

Отчёт по лабораторной работе №9

Операционные системы

Сячинова Ксения Ивановна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на контрольные вопросы.	19
4	Выводы	21

Список иллюстраций

2.1	Запуск Emacs	6
2.2	Создание файла	6
2.3	Текст	7
2.4	Вырезаем строку	7
2.5	Вставляем строку в конец файла	8
2.6	Выделение текста	8
2.7	Вставление текста	9
2.8	Вырезание текств	10
2.9	Отмена действия	11
2.10	Перемещение курсора в начало строки	11
2.11	Перемещение курсора в конец строки	12
2.12	Перемещение курсора в начало буфера	12
2.13	Перемещение курсора в конец буфера	13
2.14	Список активных буферов	13
2.15	Переключение на другой буфер	14
2.16	Окно	14
2.17	Переключение между буферами	15
2.18	Разделение окна на 4 части	15
2.19	Новые файлы	16
2.20	Поиск слов	16
2.21	Переключение между результатами	17
2.22	Замена слова	17
2.23	Итог	17

Список таблиц

1 Цель работы

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

2 Выполнение лабораторной работы

1. Для работы в “Emacs” необходимо установить его. Открываем его с помощью команды “emacs &”.(рис. 2.1)

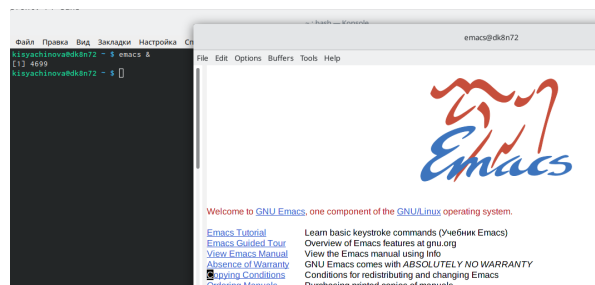


Рис. 2.1: Запуск Emacs

2. Создаём файл lab07.sh с помощью комбинаций “ctrl-x”, “ctrl-f”.(рис. 2.2)

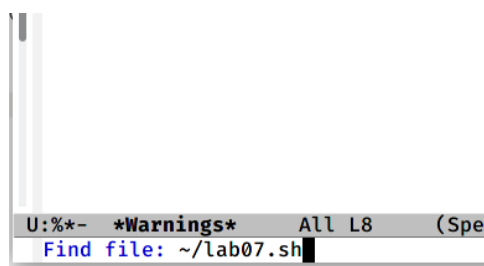


Рис. 2.2: Создание файла

3. Напишем необходим текст. (рис. 2.3)

A screenshot of the GNU Emacs editor window. The title bar shows "Ordering Manuals" and "Emacs". The window title is "U:%%- *GNU Emacs* Top". The editor contains a shell script with the following text:

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

The cursor is at the end of the "hello" line.

Рис. 2.3: Текст

4. Сохраняем файл с помощью комбинаций клавиш "ctrl-x", "ctrl-s"
5. Далее сделаем с файлом определённые действия
 - 5.1. Вырезать одной командой целую строку (C-k).(рис. 2.4)

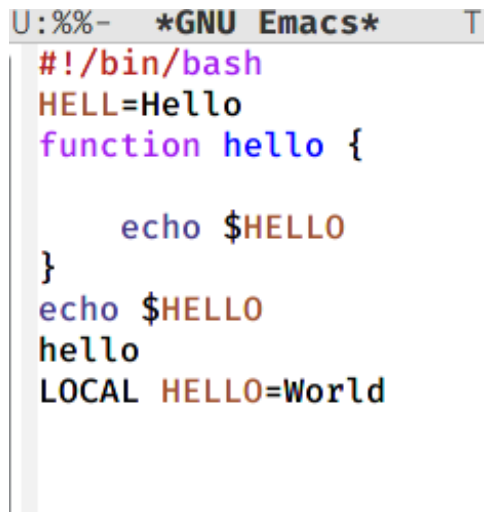
A screenshot of the GNU Emacs editor window, similar to Figure 2.3. The title bar shows "Ordering Manuals" and "Emacs". The window title is "U:%%- *GNU Emacs*". The editor contains the same shell script as in Figure 2.3:

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

The cursor is at the end of the "hello" line.

Рис. 2.4: Вырезаем строку

- 5.2. Вставить эту строку в конец файла (C-y).(рис. 2.5)




The screenshot shows the GNU Emacs editor window with a file named "T". The script content is as follows:

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

A vertical cursor is positioned at the end of the file, after the last line.

Рис. 2.5: Вставляем строку в конец файла

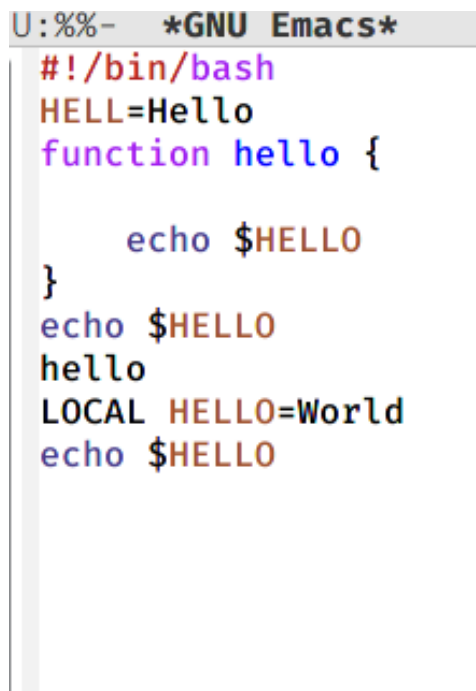
- 5.3. Выделить область текста (C-space).(рис. 2.6)



The screenshot shows the same GNU Emacs editor window. The line "echo \$HELLO" within the function definition is now highlighted with a light blue background, indicating it has been selected.

Рис. 2.6: Выделение текста

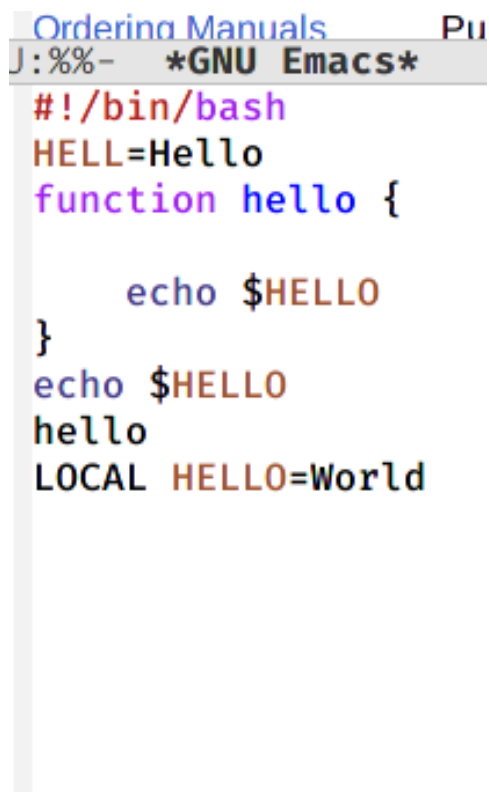
- 5.4. Скопировать область в буфер обмена (M-w).
- 5.5. Вставить область в конец файла(рис. 2.7)

A screenshot of the GNU Emacs editor interface. The title bar at the top reads "U:%%- *GNU Emacs*". The main editing area shows a Bash script being pasted into a buffer. The text is color-coded: "#!/bin/bash" is in red, "HELL=Hello" is in orange, "function hello {" is in blue, "echo \$HELLO" is in orange, "}" is in blue, "echo \$HELLO" is in orange, "hello" is in blue, "LOCAL HELLO=World" is in orange, and "echo \$HELLO" is in orange. A vertical line on the left indicates the current cursor position.

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

Рис. 2.7: Вставление текста

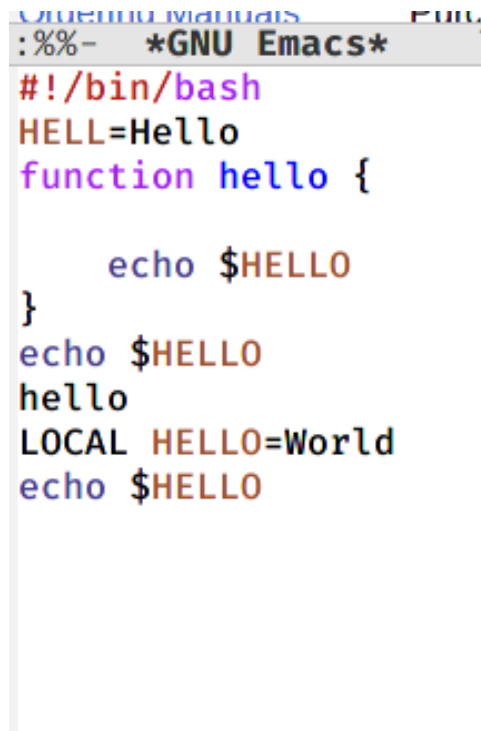
- 5.6. Вновь выделить эту область и на этотраз вырезать её (C-w).(рис. 2.8)

A screenshot of a GNU Emacs editor window. The title bar at the top shows "Ordering Manuals" and "Pu". The Emacs status bar at the top displays "J:%%- *GNU Emacs*". The main editing area contains a shell script with the following text:

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

Рис. 2.8: Вырезание текств

- 5.7. Отмените последнее действие (C-/).(рис. 2.9)



```

:%%- *GNU Emacs*
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
echo $HELLO

```

Рис. 2.9: Отмена действия

6. Действия с курсором.

- 6.1. Перемещаем курсор в начало строки (C-a), (рис. 2.10)



```

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

```

Рис. 2.10: Перемещение курсора в начало строки

- 6.2. Перемещаем курсор в конец строки (C-e), (рис. 2.11)

```

    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
echo $HELLO

```

Рис. 2.11: Перемещение курсора в конец строки

- 6.3. Перемещаем курсор в начало буфера (M-<), (рис. 2.12)

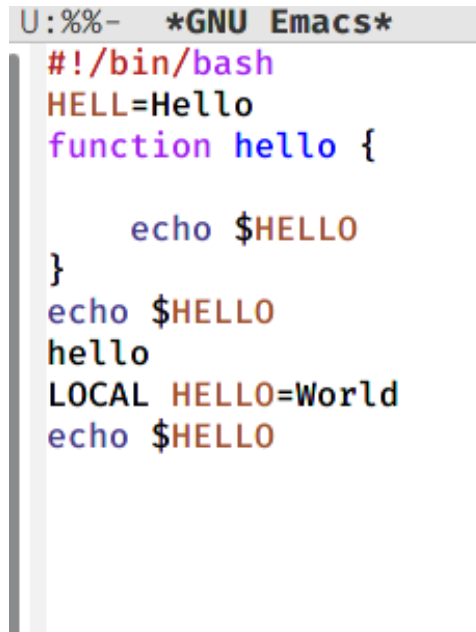
```

Ordering Manuals      Purch
U:%%- *GNU Emacs* T
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
echo $HELLO

```

Рис. 2.12: Перемещение курсора в начало буфера

- 6.4. Перемещаем курсор в конец буфера (M->), (рис. 2.13)

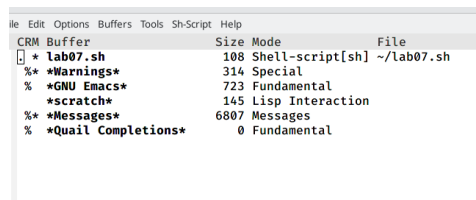
A screenshot of the GNU Emacs editor window. The title bar reads "U:%%- *GNU Emacs*". The buffer contains a shell script with the following content:

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

Рис. 2.13: Перемещение курсора в конец буфера

7. Управление буферами.

- 7.1. Выведем список активных буферов на экран (C-x C-b).(рис. 2.14)

A screenshot of the GNU Emacs editor window showing the list of active buffers. The title bar reads "U:%%- *GNU Emacs*". The buffer list is displayed as follows:

CRM	Buffer	Size	Mode	File
	* lab07.sh	108	Shell-script[sh]	~/lab07.sh
%	*Warnings*	314	Special	
%	*GNU Emacs*	723	Fundamental	
%	*scratch*	145	Lisp Interaction	
%	*Messages*	6807	Messages	
%	*Quail Completions*	0	Fundamental	

Рис. 2.14: Список активных буферов

- 7.2. Переместим во вновь открытое окно (C-x o) со списком открытых буферов и переключимся на другой буфер (для этого нажмём “enter”).(рис. 2.15)

```

:***- lab07.sh All L13 (Shell-script[bash]) Пр ма
CRM Buffer Size Mode File
lab07.sh 110 Shell-script[... ~/lab07.s
% *GNU Emacs* 723 Fundamental
% *scratch* 145 Lisp Interaction
% *Messages* 4045 Messages
% *Warnings* 314 Special
% *Quail Completions* 0 Fundamental

```

Рис. 2.15: Переключение на другой буфер

- 7.3. Закроем это окно (C-x 0), (рис. 2.16)

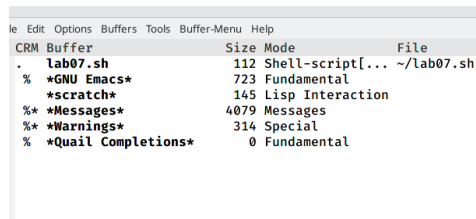
```

file Edit Options Buffers Tools Sh-Scr
#!/bin/bash
HELL=Hello
function hello {
    echo $HELLO
}
echo $HELLO
hello
LOCAL HELLO=World
echo $HELLO

```

Рис. 2.16: Окно

- 7.4. Теперь вновь переключимся между буферами, но уже без вывода их списка на экран (C-x b), (рис. 2.17)



CRM	Buffer	Size	Mode	File
.	lab07.sh	112	Shell-script[...]	~/lab07.sh
%	*GNU Emacs*	723	Fundamental	
%	*scratch*	145	Lisp Interaction	
%*	*Messages*	4079	Messages	
%*	*Warnings*	314	Special	
%	*Quail Completions*	0	Fundamental	

Рис. 2.17: Переключение между буферами

8. Управление окнами

- 8.1. Поделим фрейм на 4 части: разделим фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части по горизонтали (C-x 2). (рис. 2.18)



Рис. 2.18: Разделение окна на 4 части

- 8.2. В каждом из четырёх созданных окон откроем новый буфер (файл) и введём несколько строк текста. Для этого я заранее создала 4 файла с разным текстом. (рис. 2.19)

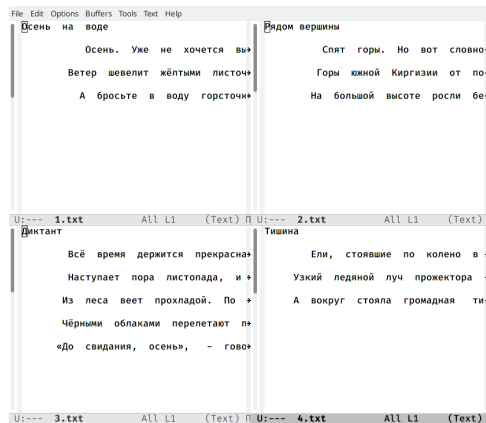


Рис. 2.19: Новые файлы

9. Режим поиска

- 9.1. Перключимся в режим поиска (C-s) и найдём несколько слов, присутствующих в тексте.(рис. 2.20)

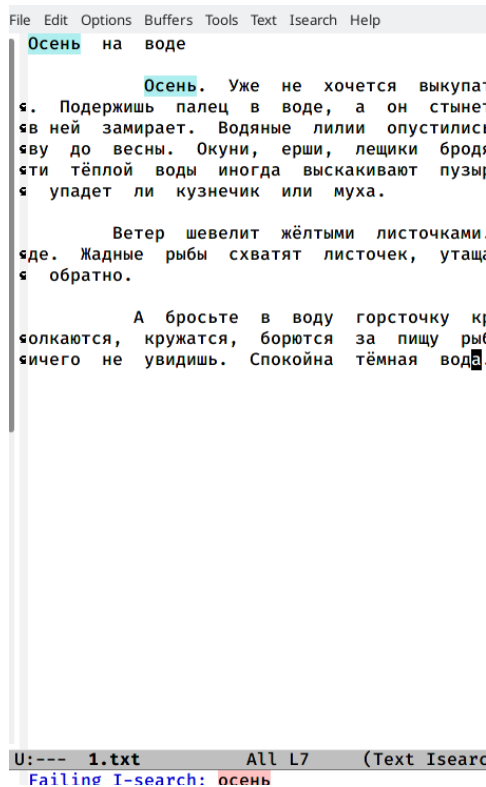


Рис. 2.20: Поиск слов

- 9.2. Переключимся между результатами поиска, нажимая C-s. (рис. 2.21)

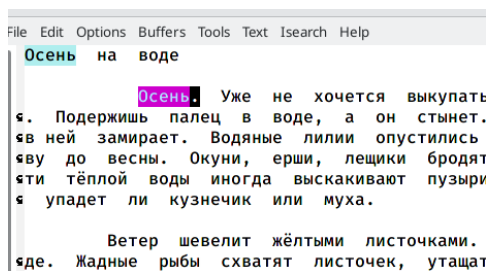


Рис. 2.21: Переключение между результатами

- 9.3. Выходим из режима поиска, нажав C-g.
- 9.4. Переходим в режим поиска и замены (M-%), вводим текст, который следует найти и заменить, нажимаем Enter, затем вводим текст для замены. После того как будут подсвечены результаты поиска, нажимаем “!” для подтверждения замены. (рис. 2.22), (рис. 2.23)

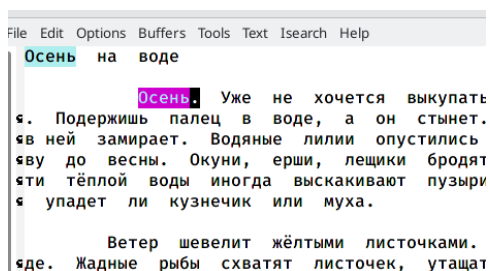


Рис. 2.22: Замена слова

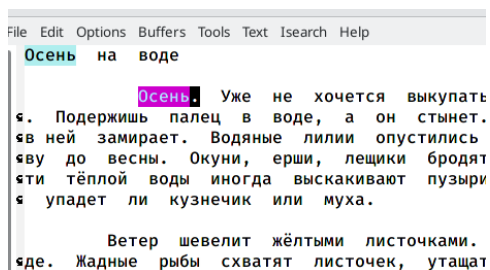



Рис. 2.23: Итог

- 9.5. Попробуем режим поиска (M-s o). Данный поиск отличается тем, что тут считывается строка поиска, которая трактуется как регулярное выражение, и не осуществляем поиск точно совпадения в тексте буфера. Регулярное выражение - это образец, который обозначает набор строк, возможно, и неограниченный набор.

3 Ответы на контрольные вопросы.

1. Emacs – один из наиболее мощных и широко распространённых редакторов, используемых в мире Unix. По популярности он соперничает с редактором vi и его клонами. В зависимости от ситуации, Emacs может быть:
 - текстовым редактором;
 - программой для чтения почты и новостей Usenet;
 - интегрированной средой разработки (IDE);
 - операционной системой и т.д. Всё это разнообразие достигается благодаря архитектуре Emacs, которая позволяет расширять возможности редактора при помощи языка Emacs Lisp. На языке C написаны лишь самые базовые и низкоуровневые части Emacs, включая полнофункциональный интерпретатор языка Lisp. Таким образом, Emacs имеет встроенный язык программирования, который может использоваться для настройки, расширения и изменения поведения редактора. В действительности, большая часть того редактора, с которым пользователи Emacs работают в наши дни, написана на языке Lisp.
2. Основную трудность для новичков при освоении данного редактора могут составлять большое количество команд, комбинаций клавиш, которые не получится все запомнить с первого раза и поэтому придется часто обращаться к справочным материалам.
3. Буфер – это объект, представляющий собой текст. Если имеется несколько буферов, то редактировать можно только один. Обычно буфер считывает

данные из файла или записывает в файл данные из буфера. Окно – это область экрана, отображающая буфер. При запуске редактора отображается одно окно, но при обращении к некоторым функциям могут открыться дополнительные окна. Окна Emacs и окна графической среды X Window – разные вещи. Одно окно X Window может быть разбито на несколько окон в смысле Emacs, в каждом из которых отображается отдельный буфер.

4. Да, можно.
5. При запуске Emacs по умолчанию создаются следующие буферы:
 - «scratch» (буфер для несохраненного текста)
 - «Messages» (журнал ошибок, включающий также информацию, которая появляется в области EchoArea)
 - «GNU Emacs» (справочный буфер о редакторе)
6. C-c | сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и нажимаю «|» C-c C-| сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и, удерживая «ctrl», нажимаю «|»
7. Чтобы разделить окно на две части необходимо воспользоваться комбинацией «Ctrl-x 3» (по вертикали) или «Ctrl-x 2» (по горизонтали).
8. Настройки Emacs хранятся в файле .emacs.
9. По умолчанию клавиша «» удаляет символ перед курсором, но в редакторе её можно переназначить. Для этого необходимо изменить конфигурацию файла .emacs.
10. Более удобным я считаю редактор emacs, потому что в нем проще открывать другие файлы, можно использовать сразу несколько окон, нет «Командного режима», «Режима ввода», «Режима командной строки», которые являются немного непривычными и в какой-то степени неудобными.

4 Выводы

В ходе выполнения данной лабораторной работы я познакомилась с операционной системой Linux и получила практические навыки по работе с редактором Emacs.