

Отчет по лабораторной работе №10

Дисциплина: Операционные системы

Пермякова Елизавета Евгеньевна

Содержание

1	Цель работы	4
2	Задачи	5
3	Выполнение лабораторной работы	6
4	Контрольные вопросы	21
5	Выводы	24
6	Библиография	25

List of Figures

3.1	Текстовый редактор emacs	6
3.2	Создание файла lab07.sh	7
3.3	Ввод текста	7
3.4	Вырезаем одну из строк	8
3.5	Вставка строки в конец	8
3.6	Выделяем текст	8
3.7	Вставка текста в конец	9
3.8	Вырезаем эту область текста	9
3.9	Отмена последнего действия	9
3.10	Курсор в начале строки	10
3.11	Курсор в конце строки	10
3.12	Курсор в начало буфера	11
3.13	Курсор в конец буфера	11
3.14	Список активных буферов	12
3.15	Переключение на другой буфер	13
3.16	Заккрытие окна	14
3.17	Переход к другому буферу	14
3.18	Буфер <i>scratch</i>	14
3.19	Деление фрейма на 4 части	15
3.20	Создание 4 файлов	15
3.21	Переход к новому буферу	15
3.22	Новые буферы	16
3.23	Режим поиска	16
3.24	Результаты поиска	16
3.25	Переключение между результатами поиска	17
3.26	Ввод текста, который нужно заменить	17
3.27	Ввод текста для замены	18
3.28	Подтверждение замены	18
3.29	Итоговая замена	19
3.30	Другой режим поиска	19
3.31	Результаты поиска	20

1 Цель работы

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

2 Задачи

1. Познакомиться с текстовым редактором etacs.
2. Изучить команды управления (для перемещения курсора, работы с текстом, работы с выделенной областью текста, для поиска и замены).
3. В ходе работы использовать эти команды и интерпретировать их вывод.
4. Выполнить отчет.

3 Выполнение лабораторной работы

1. Откроем редактор Emacs с помощью команды «emacs &». (рис. 3.1)

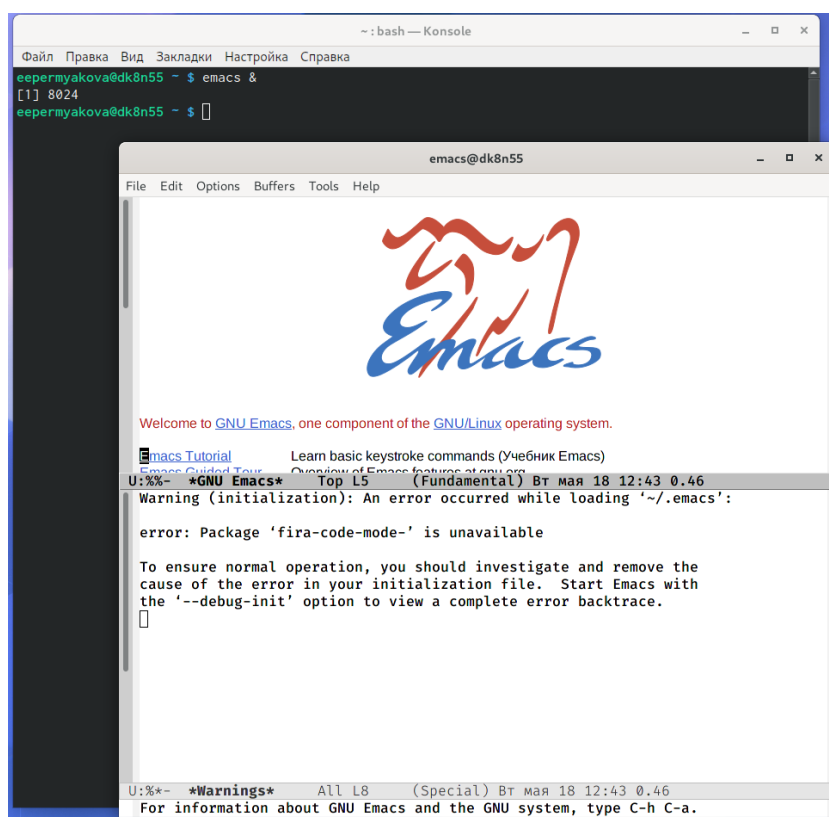


Figure 3.1: Текстовый редактор emacs

2. Создадим файл lab07.sh с помощью комбинации «Ctrl-x»«Ctrl-f». (рис. 3.2)

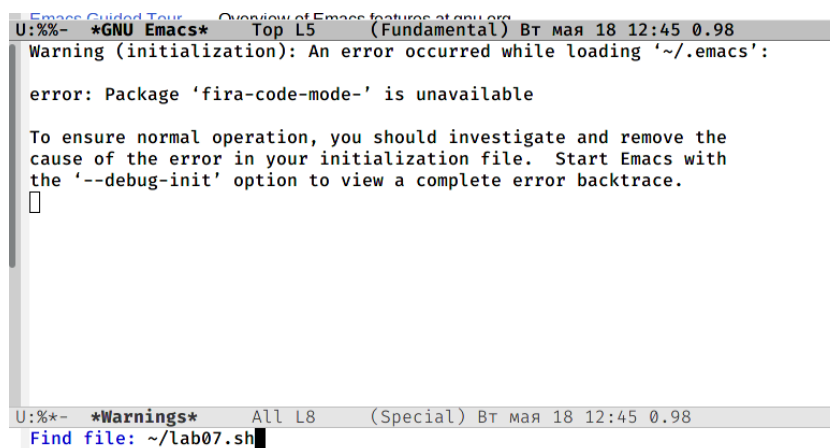


Figure 3.2: Создание файла lab07.sh

3. В открывшемся буфере наберем необходимый текст. (рис. 3.3)

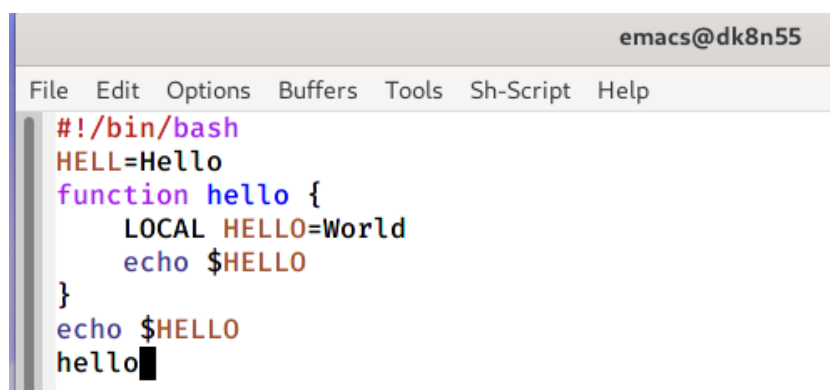
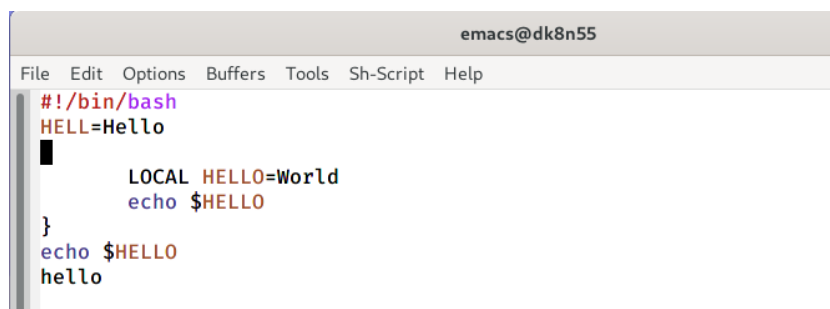


Figure 3.3: Ввод текста

4. Сохраним файл с помощью комбинации «Ctrl-x»«Ctrl-s».

5. Проделаем с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш.

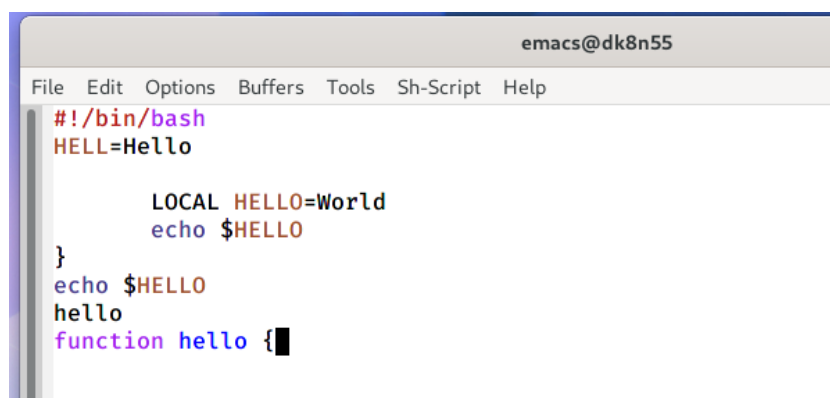
5.1. Вырежем одной командой целую строку («Ctrl-k»). (рис. 3.4)



```
emacs@dk8n55
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
```

Figure 3.4: Вырезаем одну из строк

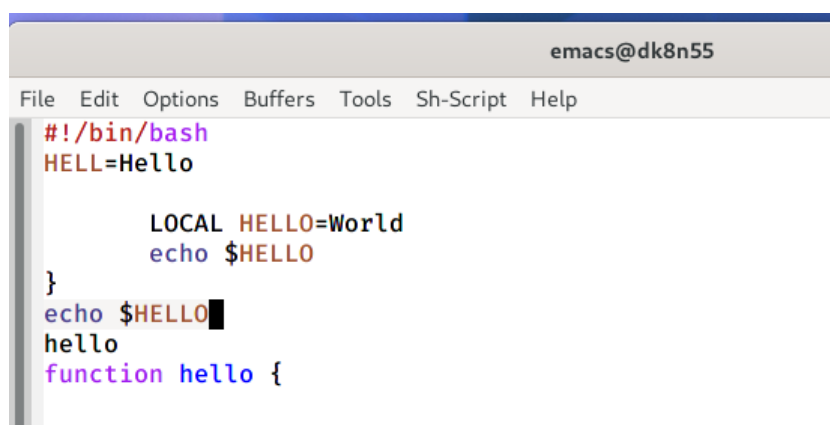
5.2. Вставим эту строку в конец файла («Ctrl-y»). (рис. 3.5)



```
emacs@dk8n55
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
```

Figure 3.5: Вставка строки в конец

5.3. Выделим область текста («Ctrl-space»). (рис. 3.6)

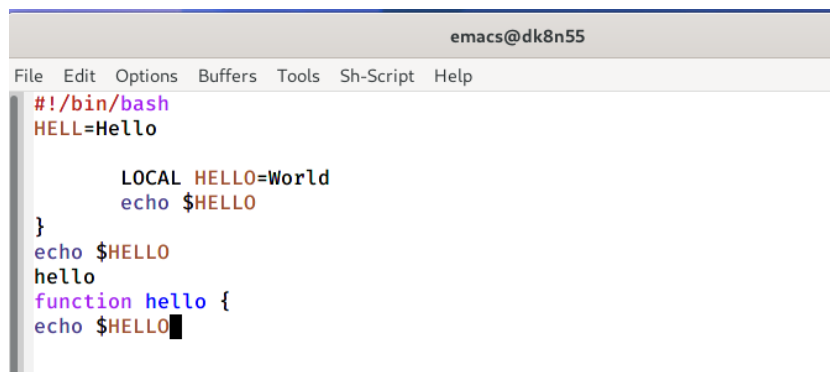


```
emacs@dk8n55
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello
    LOCAL HELLO=World
    echo $HELLO
}
echo $HELLO
hello
function hello {
```

Figure 3.6: Выделяем текст

5.4. Скопируем область в буфер обмена («Alt-w»).

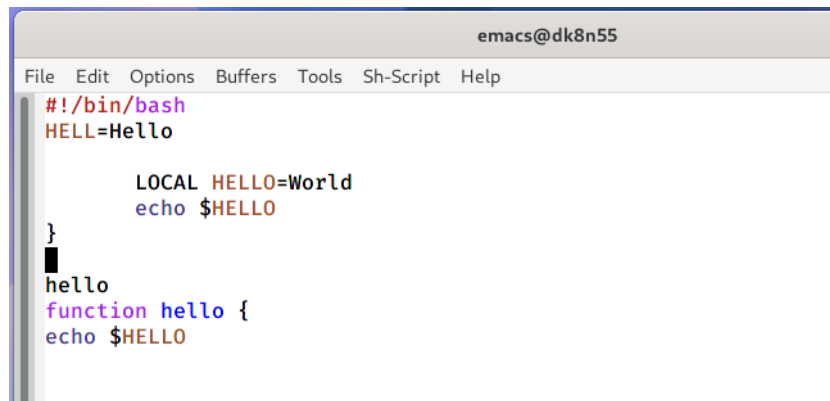
5.5. Вставим область в конец файла («Ctrl-y»). (рис. 3.7)



The screenshot shows the Emacs editor window titled 'emacs@dk8n55'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The code in the buffer is a shell script: `#!/bin/bash`, `HELL=Hello`, a blank line, `LOCAL HELLO=World`, `echo $HELLO`, `}`, `echo $HELLO`, `hello`, `function hello {`, and `echo $HELLO`. A cursor is positioned at the end of the last line, indicating that text has been pasted there.

Figure 3.7: Вставка текста в конец

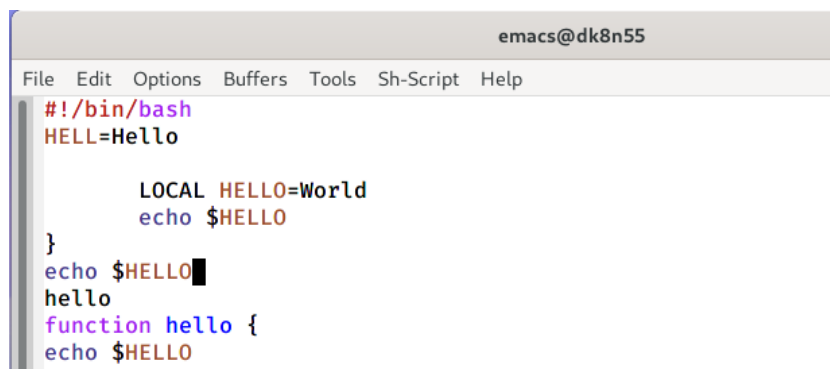
5.6. Вновь выделим эту область («Ctrl-space») и на этот раз вырежем её («Ctrl-w»). (рис. 3.8)



The screenshot shows the Emacs editor window titled 'emacs@dk8n55'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The code in the buffer is a shell script: `#!/bin/bash`, `HELL=Hello`, a blank line, `LOCAL HELLO=World`, `echo $HELLO`, `}`, `hello`, `function hello {`, and `echo $HELLO`. A cursor is positioned at the end of the last line, indicating that text has been pasted there.

Figure 3.8: Вырезаем эту область текста

5.7. Отменим последнее действие («Ctrl-/»). (рис. 3.9)

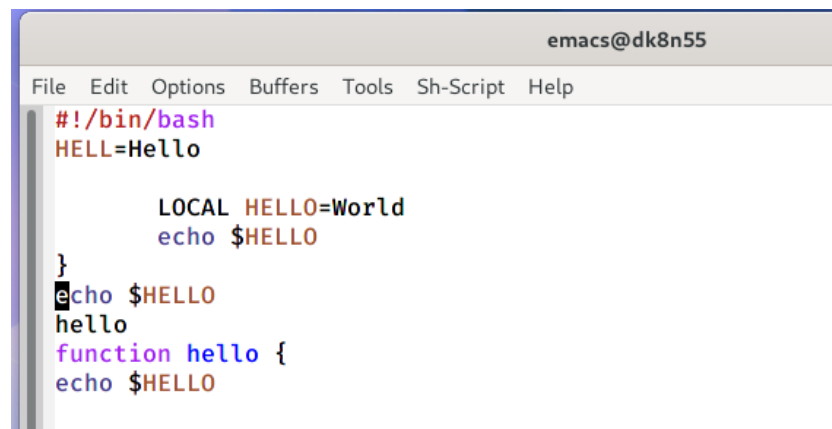


The screenshot shows the Emacs editor window titled 'emacs@dk8n55'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The code in the buffer is a shell script: `#!/bin/bash`, `HELL=Hello`, a blank line, `LOCAL HELLO=World`, `echo $HELLO`, `}`, `echo $HELLO`, `hello`, `function hello {`, and `echo $HELLO`. A cursor is positioned at the end of the last line, indicating that text has been pasted there.

Figure 3.9: Отмена последнего действия

6. Учимся использовать команды по перемещению курсора.

6.1. Переместим курсор в начало строки («Ctrl-a»). (рис. 3.10)

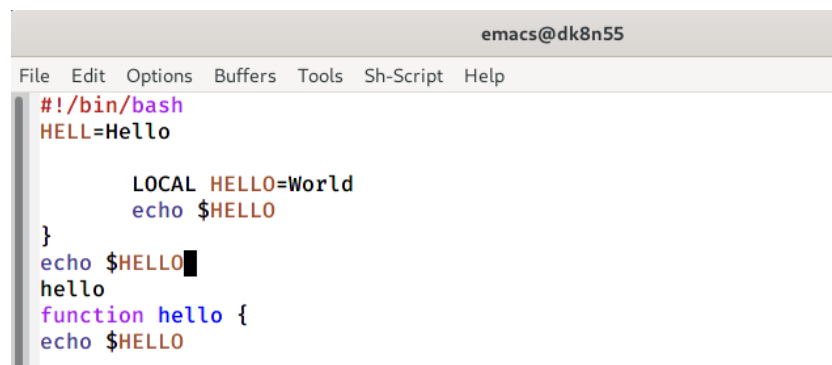


```
emacs@dk8n55
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello

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

Figure 3.10: Курсор в начале строки

6.2. Переместим курсор в конец строки («Ctrl-e»). (рис. 3.11)

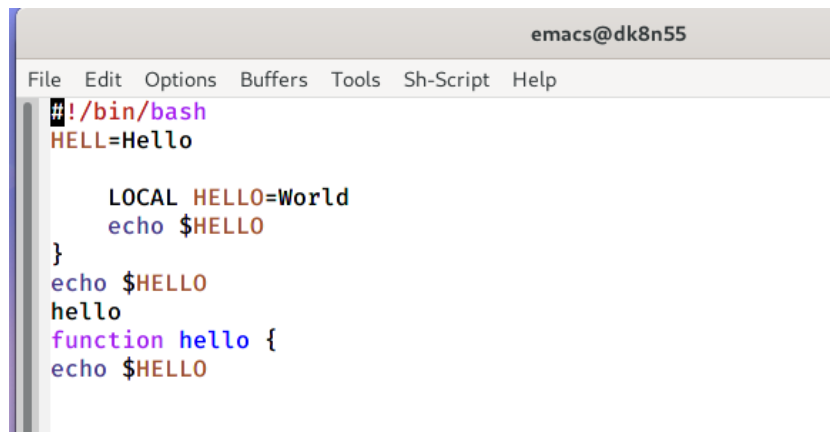


```
emacs@dk8n55
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
HELL=Hello

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

Figure 3.11: Курсор в конце строки

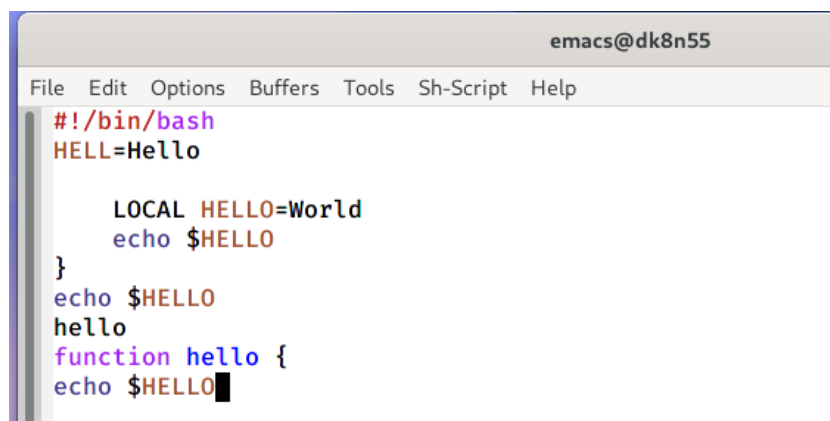
6.3. Переместим курсор в начало буфера («Alt-<»). (рис. 3.12)



The screenshot shows the Emacs editor window titled 'emacs@dk8n55'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The buffer content is a shell script: `#!/bin/bash`, `HELL=Hello`, `LOCAL HELLO=World`, `echo $HELLO`, `}`, `echo $HELLO`, `hello`, `function hello {`, and `echo $HELLO`. The cursor is positioned at the start of the first line, `#!/bin/bash`.

Figure 3.12: Курсор в начало буфера

6.4. Переместим курсор в начало буфера («Alt->»). (рис. 3.13)



The screenshot shows the Emacs editor window titled 'emacs@dk8n55'. The menu bar includes 'File', 'Edit', 'Options', 'Buffers', 'Tools', 'Sh-Script', and 'Help'. The buffer content is the same shell script as in Figure 3.12. The cursor is now positioned at the end of the last line, `echo $HELLO`.

Figure 3.13: Курсор в конец буфера

7. Управление буферами.

7.1. Выведем список активных буферов на экран («Ctrl-x»«Ctrl-b»). (рис. 3.14)

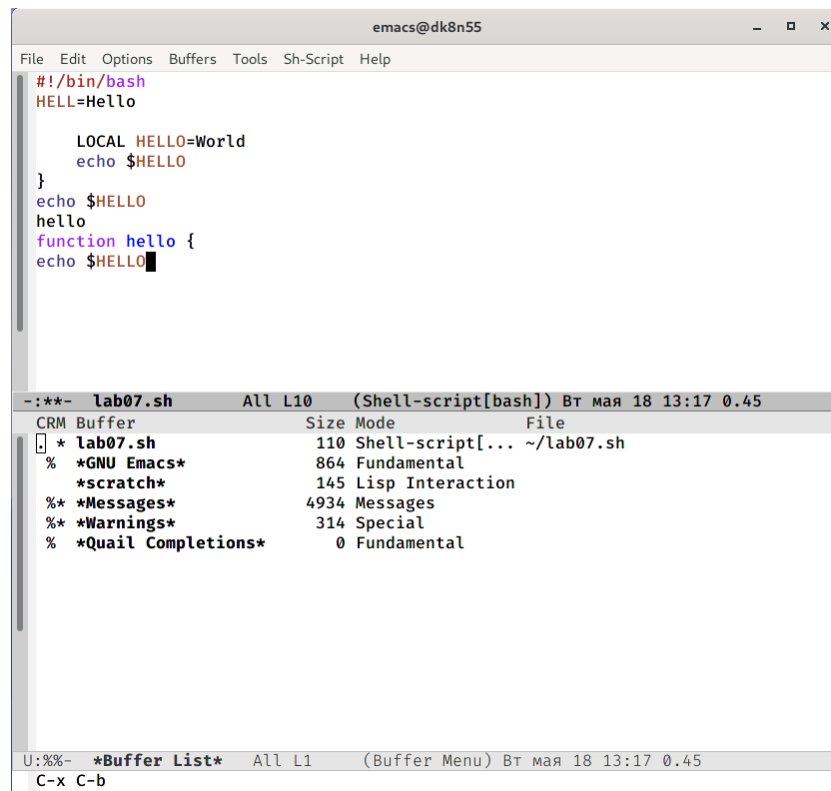


Figure 3.14: Список активных буферов

7.2. Переместимся во вновь открытое окно («Ctrl-хо») со списком открытых буферов и переключимся на другой буфер (для этого необходимо нажать на «enter» после выбора необходимого буфера). (рис. 3.15)

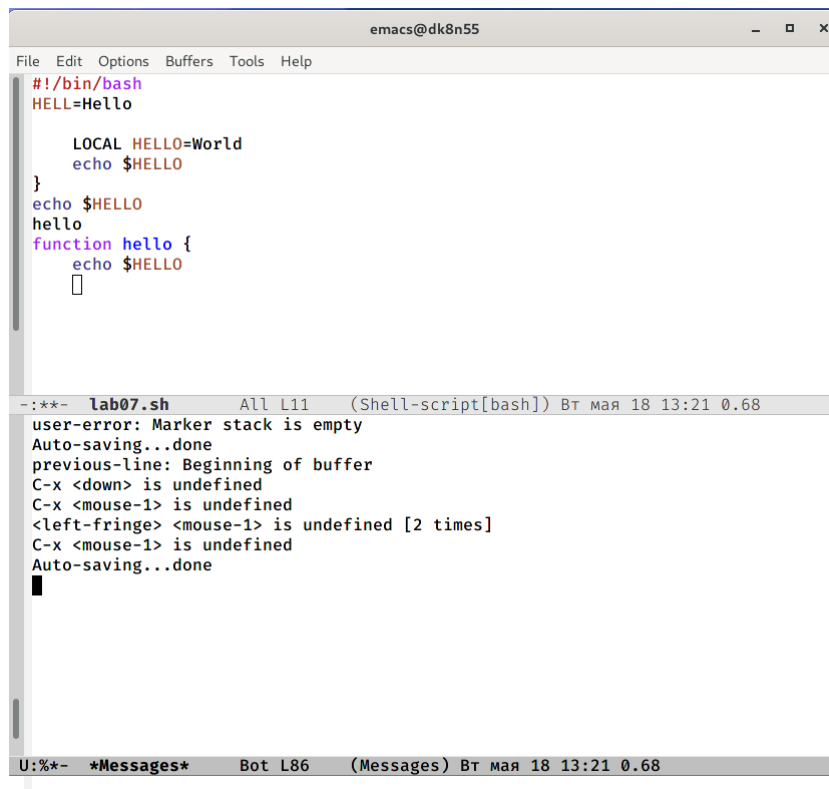


Figure 3.15: Переключение на другой буфер

7.3. Закроем это окно. (рис. 3.16)

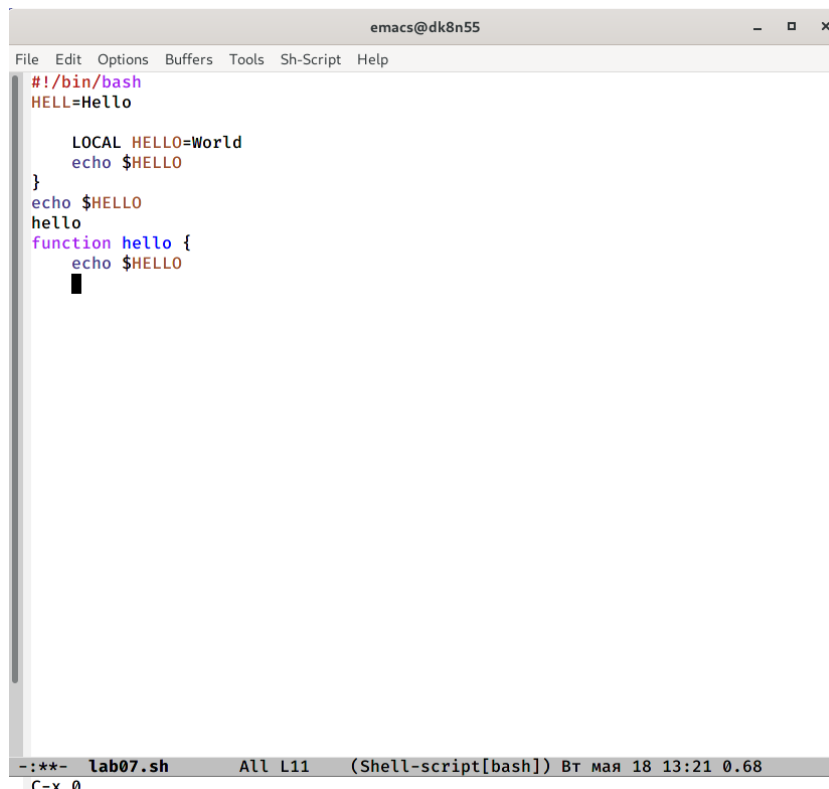


Figure 3.16: Заккрытие окна

7.4. Теперь вновь переключимся между буферами, но уже без вывода их списка на экран («Ctrl-x b»). (рис. 3.17) (рис. 3.18)

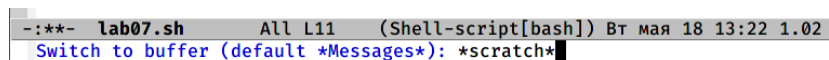


Figure 3.17: Переход к другому буферу

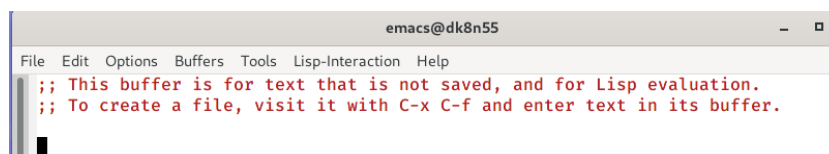


Figure 3.18: Буфер *scratch*

8. Управление окнами.

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

(рис. 3.19)



Figure 3.19: Деление фрейма на 4 части

8.2. 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. 3.20) (рис. 3.21) (рис. 3.22)

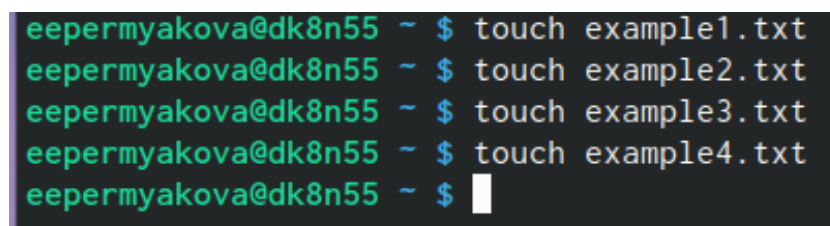


Figure 3.20: Создание 4 файлов

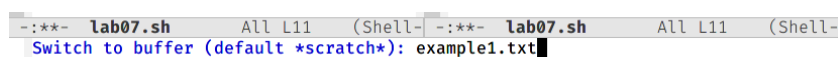


Figure 3.21: Переход к новому буферу



Figure 3.22: Новые буферы

9. Режим поиска.

9.1. Переключимся в режим поиска («Ctrl-s») и найдем несколько слов, присутствующих в тексте. (рис. 3.23) (рис. 3.24)

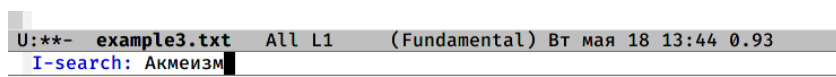


Figure 3.23: Режим поиска

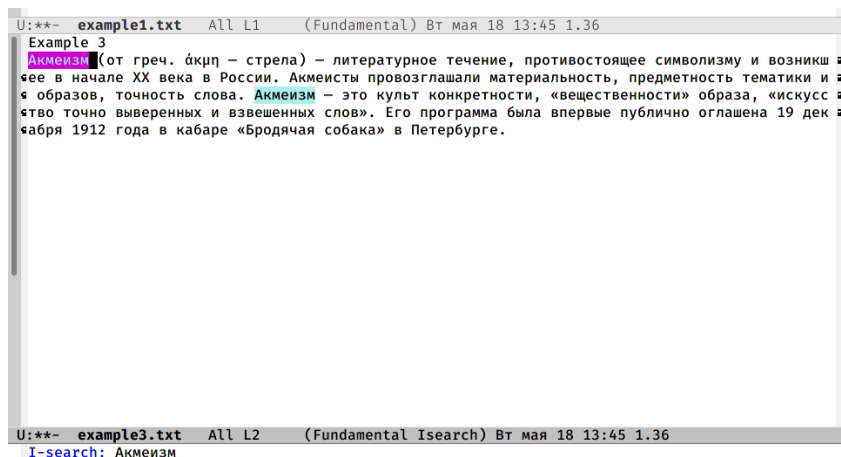


Figure 3.24: Результаты поиска

9.2. Переключимся между результатами поиска, нажимая «Ctrl-s». (рис. 3.25)

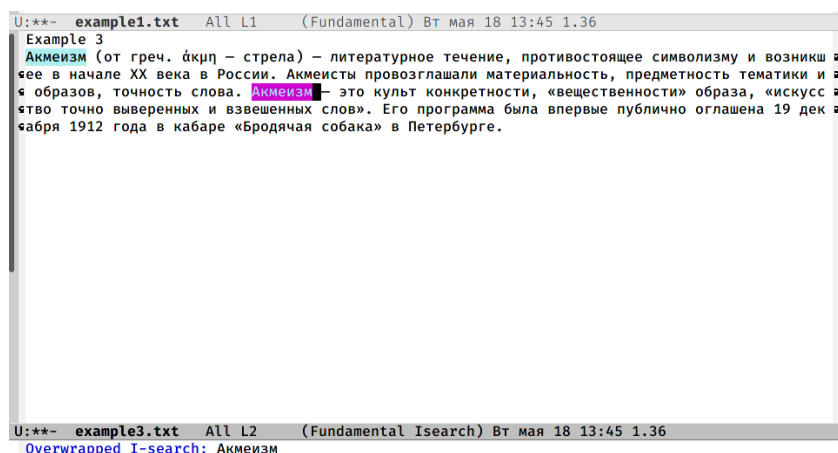


Figure 3.25: Переключение между результатами поиска

9.3. Выйдем из режима поиска, нажав «Ctrl-g».

9.4. Перейдем в режим поиска и замены («Alt-%»), введем текст, который сле-
 дует найти и заменить, нажмем «enter», затем введем текст для замены. После
 того как будут подсвечены результаты поиска, нажмем «!» для подтверждения
 замены. Важно, чтобы курсор находился в начале текста. (рис. 3.26) (рис. 3.27)
 (рис. 3.28) (рис. 3.29)

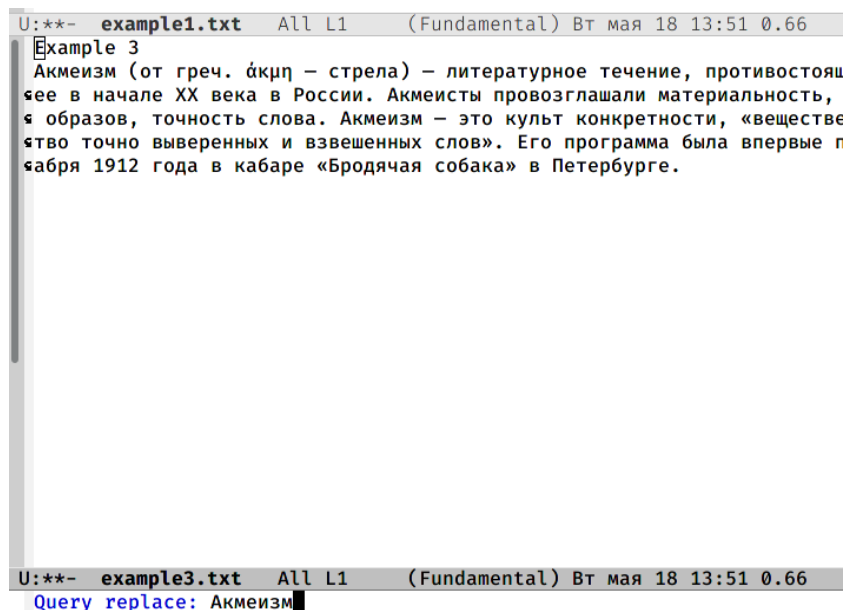


Figure 3.26: Ввод текста, который нужно заменить

```
U:**- example1.txt All L1 (Fundamental) Вт мая 18 13:52 1.21
Example 3
Акмеизм (от греч. ἀκμή — стрела) — литературное течение, противостоящее
ее в начале XX века в России. Акмеисты провозглашали материальность, пр
а образов, точность слова. Акмеизм — это культ конкретности, «вещественн
ство точно выверенных и взвешенных слов». Его программа была впервые пуб
абря 1912 года в кабаре «Бродячая собака» в Петербурге.

U:**- example3.txt All L1 (Fundamental) Вт мая 18 13:52 1.21
Query replace Акмеизм with: Модернизм
```

Figure 3.27: Ввод текста для замены

```
U:**- example1.txt All L1 (Fundamental) Вт мая 18 13:52 1.21
Example 3
Акмеизм (от греч. ἀκμή — стрела) — литературное течение, противостоящее символизму и возникш
ее в начале XX века в России. Акмеисты провозглашали материальность, предметность тематики и а
а образов, точность слова. Акмеизм — это культ конкретности, «вещественности» образа, «искусс
ство точно выверенных и взвешенных слов». Его программа была впервые публично оглашена 19 дек
абря 1912 года в кабаре «Бродячая собака» в Петербурге.

U:**- example3.txt All L2 (Fundamental) Вт мая 18 13:52 1.21
Query replacing Акмеизм with Модернизм: (? for help)
```

Figure 3.28: Подтверждение замены

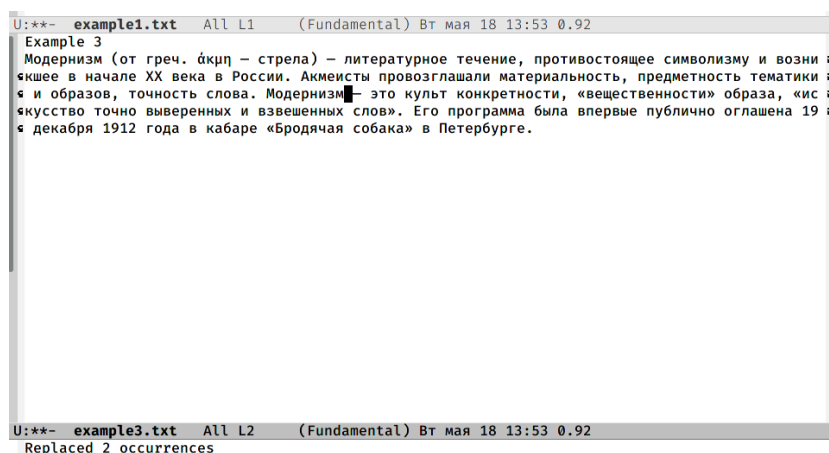


Figure 3.29: Итоговая замена

9.5. Пробуем другой режим поиска, нажав «Alt-s o». Но у меня не работала эта комбинация, поэтому я воспользовалась «Alt-x ossur». (рис. 3.30) (рис. 3.31)

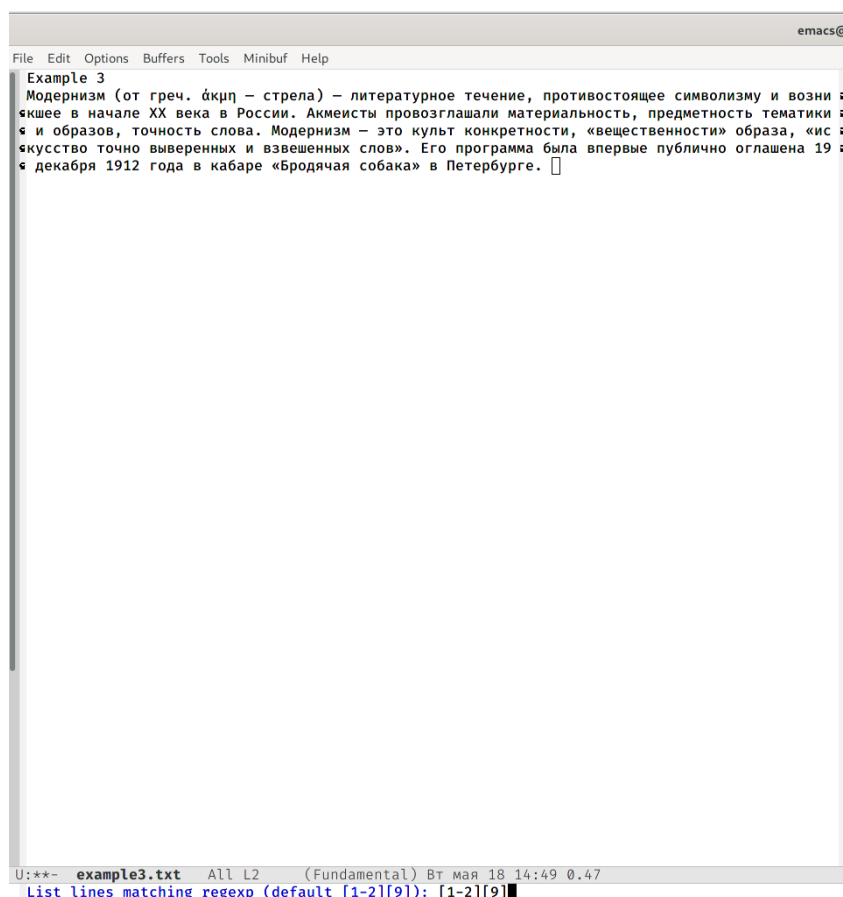


Figure 3.30: Другой режим поиска

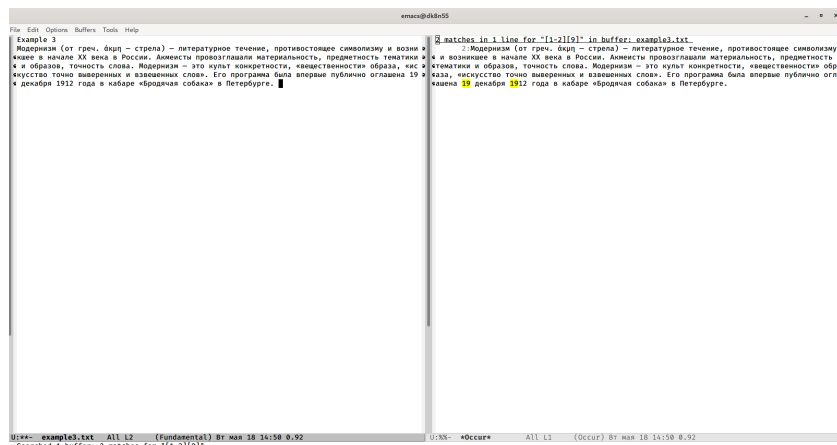


Figure 3.31: Результаты поиска

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.

4 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 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 для меня является редактор emacs, так как в нем проще открывать другие файлы, можно использовать сразу несколько окон, нет «Командного режима», «Режима ввода», «Режима командной строки», которые являются в какой-то степени неудобными и непривычными.

5 Выводы

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

6 Библиография

1. https://esystem.rudn.ru/pluginfile.php/1142084/mod_resource/content/2/006-lab_vi.pdf
2. Кулябов Д.С. Операционные системы: лабораторные работы: учебное пособие / Д.С. Кулябов, М.Н. Геворкян, А.В. Королькова, А.В. Демидова. — М. : Изд-во РУДН, 2016. — 117 с. — ISBN 978-5-209-07626-1 : 139.13; То же [Электронный ресурс]. — URL: <http://lib.rudn.ru/MegaPro2/Download/MObject/6118>.
3. Робачевский А.М. Операционная система UNIX [текст] : Учебное пособие / А.М. Робачевский, С.А. Немнюгин, О.Л. Стесик. — 2-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2005, 2010. — 656 с. : ил. — ISBN 5-94157-538-6 : 164.56. (ЕТ 60)
4. Таненбаум Эндрю. Современные операционные системы [Текст] / Э. Таненбаум. — 2-е изд. — СПб. : Питер, 2006. — 1038 с. : ил. — (Классика Computer Science). — ISBN 5-318-00299-4 : 446.05. (ЕТ 50)