

Лабораторная работа №10

Отчет

Бондарь Алексей Олегович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6

List of Tables

List of Figures

2.1	Emacs	6
2.2	Файл lab07.sh	7
2.3	Текст	7
2.4	Ctrl-k	8
2.5	Ctrl-y	9
2.6	Ctrl-space	9
2.7	Ctrl-y	10
2.8	Ctrl-w	11
2.9	Ctrl-/	12
2.10	Ctrl-a	13
2.11	Ctrl a	14
2.12	Ctrl-e	15
2.13	Alt-<	16
2.14	Alt->	17
2.15	«Ctrl-x»«Ctrl-b»	18
2.16	Выбор буфера	19
2.17	Ctrl-x 0	20
2.18	Переключение	20
2.19	Другой буфер	20
2.20	Два окна	21
2.21	Три окна	21
2.22	Четыре окна	22
2.23	Создание файлов	22
2.24	Открытие буферов	23
2.25	Режим поиска	23
2.26	Замена результата - начальное значение	24
2.27	Замена результата - конечное значение	24

1 Цель работы

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

2 Выполнение лабораторной работы

1. Откроем редактор Emacsc помощью команды «emacs&».(рис. 2.1)



Figure 2.1: Emacsc

2. Создадим файл lab10.sh с помощью комбинации «Ctrl-x» «Ctrl-f».(рис. 2.2)



Figure 2.2: Файл lab07.sh

3. В открывшемся буфере наберем необходимый текст (рис. 2.3)



Figure 2.3: Текст

Сохраним файл с помощью комбинации «Ctrl-x»«Ctrl-s». 4. 1) Вырежем одной командой целую строку («Ctrl-k»).(рис. 2.4)

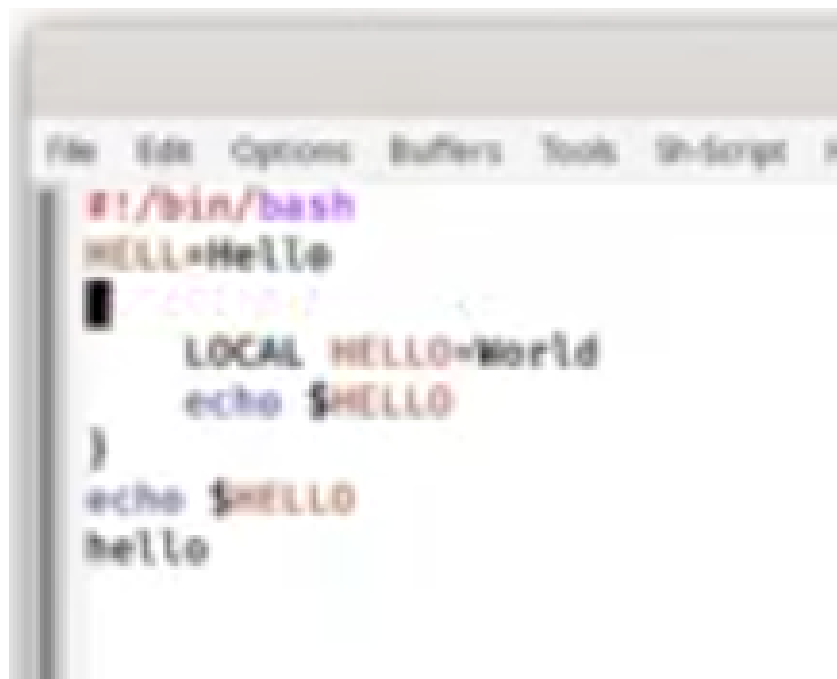
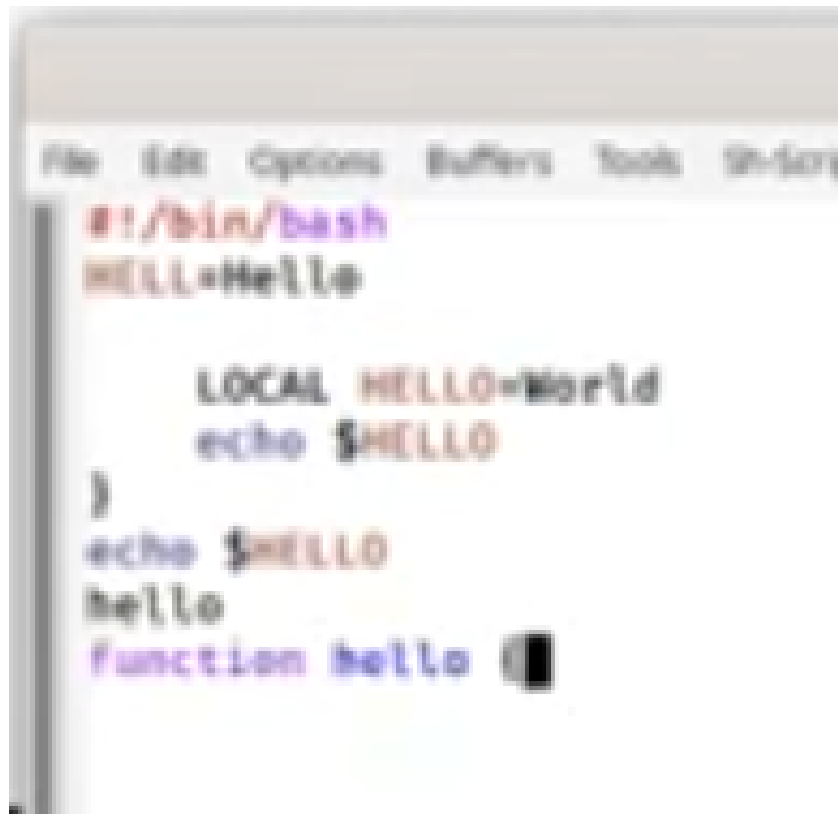


Figure 2.4: Ctrl-k

2) Вставимэту строку в конец файла («Ctrl-y»).(рис. 2.5)

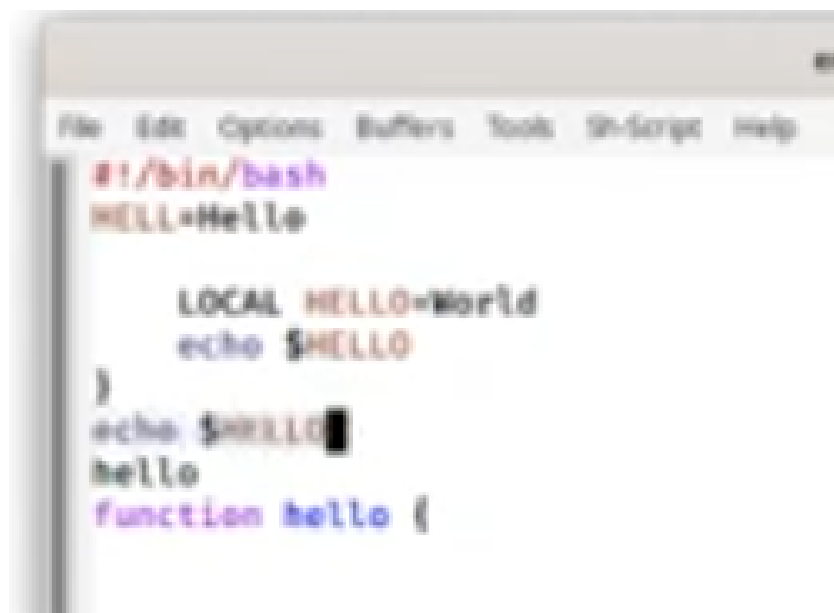


```
File Edit Options Buffers Tools Shell Script Help
#!/bin/bash
HELLO=Hello

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

Figure 2.5: Ctrl-y

3) Выделим область текста («Ctrl-space»).(рис. 2.6)



```
File Edit Options Buffers Tools Shell Script Help
#!/bin/bash
HELLO=Hello

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

Figure 2.6: Ctrl-space

- 4) Скопируем область в буфер обмена («Alt-w»).
- 5) Вставим область в конец файла («Ctrl-y»).(рис. 2.7)

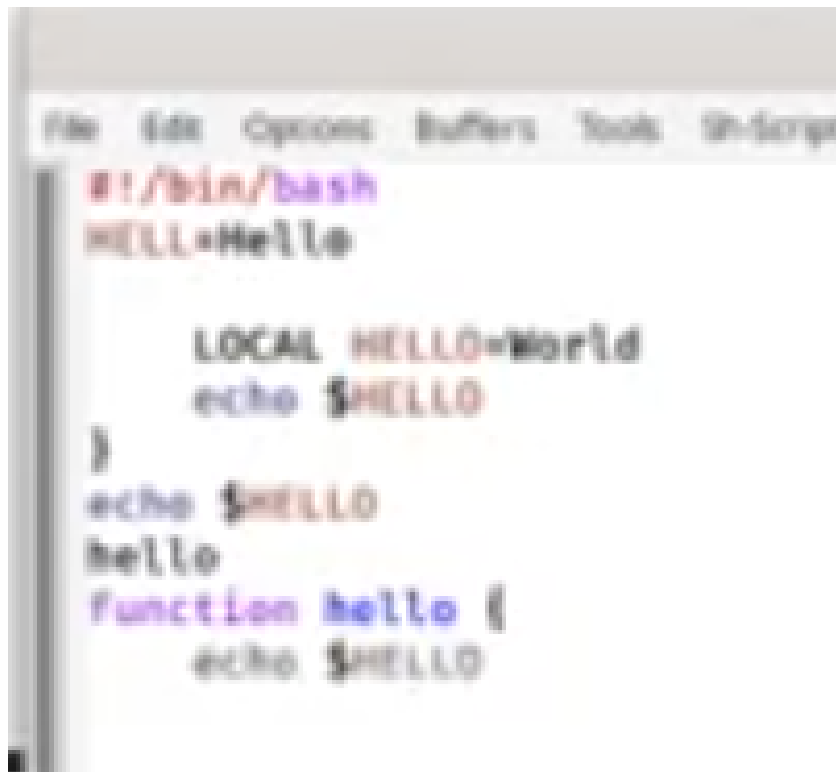
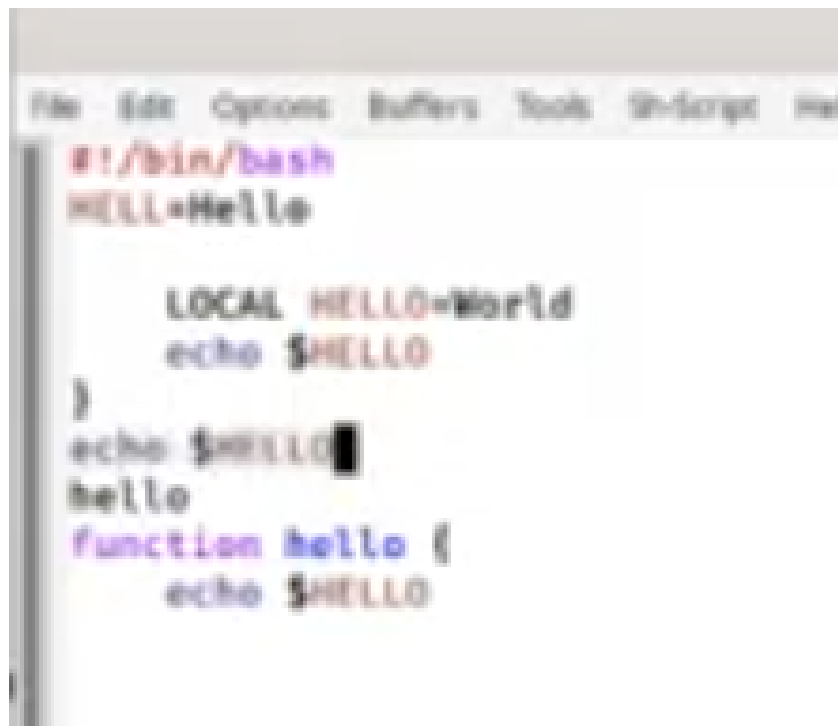


Figure 2.7: Ctrl-y

- 6) Вновь выделим эту область («Ctrl-space») и на этот раз вырежем её («Ctrl-w»).(рис. 2.8)

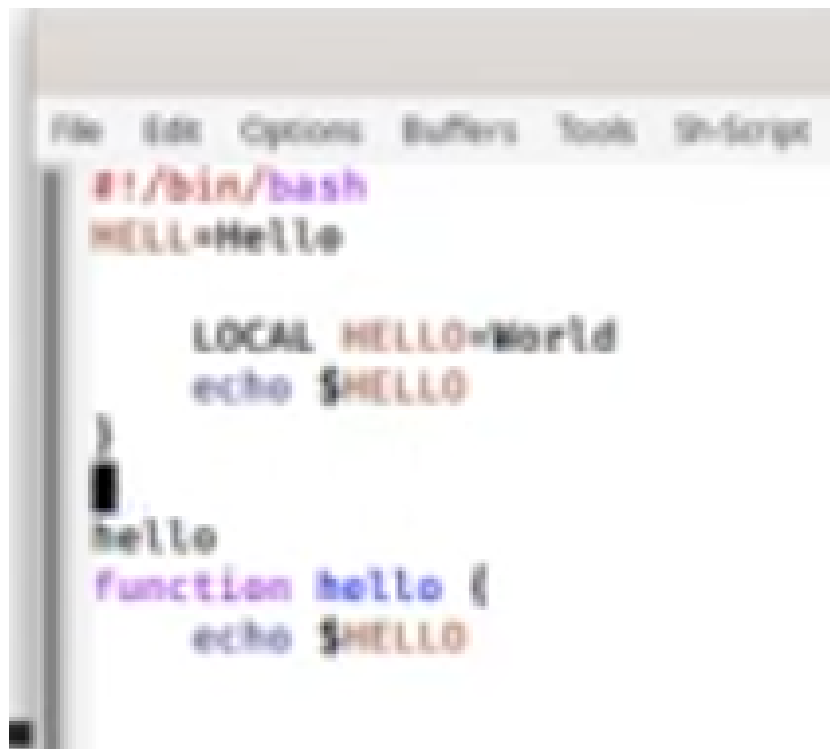


```
File Edit Options Buffers Tools Shell Script Help
#!/bin/bash
HELL=Hello

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

Figure 2.8: Ctrl-w

7) Отменим последнее действие («Ctrl-/»).(рис. 2.9)



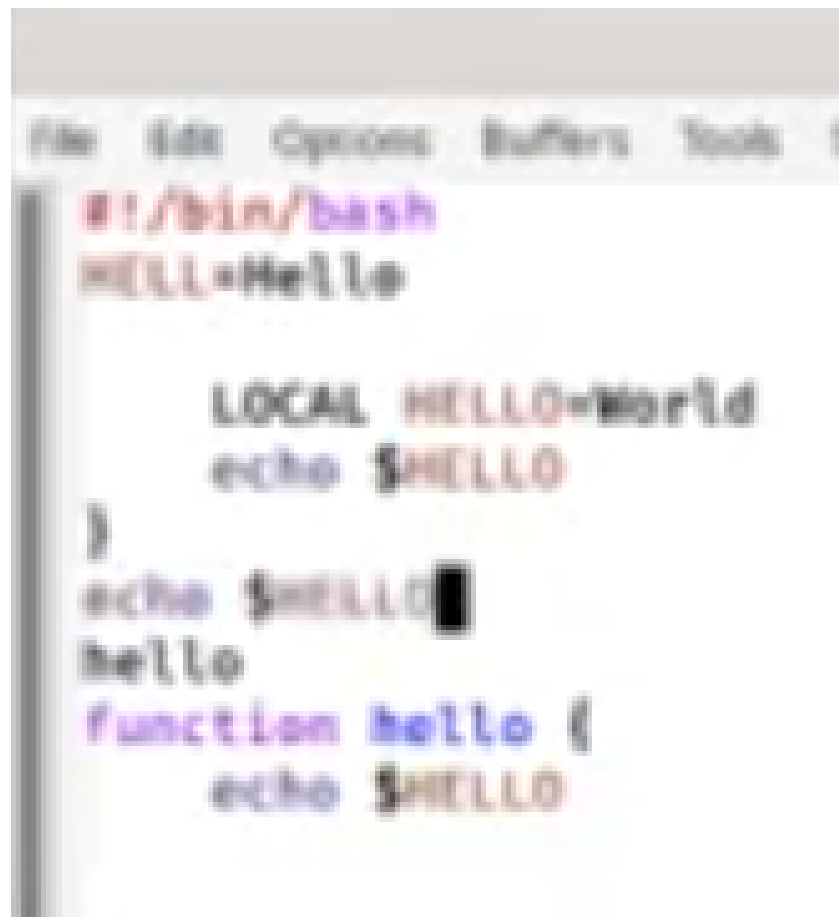
```
File Edit Options Buffers Tools Shell-script
#!/bin/bash
HELLO=Hello

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

Figure 2.9: Ctrl-/

5.

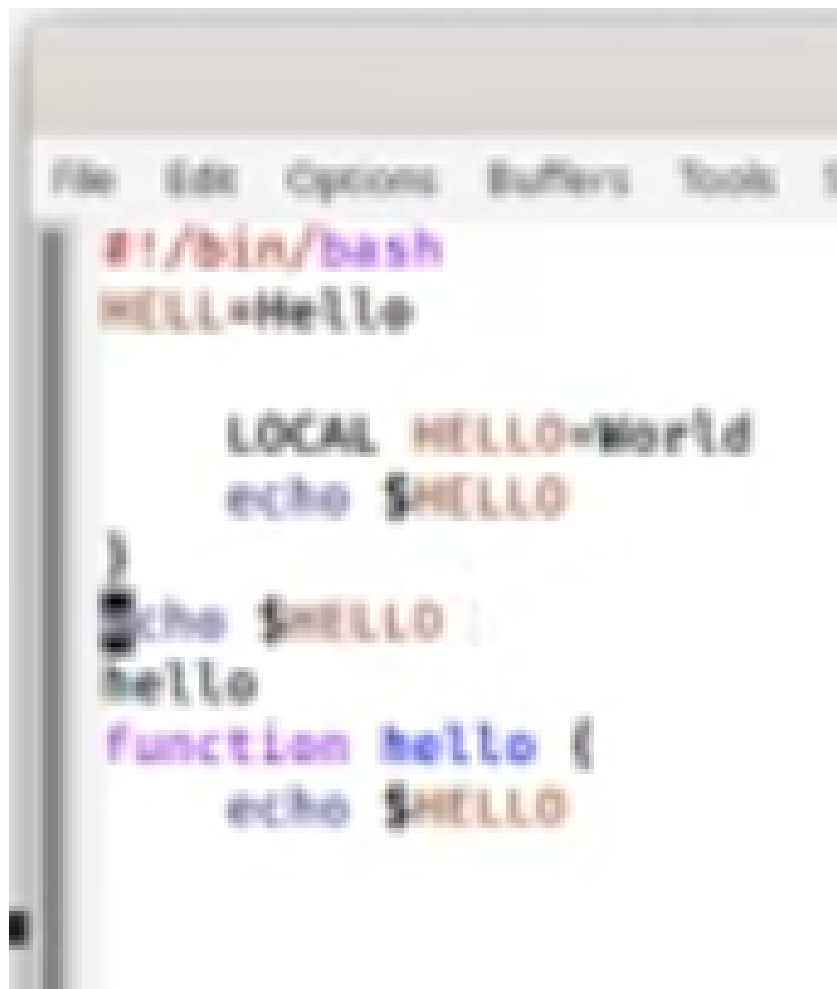
1) Переместим курсор в начало строки («Ctrl-a»).(рис. 2.10) и (рис. 2.11)



```
#!/bin/bash
HELLO=Hello

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

Figure 2.10: Ctrl-a

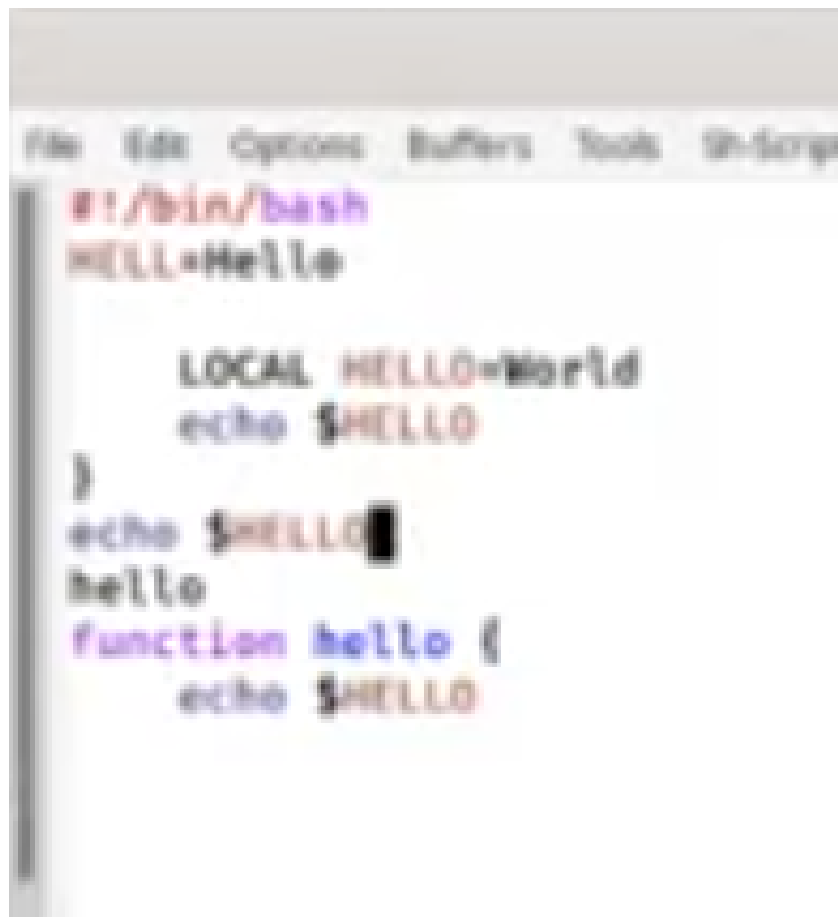


```
File Edit Options Buffers Tools S
#!/bin/bash
HELLO=Hello

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

Figure 2.11: Ctrl a

2) Переместим курсор в конец строки («Ctrl-e»). (рис. 2.12)

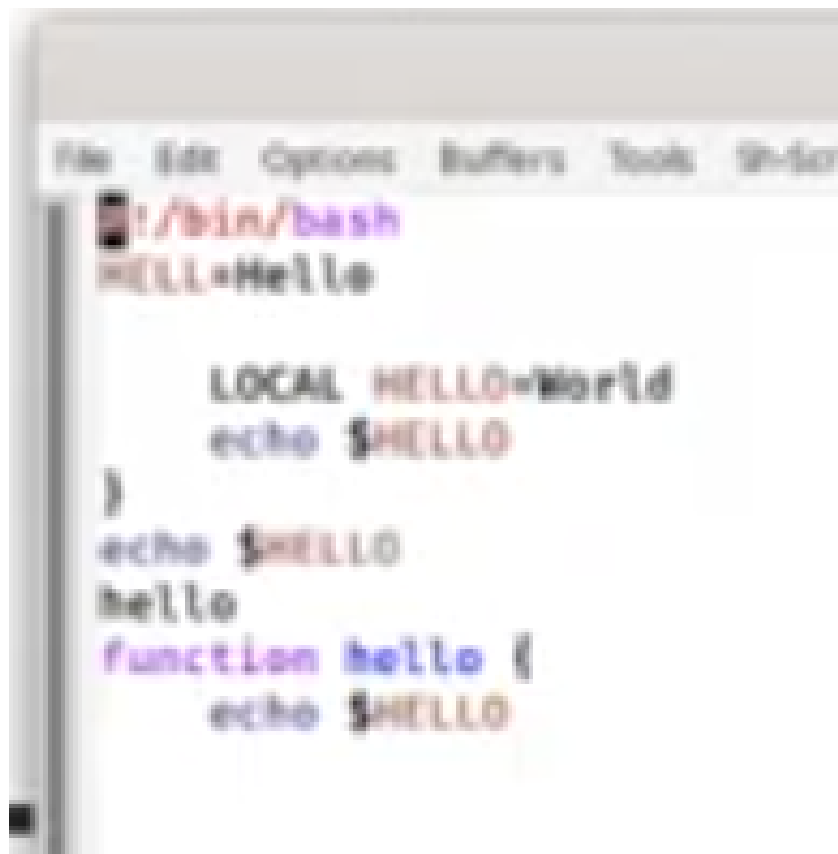


```
File Edit Options Buffers Tools Shell-Script
#!/bin/bash
HELL=Hello

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

Figure 2.12: Ctrl-e

3) Переместим курсор в начало буфера («Alt-<»).(рис. 2.13)

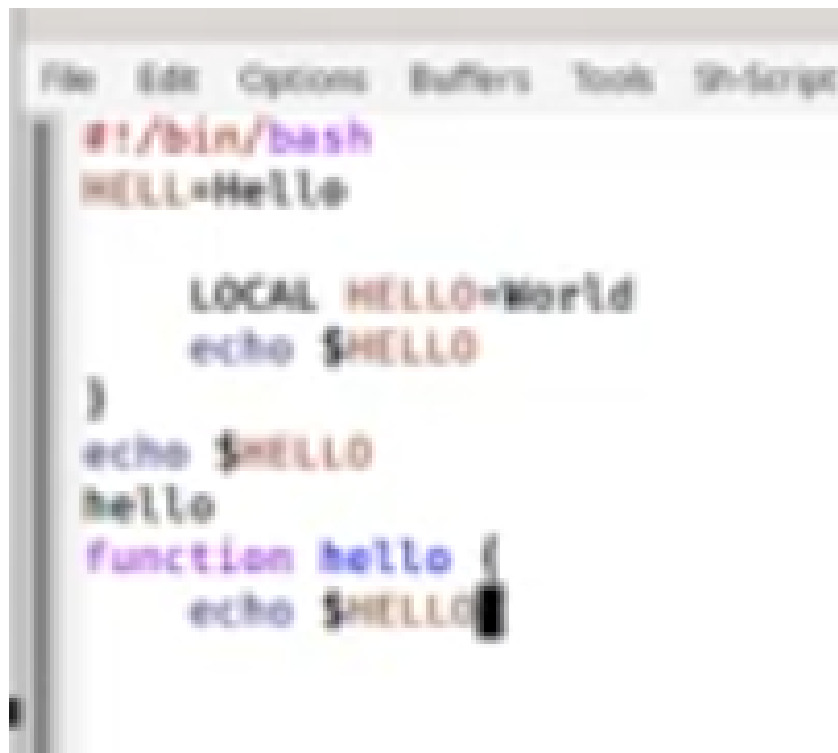


```
File Edit Options Buffers Tools Window
#!/bin/bash
HELLO=Hello

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

Figure 2.13: Alt-<

4) Переместим курсор в конец буфера («Alt->»). (рис. 2.14)



```
File Edit Options Buffers Tools Shell-script
#!/bin/bash
HELLO=Hello

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

Figure 2.14: Alt->

6.

1) Выведем список активных буферов на экран («Ctrl-x»«Ctrl-b»).(рис. 2.15)



Figure 2.15: «Ctrl-x»«Ctrl-b»

- 2) Переместимся во вновь открытое окно («Ctrl-x o») со списком открытых буферов и переключимся на другой буфер (для этого необходимо нажать на «enter» после выбора необходимого буфера) .(рис. 2.16)



Figure 2.16: Выбор буфера

3) Закроем это окно («Ctrl-x 0»).(рис. 2.17)



Figure 2.17: Ctrl-x 0

- 4) Теперь вновь переключимся между буферами, но уже без вывода их списка на экран («Ctrl-x b»).(рис. 2.18) и (рис. 2.19)

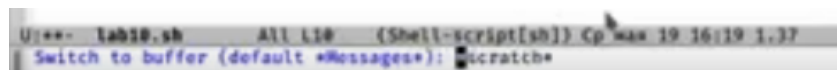


Figure 2.18: Переключение



Figure 2.19: Другой буфер

7.

- 1) Поделитесь фрейм на 4 части: разделите фрейм на два окна по вертикали («Ctrl-x 3»), а затем каждое из этих окон на две части по горизонтали («Ctrl-x 2»).(рис. 2.20) , (рис. 2.21) , (рис. 2.22)

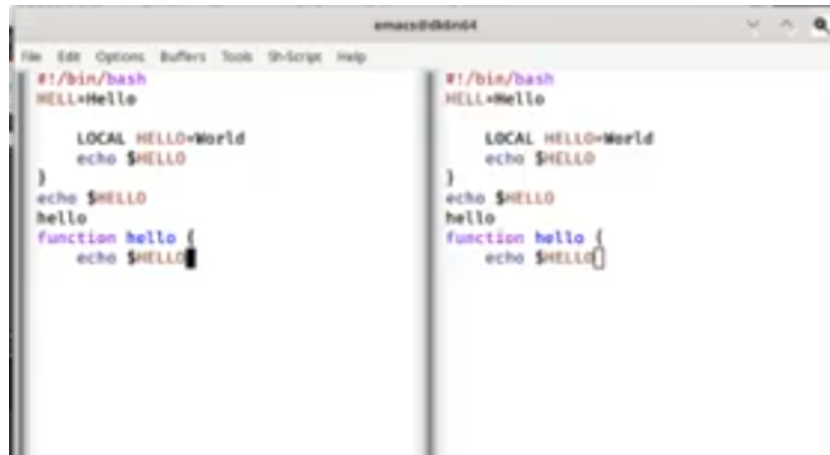


Figure 2.20: Два окна



Figure 2.21: Три окна

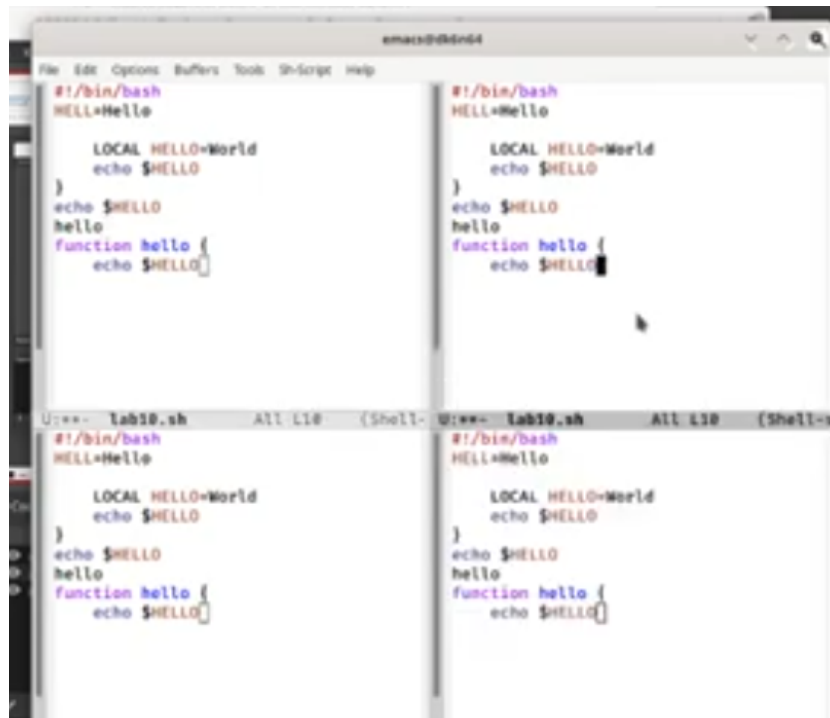


Figure 2.22: Четыре окна

- 2) В каждом из четырёх созданных окон откроем новый буфер (файл) и введем несколько строк текста. Для этого предварительно создадим эти файлы с помощью команд «touch example1.txt», «touch example2.txt», «touch example3.txt», «touch example4.txt». (рис. 2.23) и (рис. 2.24)

```

labondar-j@b6k6n64 - $ touch example1.txt
labondar-j@b6k6n64 - $ touch example2.txt
labondar-j@b6k6n64 - $ touch example3.txt
labondar-j@b6k6n64 - $ touch example4.txt
labondar-j@b6k6n64 - $

```

Figure 2.23: Создание файлов

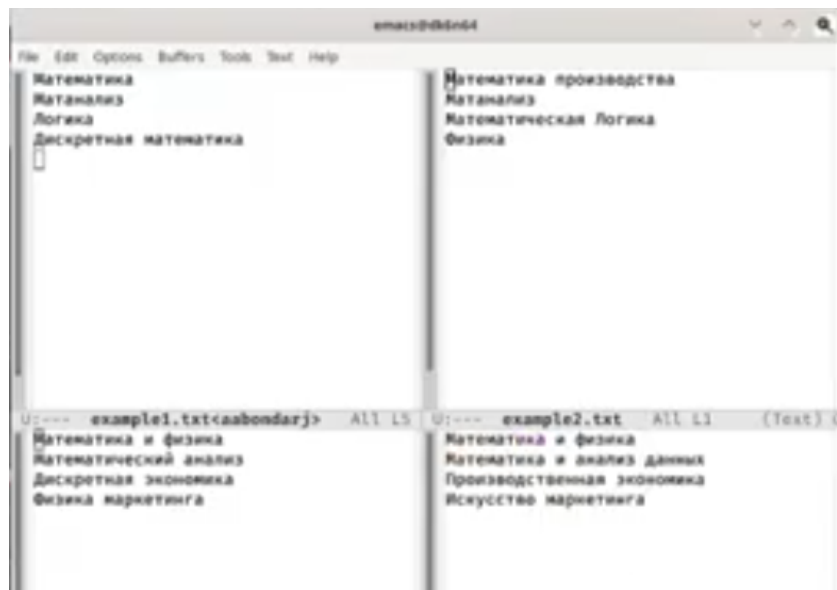


Figure 2.24: Открытие буферов

8.

- 1) Переключимся в режим поиска («Ctrl-s») и найдем несколько слов, присутствующих в тексте.(рис. 2.25)

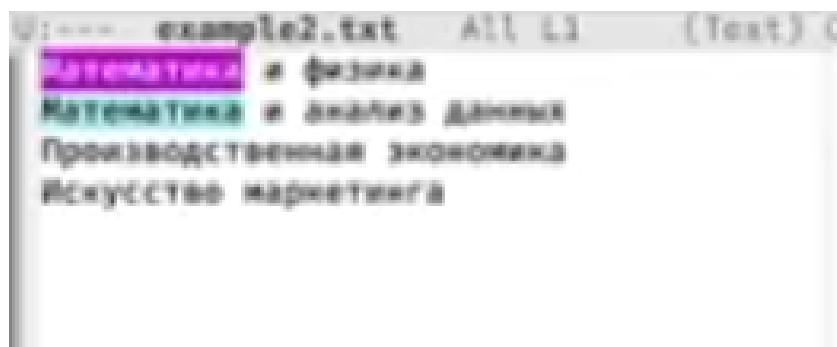


Figure 2.25: Режим поиска

- 2) Переключимся между результатами поиска, нажимая «Ctrl-s».
- 3) Выйдем из режима поиска, нажав «Ctrl-g»
- 4) Перейдем в режим поиска и замены («Alt-%»), введем текст, который следует найти и заменить, нажмем «enter», затем введем текст для замены. После то-

го как будут подсвечены результаты поиска, нажмем «!» для подтверждения замены.(рис. 2.26) и (рис. 2.27)

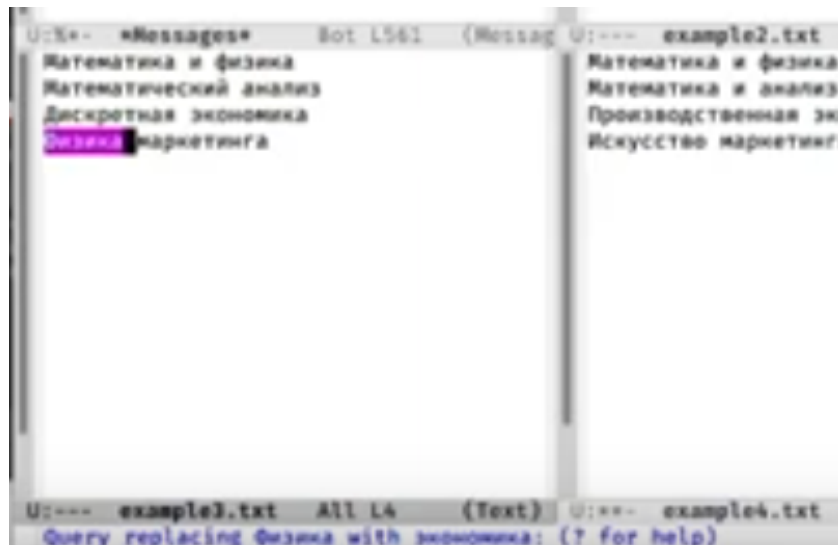


Figure 2.26: Замена результата - начальное значение



Figure 2.27: Замена результата - конечное значение

- 5) Пробуем другой режим поиска, нажав «Alt-so» Данный вид поиска отличается от обычного тем, что тут считывается строка поиска, которая трактуется как регулярное выражение, и не осуществляется поиск точного совпадения в тексте буфера. Регулярное выражение – это образец, который обозначает набор строк, возможно, и неограниченный набор. # 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. 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 поделить 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. # Выводы

В ходе выполнения данной лабораторной работы я познакомился с операционной системой Linux и получил практические навыки работы с редактором Emacs.