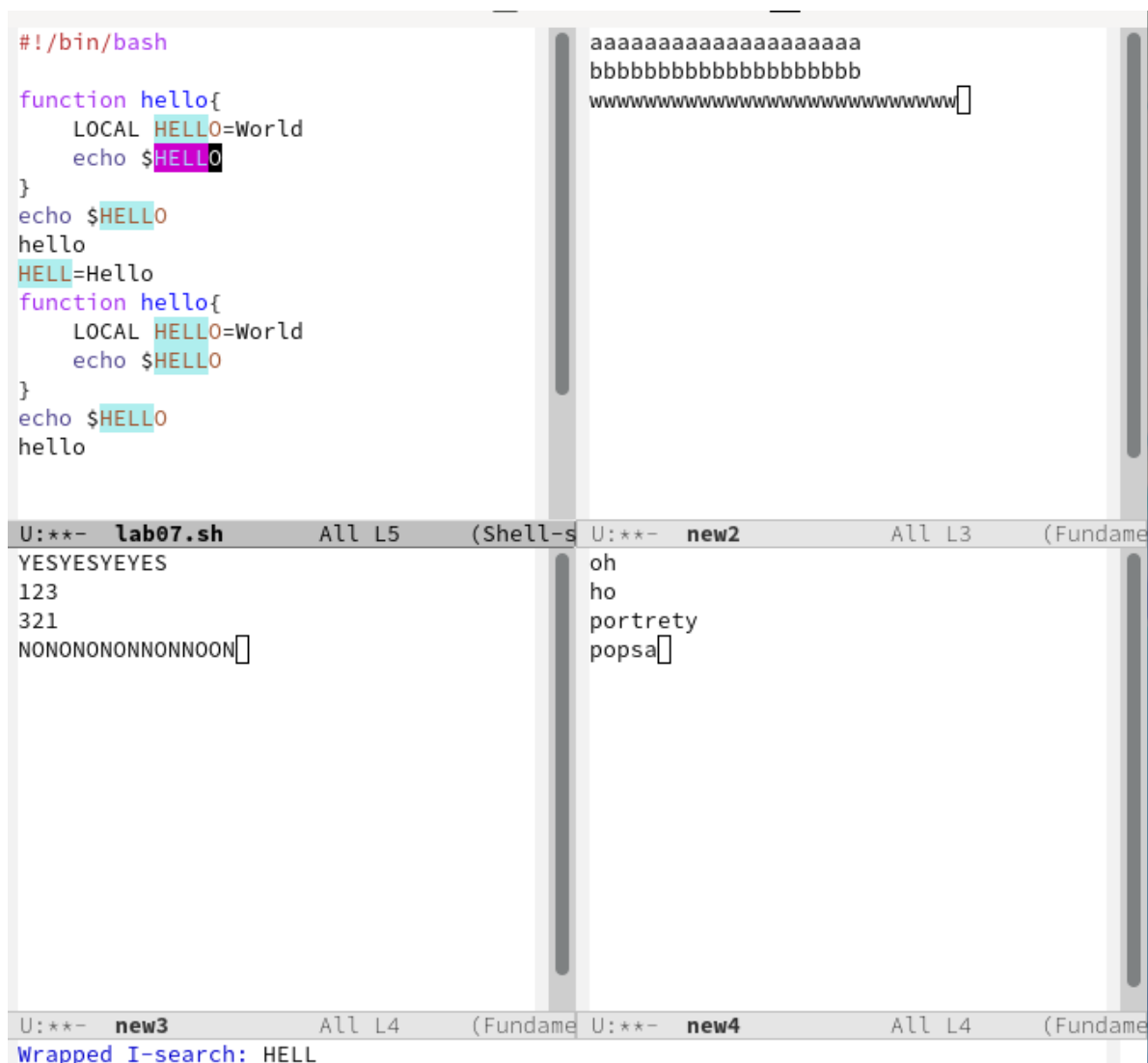
oh
ho
portrety
popsa

U:*** new3 All L4 (Fundame U:*** new4 All L4 (Fundame

I-search: HELL

Рис. 0.20: Режим поиска

9.2. Переключаюсь между результатами поиска:



```
#!/bin/bash

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

U:*** lab07.sh All L5 (Shell-s U:*** new2 All L3 (Fundame

YESYESYYES
123
321
NONONONONNONNOON

oh
ho
portrety
popsa

U:*** new3 All L4 (Fundame U:*** new4 All L4 (Fundame

Wrapped I-search: HELL

Рис. 0.21: Переключение

9.3. Выхожу из режима поиска:



Рис. 0.22: Выход из режима Поиска

9.4. Перехожу в режим поиска и замены, ввожу текст, который следует найти

и заменить, нажимаю Enter , затем ввожу текст для замены. После того как будут подсвечены результаты поиска, нажимаю ! для подтверждения замены:


```
#!/bin/bash

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

U:***- lab07.sh All L14 (Shell-s U:***- new2 All L3 (Fundame
YESYESYEYES
123
321
NONONONONNONNOON

U:***- new3 All L4 (Fundame U:***- new4 All L4 (Fundame
I-search: HELL
```

9.5. Отпробываю другой режим поиска:

```
#!/bin/bash

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

U:*** lab07.sh All L5 (Shell-s U:*** new2 All L3 (Fundame

YESYESYYES
123
321
NONONONONNONNOON

oh
ho
portrety
popsa

U:*** new3 All L4 (Fundame U:*** new4 All L4 (Fundame

Wrapped I-search: HELL

Рис. 0.23: Другой режим поиска

Ответы на вопросы:

1. Кратко охарактеризуйте редактор emacs. Ответ: Emacs представляет собой мощный экраный редактор текста, написанный на языке высокого уровня Lisp.
2. Какие особенности данного редактора могут сделать его сложным для освоения новичком? Ответ: Сложным освоение данной программы для новичка может сделать незнание комбинации клавиш или английского.
3. Своими словами опишите, что такое буфер и окно в терминологии emacs'а
Ответ: Моими словами буфер это динамическая память, а окно- то, что мы видим
4. Можно ли открыть больше 10 буферов в одном окне? Ответ: Можно если нет ограничений на систему.
5. Какие буферы создаются по умолчанию при запуске emacs? Ответ: Буферы, которые открываются по умолчанию: GNU Emacs, scratch, Messages, Quail Completions
6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|? Ответ: Ctrl+c, Shift+ и Ctrl+c Ctrl+
7. Как поделить текущее окно на две части? Ответ: Нажать C-x 3, или C-x 2.
8. В каком файле хранятся настройки редактора emacs? Ответ: Настройки хранятся в файле ~/.emacs.

9. Какую функцию выполняет клавиша Backspace и можно ли её переназначить?

Ответ: Перемещение курсора

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

Ответ: Редактор emacs ,потому что на нем можно работать сразу с несколькими файлами.

Вывод:

Я познакомился с операционной системой Linux. Получил практические навыки работы с редактором Emacs!