

Отчет по лабораторной работе

Лабораторная работа № 9

Мурзаев Замир Зейнадинович

Содержание

1 Цель работы	5
2 Задание	6
3 Выполнение лабораторной работы	8
4 Ответы на вопросы	25
5 Выводы	26
Список литературы	27

Список иллюстраций

3.1	запуск emacs	8
3.2	C-x C-f	8
3.3	Набор текста	9
3.4	C-k	10
3.5	C-y	10
3.6	C-space	11
3.7	C-y	12
3.8	C-w	13
3.9	C-/	14
3.10	C-a	15
3.11	C-e	16
3.12	M-<	17
3.13	M-<	18
3.14	C-x C-b	19
3.15	C-x	19
3.16	C-x b	20
3.17	C-x 3 C-x 2	20
3.18	Набор текста	21
3.19	C-s	22
3.20	C-s	23
3.21	C-g	23

Список таблиц

1 Цель работы

Цель познакомиться с операционной системой Linux и получить практические навыки работы с редактором Emacs.

2 Задание

1. Открыть emacs.
2. Создать файл lab07.sh с помощью комбинации Ctrl-x Ctrl-f (C-x C-f).
3. Наберите текст
4. Сохранить файл с помощью комбинации Ctrl-x Ctrl-s (C-x C-s).
5. Проделать с текстом стандартные процедуры редактирования, каждое действие должно осуществляться комбинацией клавиш. 5.1. Вырезать одной командой целую строку (C-k). 5.2. Вставить эту строку в конец файла (C-y). 5.3. Выделить область текста (C-space). 5.4. Скопировать область в буфер обмена (M-w). 5.5. Вставить область в конец файла. 5.6. Вновь выделить эту область и на этот раз вырезать её (C-w). 5.7. Отмените последнее действие (C-/).
6. Научитесь использовать команды по перемещению курсора. 6.1. Переместите курсор в начало строки (C-a). 6.2. Переместите курсор в конец строки (C-e). 6.3. Переместите курсор в начало буфера (M-). 6.4. Переместите курсор в конец буфера (M->).
7. Управление буферами. 7.1. Вывести список активных буферов на экран (C-x C-b). 7.2. Переместитесь во вновь открытое окно (C-x) о со списком открытых буферов и переключитесь на другой буфер. 7.3. Закройте это окно (C-x 0). 7.4. Теперь вновь переключайтесь между буферами, но уже без вывода их списка на экран (C-x b).
8. Управление окнами. 8.1. Поделите фрейм на 4 части: разделите фрейм на два окна по вертикали (C-x 3), а затем каждое из этих окон на две части

по горизонтали (С-x 2) 8.2. В каждом из четырёх созданных окон откройте новый буфер (файл) и введите несколько строк текста.

9. Режим поиска 9.1. Переключитесь в режим поиска (C-s) и найдите несколько слов, присутствующих в тексте. 9.2. Переключайтесь между результатами поиска, нажимая C-s. 9.3. Выходите из режима поиска, нажав C-g. 9.4. Перейдите в режим поиска и замены (M-%), введите текст, который следует найти и заменить, нажмите Enter , затем введите текст для замены. После того как будут подсвечены результаты поиска, нажмите ! для подтверждения замены. 9.5. Испробуйте другой режим поиска, нажав M-s о. Объясните, чем он отличается от обычного режима?

3 Выполнение лабораторной работы

1)Открываем emacs (рис. 3.1).



Рис. 3.1: запуск emacs

2)Создаем файл lab09.sh (рис. 3.2).

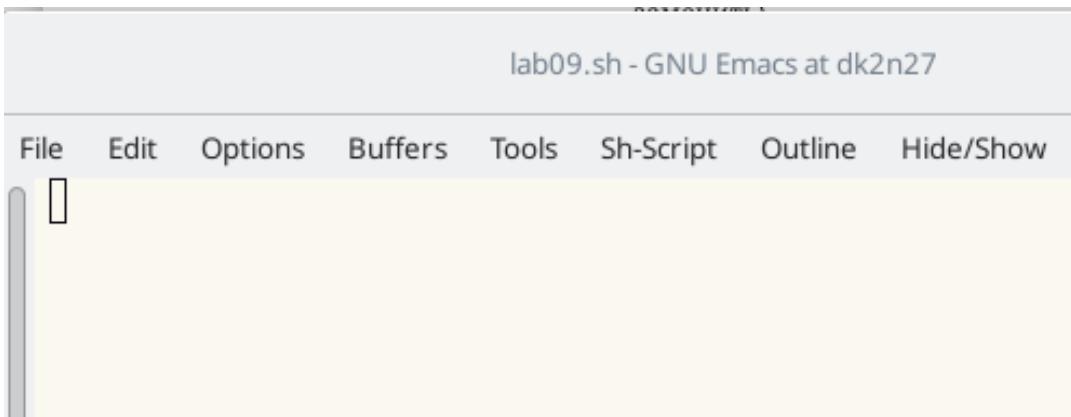


Рис. 3.2: C-x C-f

3)Набираем текст (рис. 3.3).

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

Рис. 3.3: Набор текста

5.1)Вырезаем одной командой целую строку (рис. 3.4).

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

Рис. 3.4: C-k

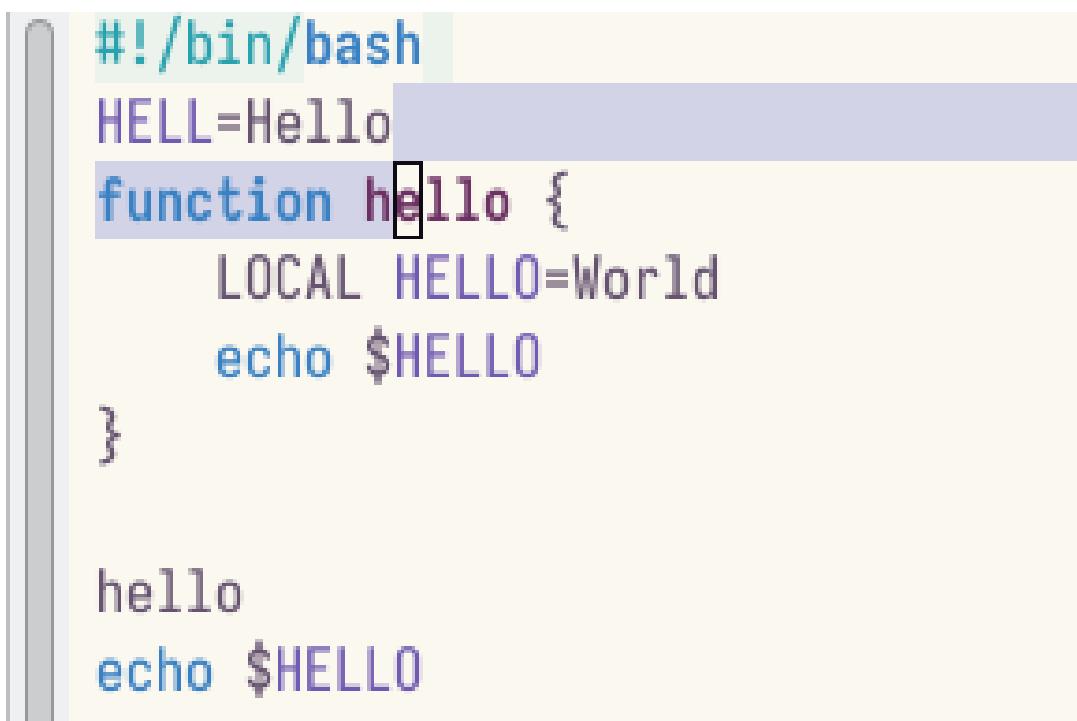
5.2) Вставляем эту строку в конец файла (рис. 3.5).

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

hello
echo $HELLO
```

Рис. 3.5: C-y

5.3)Выделяем область текста (рис. 3.6).



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

hello
echo $HELLO
```

A screenshot of a terminal window showing a Bash script. The script contains several lines of code: a shebang, variable assignment, a function definition, and two command-line executions. A rectangular selection box highlights the entire function definition, starting from 'function hello {' and ending at '}' on the same line. The background of the terminal window is light gray, and the selected text is highlighted in a darker shade.

Рис. 3.6: C-space

5.5)Скопируем область и вставим в конец файла (рис. 3.7).

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

hello
echo $HELLO

function h
```

Рис. 3.7: С-у

5.6) Вновь выделяем эту область и вырезаем её (рис. 3.8).

```
#!/bin/bash
HELL=Hellohello {
    LOCAL HELLO=World
    echo $HELLO
}

hello
echo $HELLO

function h
```

Рис. 3.8: C-w

5.7)Отменяем последнее действие (рис. 3.9).

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

hello
echo $HELLO

function h
```

Рис. 3.9: C-/

6.1)Перемещаем курсор в начало строки (рис. 3.10).

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

hello
echo $HELLO

function h
```

Рис. 3.10: C-a

6.2)Перемещаем курсор в конец строки (рис. 3.11).

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

hello
echo $HELLO

function h
```

Рис. 3.11: С-е

6.3)Перемещаем курсор в начало буфера (рис. 3.12).

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

hello
echo $HELLO

function h
```

Рис. 3.12: M-<

6.4)Перемещаем курсор в конец буфера (рис. 3.13).

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

hello
echo $HELLO

function h
```

Рис. 3.13: M-<

7.1) Выводим список активных буферов на экран (рис. 3.14).

* 110 lab09.sh Shell-script		
CRM Buffer	Size	Mode File
.* * lab09.sh	110	Shell-script[sh] ~/lab09.sh
%* *Messages*	47617	Messages
%* *GNU Emacs*	723	Fundamental
scratch	145	Lisp Interaction
%* *Compile-Log*	21198	Compilation
%* *Async-native-compile-log*	471182	Fundamental
%* *Native-compile-Log*	2748	LIMPLE
* *quelpa-build-checkout*	5085	Fundamental
%* *Warnings*	7162	Special
%* *Quail Completions*	0	Fundamental

Рис. 3.14: C-x C-b

7.2)Перемещаемся во вновь открытое окно со списком открытых буферов и переключаемся на другой буфер (рис. 3.15).

CRM Buffer	Size	Mode	File
.* * lab09.sh	110	Shell-script[sh]	~/lab09.sh
%* *Messages*	47617	Messages	
%* *GNU Emacs*	723	Fundamental	
scratch	145	Lisp Interaction	
%* *Compile-Log*	21198	Compilation	
%* *Async-native-compile-log*	471182	Fundamental	
%* *Native-compile-Log*	2748	LIMPLE	
* *quelpa-build-checkout*	5085	Fundamental	
%* *Warnings*	7162	Special	
%* *Quail Completions*	0	Fundamental	

Рис. 3.15: C-x

7.4)Переключаемся между буферами, но уже без выхода их списка на экран (рис. 3.16).

```

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

function h

```

* 110 lab09.sh Shell-script 0 chars unix | 7: 0 All
Indentation variables are now local.
Indentation setup for shell type sh
End of buffer [3 times]
Beginning of buffer [5 times]
Auto-saving...done
Saving file /afs/.dk.sci.pfu.edu.ru/home/z/z/zzmurzaev/lab09.sh...
Wrote /afs/.dk.sci.pfu.edu.ru/home/z/z/zzmurzaev/lab09.sh
(No changes need to be saved)
Scroll-Lock mode enabled in current buffer
Mark set

Рис. 3.16: C-x b

8.1) Делим фрейм на 4 части (рис. 3.17).

```

}

hello
echo $HELLO

function h[]

* 110 lab09.sh      0 chars   unix | 11:10   Bottom
}

hello
echo $HELLO

function h[]

* 110 lab09.sh      0 chars   unix | 11:10   Bottom
}

hello
echo $HELLO

function h[]

* 110 lab09.sh      0 chars   unix | 11:10   Bottom
}

hello
echo $HELLO

function h[]

* 110 lab09.sh      0 chars   unix | 11:10   Bottom

```

Рис. 3.17: C-x 3 C-x 2

8.2) Вводим несколько строк текста (рис. 3.18).

The screenshot shows a terminal window with three separate sessions of text input:

- Session 1:** The user types a script with several "I wanna" lines. The cursor is at the end of the last line: "I wanna kill you".
Content:

```
}
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wtill you
I wanna kill you
* 180 lab09.sh      unix | 12:17   Bottom
}
```
- Session 2:** The user continues to type the same script, with the cursor at the end of the last line.
Content:

```
}
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wtill you
I wanna kill you
* 180 lab09.sh      unix | 13:16   Bottom
}
```
- Session 3:** The user continues to type the script, with the cursor at the end of the last line.
Content:

```
}
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wtill you
I wanna kill you
* 180 lab09.sh      unix | 14:17   Bottom
* 180 lab09.sh      unix | 15:16   Bottom
}
```

Рис. 3.18: Набор текста

9.1) Переключаемся в режим поиска и находим несколько слов, присутствующих в тексте (рис. 3.19).

```

}

hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wtill you
I wanna kill you

* 180 lab09.sh          unix | 14:17   Bottom * 180 lab09.sh
Indentation variables are now local.
Indentation setup for shell type sh
End of buffer [3 times]
Beginning of buffer [5 times]
Auto-saving...done
Saving file /afs/.dk.sci.pfu.edu.ru/home/z/z/zzmurzaev/lab09.sh...
Wrote /afs/.dk.sci.pfu.edu.ru/home/z/z/zzmurzaev/lab09.sh
(No changes need to be saved)
Scroll-Lock mode enabled in current buffer
Mark set
Scroll-Lock mode disabled in current buffer
Mark set
Auto-saving...done
Mark set [6 times]
Mark activated[]
Mark set [2 times]
Undo
xref-pop-marker-stack: Marker stack is empty [3 times]
C-x <mouse-1> is undefined
Beginning of buffer
C-x <down> is undefined
% 48k *Messages*  Messages
Failing I-search: you

```

Рис. 3.19: C-s

9.2)Переключаемся между результатами поиска (рис. 3.20).

```
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wwill you
I wanna kill you
}
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wwill you
I wanna kill you
I wanna kill you
```

Рис. 3.20: C-s

9.3)Выходим из режима поиска (рис. 3.21).

```
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wwill you
I wanna kill you
}
hello
echo $HELLO

function h
I wanna teach you
I wanna hurt you
I wanna wwill you
I wanna kill you
I wanna kill you
```

Рис. 3.21: C-g

#Контрольные вопросы

1. Кратко охарактеризуйте редактор emacs.
2. Какие особенности данного редактора могут сделать его сложным для освоения новичком?
3. Своими словами опишите, что такое буфер и окно в терминологии emacs'а.
4. Можно ли открыть больше 10 буферов в одном окне?
5. Какие буфера создаются по умолчанию при запуске emacs?
6. Какие клавиши вы нажмёте, чтобы ввести следующую комбинацию C-c | и C-c C-|?
7. Как поделить текущее окно на две части?
8. В каком файле хранятся настройки редактора emacs?
9. Какую функцию выполняет клавиша назад и можно ли её переназначить?
10. Какой редактор вам показался удобнее в работе vi или emacs? Поясните почему.

4 Ответы на вопросы

- 1)Emacs представляет собой мощный экранный редактор текста, написанный на языке вида LISP.
- 2)Развитие Emacs в сторону его многогранности послужило причиной того, что и без того гигантский редактор стал еще более гигантским.
- 3)Буфер – что-то, состоящее из текста. Окно – область с одним из буферов.
- 4)В одном окне можно открыть больше 10 буферов.
- 5)После запуска emacs без каких-либо параметров в основном окне отображается буфер *scratch*.
- 6)Чтобы ввести следующую комбинацию С-с | я нажму клавиши: Control+c и Shift+, а с С-|: Control+c и Control+Shift+.
- 7)Поделить текущее окно на две части можно двумя комбинациями клавиш: С-x 3 или С-x 2.
- 8)Настроить или расширить Emacs можно написав или изменив файл ~/.emacs.
- 9)Клавиша ✕ выполняет функцию перемещения курсора в открытом окне также, как и мышь.
- 10)Редактор emacs показался мне удобнее из-за возможности открытия нескольких окон одновременно.

5 Выводы

Познакомились с операционной системой Linux и получили практические навыки работы с редактором Emacs.

Список литературы