

Отчет по лабораторной работе №9

дисциплина: Операционные системы

Старков Никита Алексеевич

Содержание

1	Цель работы	4
1.1	Основные команды Emacs	4
1.2	Стандартные процедуры	5
2	Контрольные вопросы	15
3	Выводы	17

Список иллюстраций

1.1	Emacs.	4
1.2	Создание файла lab07.sh.	4
1.3	Ввод текста.	4
1.4	Сохранение.	5
1.5	Вырезание строки.	5
1.6	Вставка в конец файла.	6
1.7	Выделение.	6
1.8	Вставка.	6
1.9	Вырезание.	7
1.10	Отмена последнего действия.	7
1.11	Перемещение курсора в начало строки.	7
1.12	Перемещение курсора в конец строки.	8
1.13	Перемещение курсора в начало буфера обмена.	8
1.14	Перемещение курсора в конец буфера обмена.	9
1.15	Выведение списка.	9
1.16	Переключение на другой буфер.	10
1.17	Переключение без вывода списка на экран.	10
1.18	Деление на 4 части.	11
1.19	Создание файлов.	11
1.20	Вставка нескольких строк текста.	12
1.21	Поиск слов.	13
1.22	Другой режим поиска.	14

1 Цель работы

Цель работы: познакомиться с операционной системой Linux. Получить практические навыки работы с редактором Emacs.

1.1 Основные команды Emacs

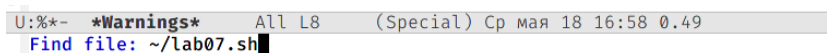
1)Открыл emacs:



```
nastarkov@dk6n54 ~ $ emacs
```

Рис. 1.1: Emacs.

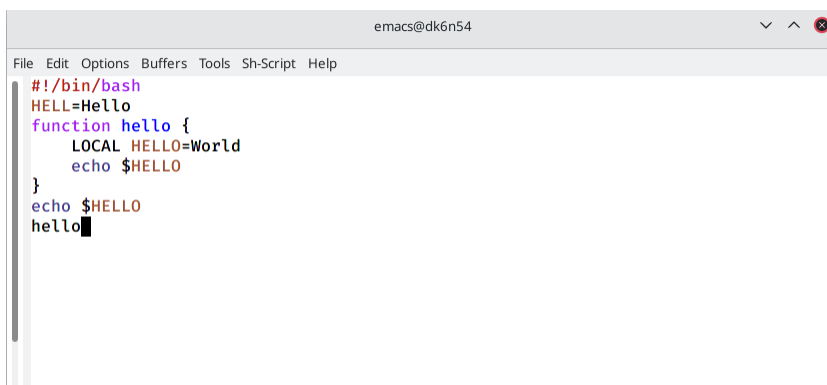
2)Создал файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f):



```
U:%*- *Warnings* All L8 (Special) Ср мая 18 16:58 0.49
Find file: ~/lab07.sh
```

Рис. 1.2: Создание файла lab07.sh.

3)Набрал текст:



```
emacs@dk6n54
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Рис. 1.3: Ввод текста.

4) Сохранил файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s):



```
emacsd@dk6n54
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
  LOCAL HELLO=World
  echo $HELLO
}
echo $HELLO
hello
```

U:--- lab07.sh All L8 (Shell-script[sh]) Ср мая 18 17:02 0.62

Рис. 1.4: Сохранение.

1.2 Стандартные процедуры

5) Проделал с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш:

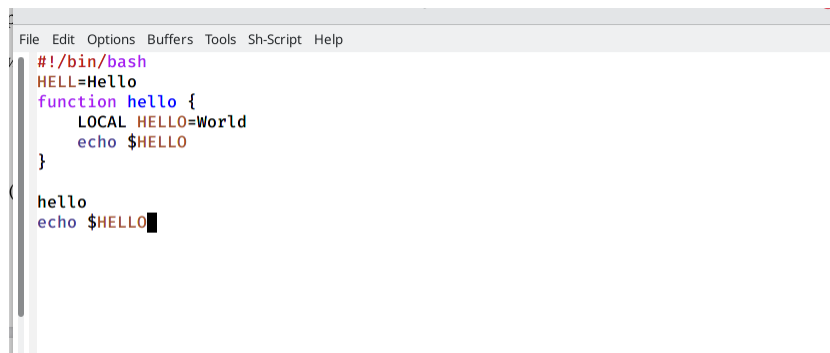
5.1. Вырезать одной командой целую строку (C-k):



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
  LOCAL HELLO=World
  echo $HELLO
}
hello
```

Рис. 1.5: Вырезание строки.

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

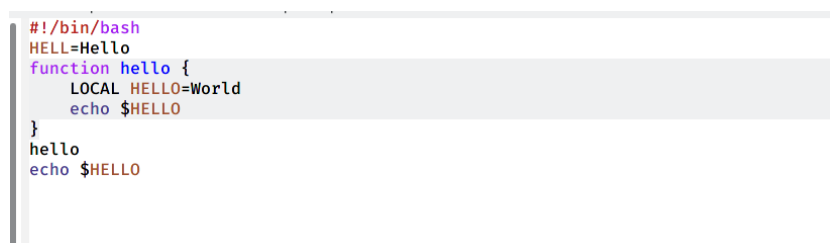


```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}

hello
echo $HELLO
```

Рис. 1.6: Вставка в конец файла.

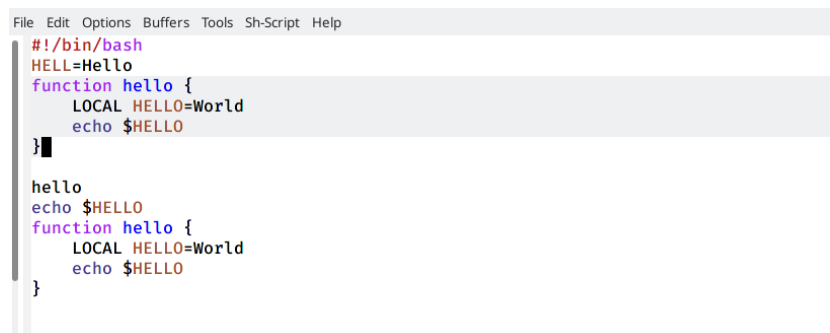
5.3 Выделить область текста (C-space):



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

Рис. 1.7: Выделение.

5.4 Скопировать область в буфер обмена (M-w). А затем вставить область в конец файла.

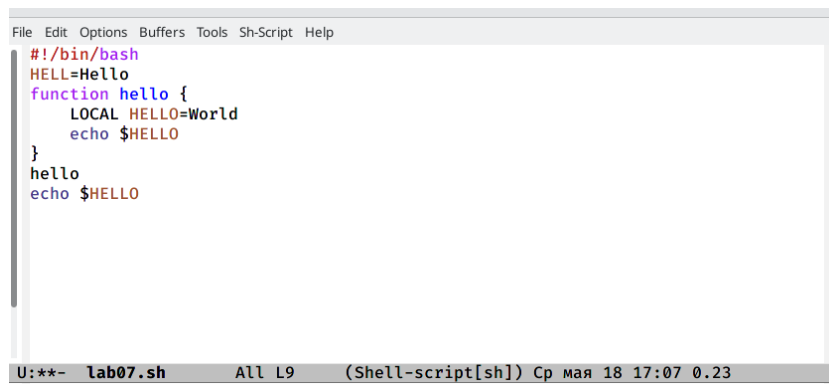


```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}

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

Рис. 1.8: Вставка.

5.5 Вновь выделить эту область и на этот раз вырезать её (C-w):

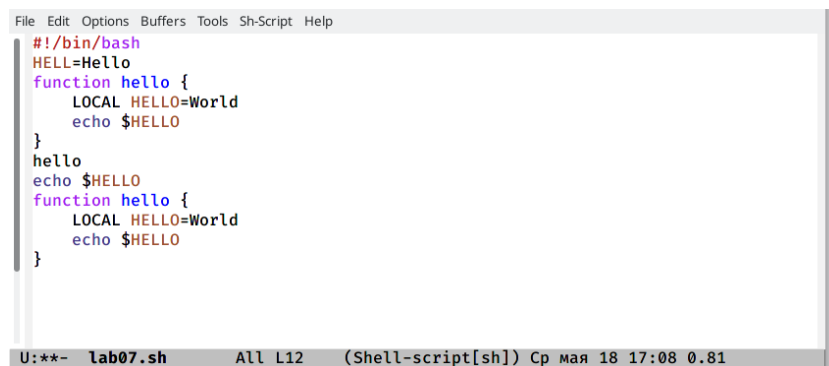


```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
echo $HELLO

U:*** lab07.sh All L9 (Shell-script[sh]) Ср мая 18 17:07 0.23
```

Рис. 1.9: Вырезание.

5.6 Отмена последнего действия (C-/):



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
echo $HELLO
function hello {
    LOCAL HELLO=World
    echo $HELLO
}

U:*** lab07.sh All L12 (Shell-script[sh]) Ср мая 18 17:08 0.81
```

Рис. 1.10: Отмена последнего действия.

6) Научился использовать команды по перемещению курсора:

6.1 Перемещение курсора в начало строки (C-a):



```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
echo $HELLO
function hello {
    LOCAL HELLO=World
    echo $HELLO
}

U:*** lab07.sh All L12 (Shell-script[sh]) Ср мая 18 17:08 0.81
```

Рис. 1.11: Перемещение курсора в начало строки.

6.2 Перемещение курсора в конец строки (C-e):

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
echo $HELLO
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
```

Рис. 1.12: Перемещение курсора в конец строки.

6.3 Перемещение курсора в начало буфера обмена (M-<):

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
hello
echo $HELLO
function hello {
    LOCAL HELLO=World
    echo $HELLO
}
```

RUU:**~ lab07.sh All L7 (Shell-script[sh]) Cp мая 18 17:12 0.45

CRM	Buffer	Size	Mode	File
	* lab07.sh	157	Shell-script[sh]	~/lab07.sh
%	*GNU Emacs*	723	Fundamental	
%	*scratch*	145	Lisp Interaction	
%*	*Messages*	5055	Messages	
%*	*Warnings*	314	Special	
%	*Quail Completions*	0	Fundamental	

Рис. 1.13: Перемещение курсора в начало буфера обмена.

6.4 Перемещение курсора в конец буфера обмена (M->):

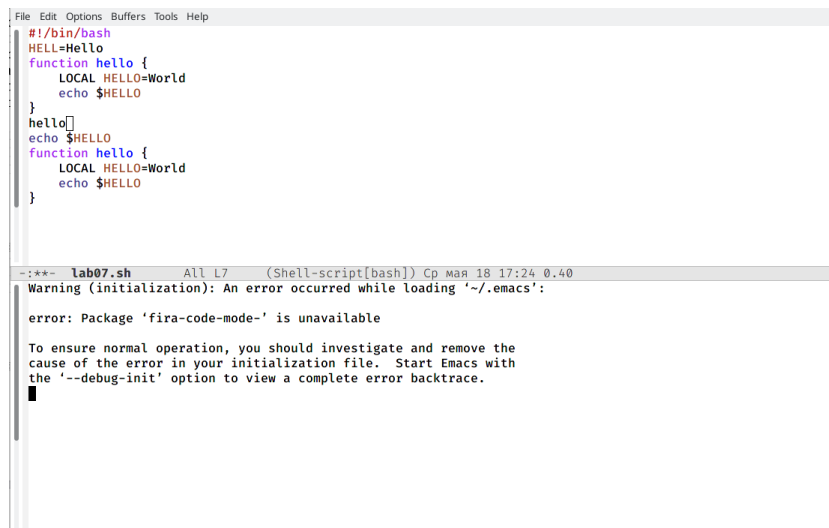


Рис. 1.14: Перемещение курсора в конец буфера обмена.

7) Управлять буферами:

7.1 Вывести список активных буферов на экран (C-x C-b):

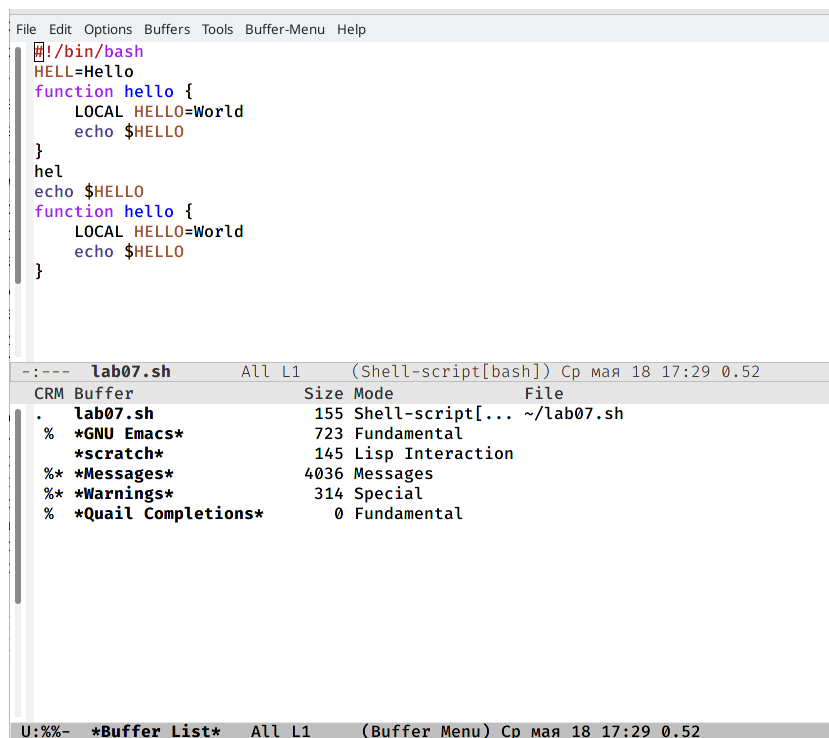


Рис. 1.15: Выведение списка.

7.2 Переместиться во вновь открытое окно (C-x) о со списком открытых буферов

и переключиться на другой буфер:



Рис. 1.16: Переключение на другой буфер.

7.3 Закрывать окно (C-x 0) и вновь переключиться между буферами, но уже без вывода их списка на экран (C-x b):

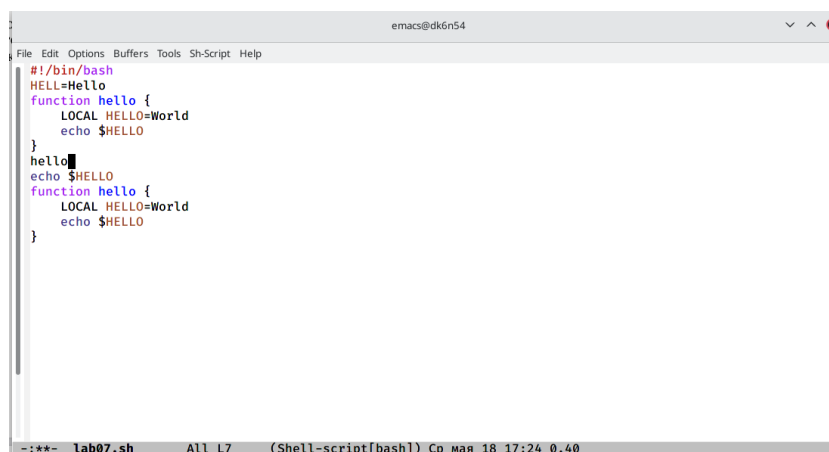


Рис. 1.17: Переключение без вывода списка на экран.

8) Освоил управление окнами:

8.1. Поделить фрейм на 4 части: разделить фрейм на два окна по вертикали (С-х 3), а затем каждое из этих окон на две части по горизонтали (С-х 2).

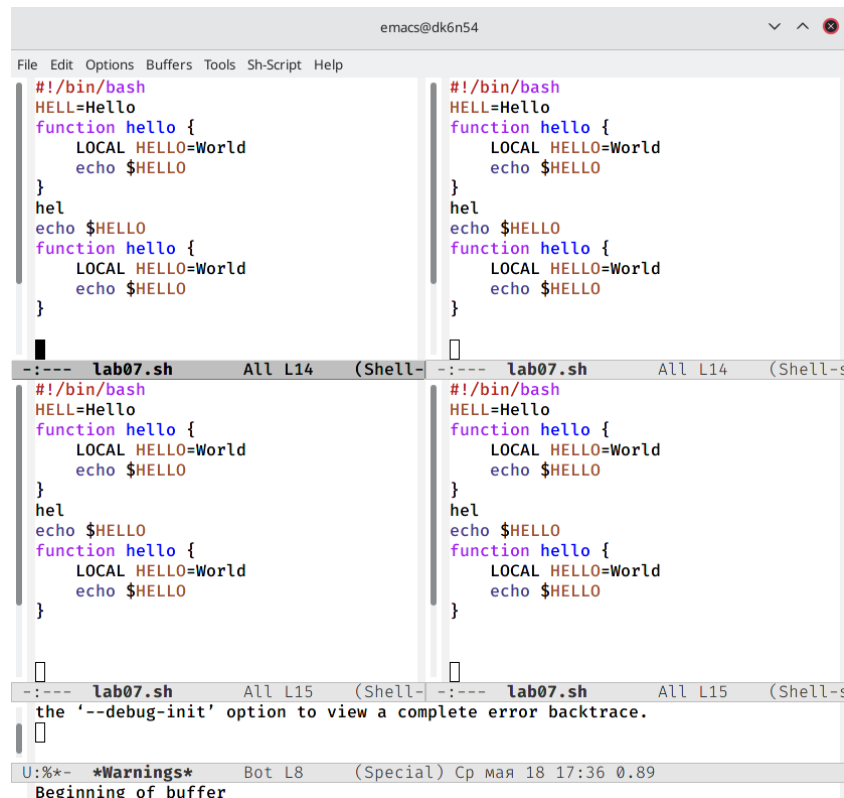


Рис. 1.18: Деление на 4 части.

8.2 В каждом из четырёх созданных окон открыть новый буфер (файл) и ввести несколько строк текста:

```
nastarkov@dk6n54 ~ $ touch f1.txt f2.txt f3.txt f4.txt
```

Рис. 1.19: Создание файлов.



Рис. 1.20: Вставка нескольких строк текста.

9)Режим поиска:

9.1. Переключил в режим поиска (C-s) и нашел несколько слов, присутствующих в тексте:

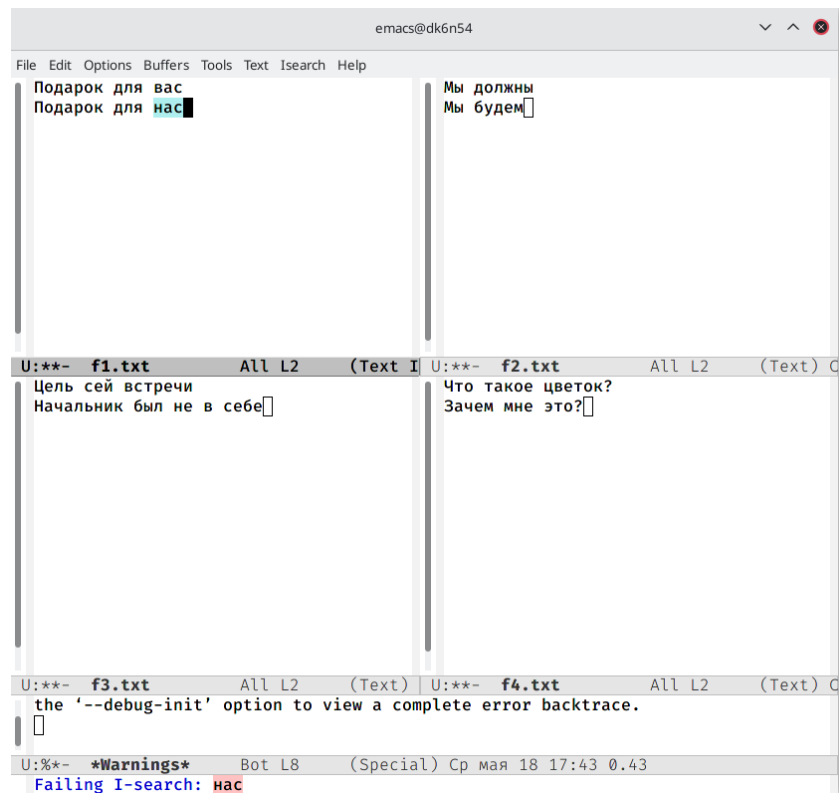


Рис. 1.21: Поиск слов.

9.2 Попробовал использовать другой режим поиска, нажав M-s о:

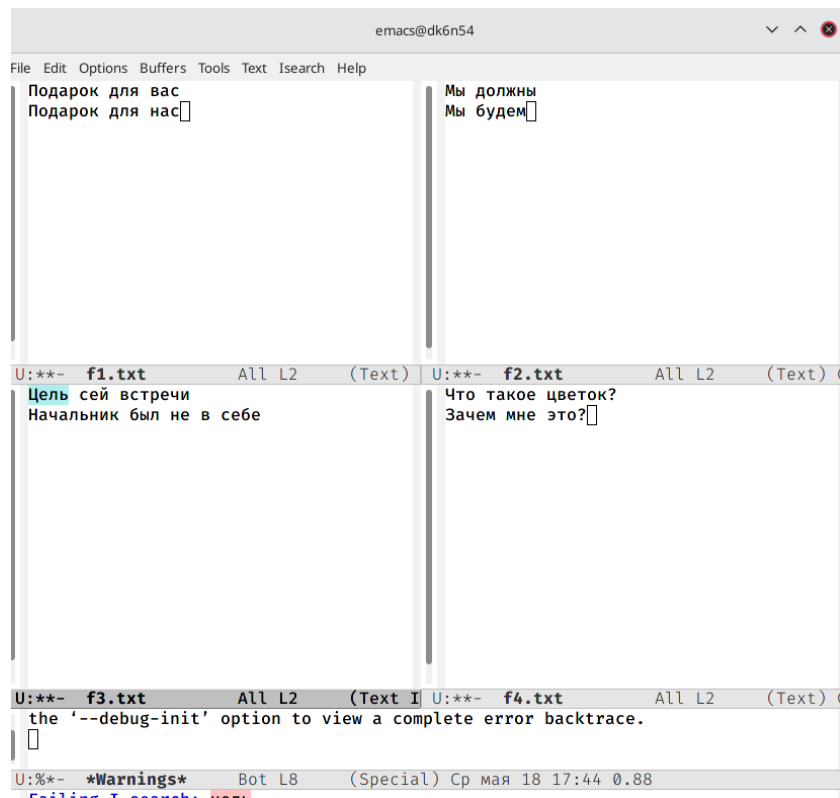


Рис. 1.22: Другой режим поиска.

2 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 XWindow – разные вещи. Одно окно XWindow-

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). С-с | сначала, удерживая «ctrl», нажимаю «с», после – отпускаю обе клавиши и нажимаю «|» С-сC-| 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 неудобным.

3 Выводы

Вывод: в ходе выполнения лабораторной работы я познакомился с операционной системой Linux: получил практические навыки работы с редактором Emacs.