

Отчёт по лабораторной работе № 11

Операционные системы

Анастасия Романовна Зинченко

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	28
	Список литературы	29

Список иллюстраций

3.1	emacs	7
3.2	emacs	8
3.3	lab07.sh	8
3.4	Текст	9
3.5	Сохранение	9
3.6	Вырезание	10
3.7	Вставка	10
3.8	Выделение	11
3.9	Копирование и вставка	11
3.10	Вырезание	12
3.11	Отмена	12
3.12	Начало строки	13
3.13	Конец строки	14
3.14	Начало буфера	15
3.15	Конец буфера	16
3.16	Список активных буферов	16
3.17	Переключение	17
3.18	Закрытие	18
3.19	Переключение	18
3.20	Деление	19
3.21	Новый буфер	20
3.22	Режим поиска	21
3.23	Переключение	22
3.24	Выход	23
3.25	Замена	24
3.26	Режим поиска	25

Список таблиц

1 Цель работы

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

2 Задание

1. Ознакомиться с теоретическим материалом.
2. Ознакомиться с редактором емас.
3. Выполнить упражнения.
4. Ответить на контрольные вопросы.

3 Выполнение лабораторной работы

Скачала emacs. (рис. 3.1).

```
[arzinchenko@arzinchenko ~]$ sudo dnf install emacs
Copr repo for gitflow owned by elegos 2.3 kB/s | 1.5 kB 00:00
Copr repo for browserpass owned by maxi 2.2 kB/s | 1.5 kB 00:00
Copr repo for iosevka owned by peterwu 6.9 kB/s | 1.5 kB 00:00
Copr repo for iosevka owned by peterwu 97 kB/s | 53 kB 00:00
```

Рис. 3.1: emacs

Открыла emacs. (рис. 3.2).

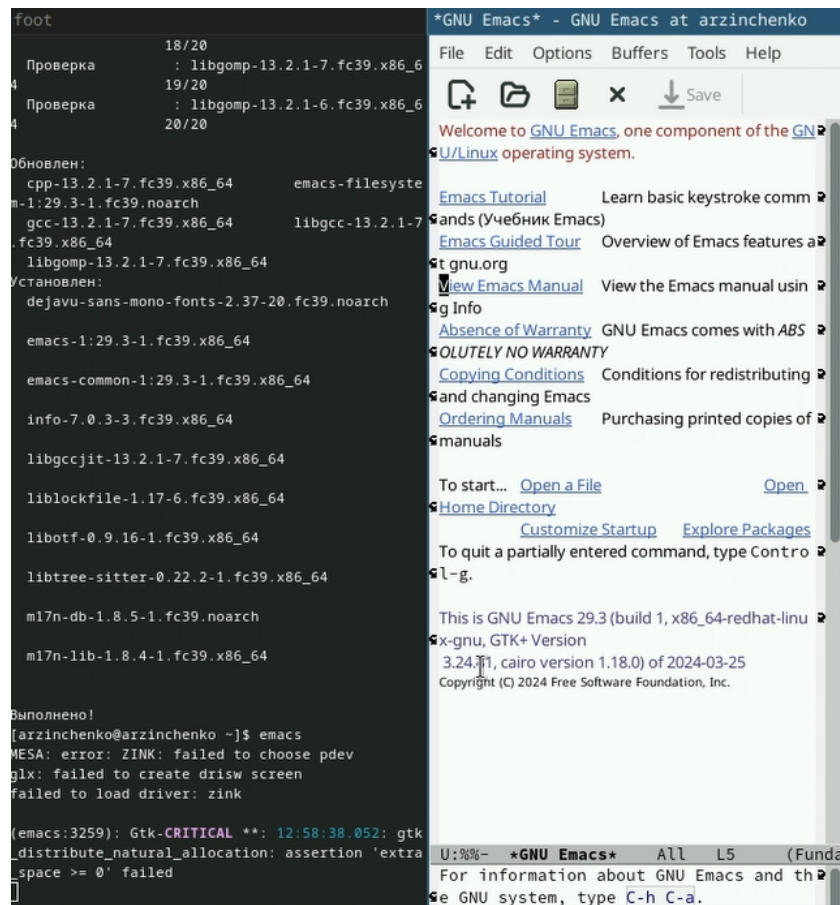


Рис. 3.2: emacs

Создала файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f. (рис. 3.2).



Рис. 3.3: lab07.sh

Набрала текст (рис. 3.4).


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

Рис. 3.4: Текст

Сохранила файл с помощью комбинации Ctrl-x Ctrl-s. (рис. 3.5).

```
Wrote /home/arzinchenko/lab07.sh
```

Рис. 3.5: Сохранение

Вырезала одной командой целую строку (C-k). (рис. 3.6).

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




Рис. 3.6: Вырезание

Вставила эту строку в конец файла (С-у). (рис. 3.7).

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




Рис. 3.7: Вставка

Выделила область текста (C-space). (рис. 3.8).

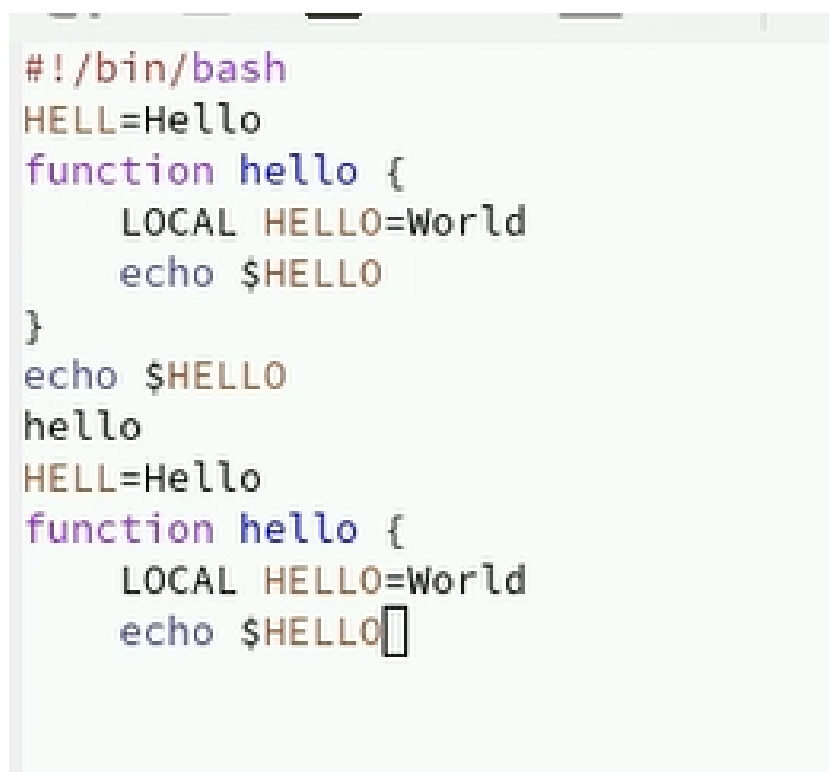


```
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

The screenshot shows a terminal window with a light green background. A shell script is displayed. The lines 'function hello {' and 'LOCAL HELLO=World' are highlighted in yellow. The line 'echo \$HELLO' is highlighted in orange. The rest of the script is in its original color.

Рис. 3.8: Выделение

Скопировала область в буфер обмена (M-w) и вставила область в конец файла. (рис. 3.9).



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

The screenshot shows the same terminal window as in Figure 3.8, but now with an additional block of code appended at the end of the file. The new block consists of 'HELL=Hello' and a new 'function hello {' block with 'LOCAL HELLO=World' and 'echo \$HELLO'. The cursor is at the end of the last line.

Рис. 3.9: Копирование и вставка

Вновь выделила эту область и на этот раз вырезала её (C-w). (рис. 3.10).

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

Рис. 3.10: Вырезание

Отменила последнее действие (C-/). (рис. 3.11).

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

Рис. 3.11: Отмена

Переместила курсор в начало строки (C-a). (рис. 3.12).



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

Рис. 3.12: Начало строки

Переместила курсор в конец строки (C-e). (рис. 3.13).

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

Рис. 3.13: Конец строки

Переместила курсор в начало буфера (M-<). (рис. 3.14).

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

Рис. 3.14: Начало буфера

Переместила курсор в конец буфера (M->). (рис. 3.15).

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

Рис. 3.15: Конец буфера

Вывела список активных буферов на экран (C-x C-b). (рис. 3.16).

-:*** lab07.sh All L12 (Shell-script[bash])				
CRM	Buffer	Size	Mode	File
	* lab07.sh	163	Shell-script[ba...	~/lab07.sh
%	*GNU Emacs*	870	Fundamental	
	scratch	145	Lisp Interaction	
%%	*Messages*	1767	Messages	
%%	*Async-native-compile...	165	Fundamental	
U:%%- *Buffer List* All L1 (Buffer Menu)				
C-x C-b				

Рис. 3.16: Список активных буферов

Переместилась во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер. (рис. 3.17).

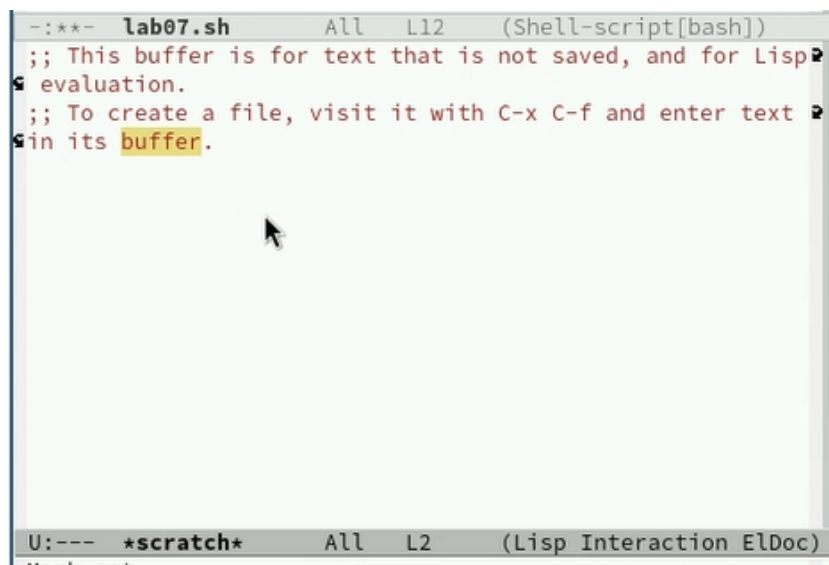


Рис. 3.17: Переключение

Закрала это окно (C-x 0). (рис. 3.18).

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



Рис. 3.18: Заккрытие

Теперь вновь переключилась между буферами, но уже без вывода их списка на экран (C-x b). (рис. 3.19).

```
Switch to buffer (default *scratch*):
```

Рис. 3.19: Переключение

Поделила фрейм на 4 части: разделила фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2). (рис. 3.20).



Рис. 3.20: Деление

В каждом из четырёх созданных окон открыла новый буфер (файл) и ввела несколько строк текста. (рис. 3.21).

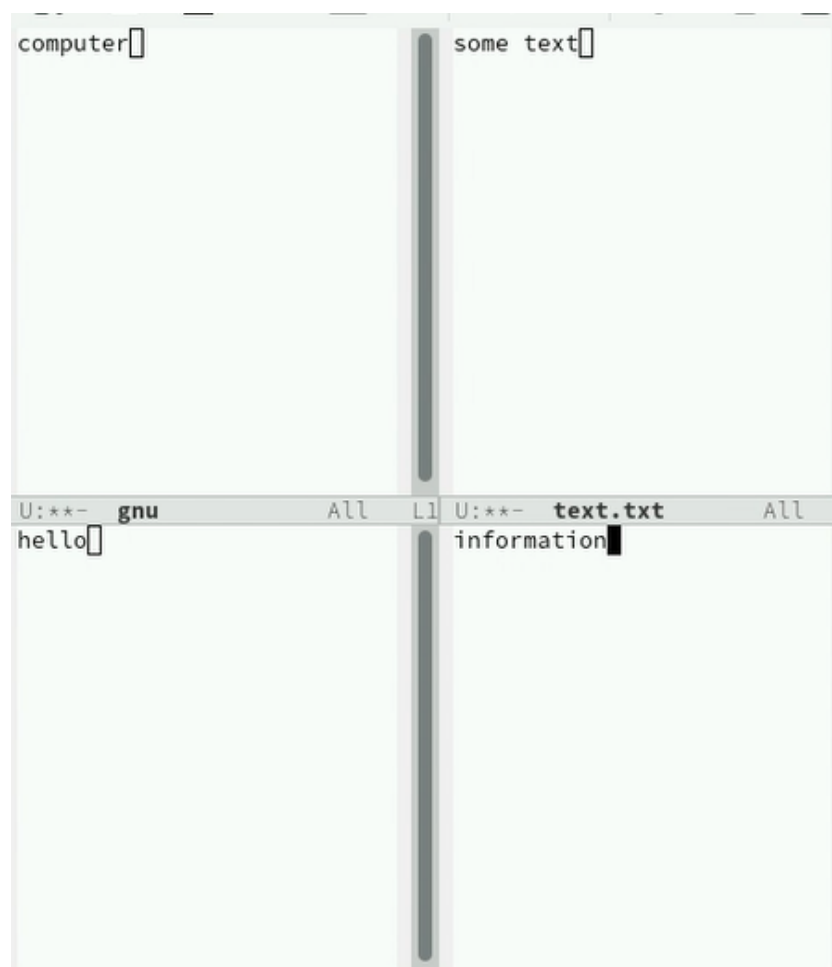


Рис. 3.21: Новый буфер

Переключилась в режим поиска (C-s) и нашла несколько слов, присутствующих в тексте. (рис. 3.22).

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

~:*** lab07.sh All L12 (Shell)

I-search: hello

Рис. 3.22: Режим поиска

Переключилась между результатами поиска, нажимая C-s. (рис. 3.23).

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

Рис. 3.23: Переключение

Вышла из режима поиска, нажав C-g. (рис. 3.24).

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

Рис. 3.24: Выход

Перешла в режим поиска и замены (M-%), ввела текст, который следует найти и заменить, нажала Enter, затем ввела текст для замены. (рис. 3.25).

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

Рис. 3.25: Замена

Испробовала другой режим поиска, нажав M-s o. (рис. 3.26).



Рис. 3.26: Режим поиска

Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.

Emacs — один из наиболее мощных и широко распространённых редакторов, используемых в мире UNIX. Написан на языке высокого уровня Lisp.

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

Большое разнообразие сложных комбинаций клавиш, которые необходимы для редактирования файла и в принципе для работа с Emacs.

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

Буфер - это объект в виде текста. Окно - это прямоугольная область, в которой отображен буфер.

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

Да, можно.

5. Какие буферы создаются по умолчанию при запуске emacs?

Emacs использует буферы с именами, начинающимися с пробела, для внутренних целей. Отчасти он обращается с буферами с такими именами особым образом — например, по умолчанию в них не записывается информация для отмены изменений.

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

Ctrl + c, а потом | и Ctrl + c Ctrl + |

7. Как поделить текущее окно на две части?

С помощью команды Ctrl + x 3 (по вертикали) и Ctrl + x 2 (по горизонтали)

8. В каком файле хранятся настройки редактора emacs?

Настройки emacs хранятся в файле .emacs, который хранится в домашней директории пользователя. Кроме этого файла есть ещё папка .emacs.

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

Выполняет функцию стереть, думаю можно переназначить.

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

Для меня удобнее был редактор Emacs, так как у него есть командная оболочка. А vi открывается в терминале, и выглядит своеобразно.

4 Выводы

Я познакомилась с операционной системой Linux. Получила практические навыки работы с редактором Emacs.

Список литературы