



The Emacs Editor

This chapter covers the *Emacs* editor. Emacs is not distributed with all UNIX systems. However, it is available on most Linux systems. The purpose of this chapter is to introduce you to the essential commands necessary to start with Emacs. This chapter introduces the Emacs editor, simple editing jobs, and commands to start a file, edit a file, and end Emacs. The help command is covered in detail because it explains how Emacs works and describes the keys to be used.

In This Chapter

7.1 INTRODUCTION

7.2 STARTING Emacs

7.3 Emacs SCREEN

- 7.3.1 Menu Bar
- 7.3.2 Text Mode Menu
- 7.3.3 Mode Line
- 7.3.4 Echo Line

7.4 ENDING Emacs

7.5 HELP IN Emacs

- 7.5.1 Using Help: **Ctrl-h**
- 7.5.2 Using Help: **Ctrl-h t**
- 7.5.3 Using Help Command: **Ctrl-h i**

7.6 CURSOR MOVEMENT KEYS

7.7 DELETING TEXT

- 7.7.1 Killing Vs Deleting
- 7.7.2 Using Delete Operators/Keys
- 7.7.3 Using the Numeric Argument Command: **Ctrl-u**
- 7.7.4 Using the Reinserting Text Command: **Ctrl-y**

7.8 REARRANGING TEXT

7.9 CASE CONVERSION COMMANDS

7.10 FILE MANIPULATION

- 7.10.1 Finding Files
- 7.10.2 Using Finding File Command: **[Ctrl-x] [Ctrl-f]**
- 7.10.3 Canceling a Command: **Ctrl-g**

7.11 Emacs BUFFERS

- 7.11.1 Saving Buffers

7.12 FILE RECOVERY OPTION

- 7.12.1 Using the File Recovery Command: **Alt-x**

7.13 SEARCHING AND REPLACING

- 7.13.1 Using the Search Commands: **[Ctrl-s]** and **[Ctrl-r]**
- 7.13.2 Using Replace String Command: **Meta-%**

7.14 Emacs WINDOWS

7.15 THE .emacs FILE

7.16 COMMAND LINE OPTIONS

COMMAND SUMMARY

REVIEW EXERCISES

Terminal Session

7.1 INTRODUCTION

The Emacs editor is a screen-oriented text editor available for many versions of UNIX and Linux operating systems. It is a popular alternative to vi. Several versions of Emacs are available, including GNU Emacs, which is distributed by the Free Software Foundation (FSF). We are not going to cover each version here; instead, the basic commands that are common to most versions are discussed.



Emacs is a comprehensive editor that contains hundreds of commands and provides many ways to format and tailor its environment and to manipulate the file you are editing. The purpose of this chapter is to introduce you to the absolutely essential commands needed to use Emacs.

The Emacs editor can be run both in terminal mode and under the X Window System. When run under the X Window System, Emacs provides its own menus and the convenience of using a mouse. But Emacs also provides many of the window system advantages on a text-only terminal. For instance, you can look at or edit several files at once, move text between files, and edit files while running shell commands.

Here we primarily show Emacs running on a text-only terminal. In this case, the Emacs display occupies the whole screen. However, the command and procedures described (except for the use of a mouse) apply to both terminal mode and X Window mode. The partial and full screen captures that are used for examples are from a GNU/Linux system running a version of Emacs for Windows XP.

Before starting with the Emacs editor, there are two keys that you should know about: [Ctrl] and [Meta]. [Ctrl] is used with most of the commands, but some commands use [Meta]. [Meta] differs from keyboard to keyboard. For the PCs and PC-style keyboards, [Meta] is usually the [Alt] key. You can use [Esc] if [Alt] is not the [Meta] key in your system. Commands in Emacs consist of combinations of [Ctrl] or [Meta], followed by some other character.



- 1. For commands starting with [Ctrl], you must hold [Ctrl] when entering the next character.**
- 2. For commands starting with [Meta], [Meta] can be pressed and released before you enter the next character.**
- 3. [Meta] is usually the [Alt] key on the keyboard. However, if the [Alt] key doesn't work in your system, use [Esc].**
- 4. To be consistent with our keyboard convention, we use [Ctrl-x] to mean simultaneously hold down [Ctrl] and press x. For commands using the [Meta] key sequence, we use [Alt-x] to mean simultaneously hold down [Alt] key and press x.**

7.2 STARTING Emacs

To start the Emacs editor, use the following command line at the system prompt (either in terminal mode or in an xterm window):

```
$ emacs [Return] . . . . . Invoke Emacs
```

Emacs clears the screen and then displays an initial help message and copyright notice. Figure 7.1 shows the Emacs screen when it is invoked without specifying a file name. Notice that the default screen shows some basic Emacs facts and commands. Unlike vi, the Emacs editor works in only one mode, and you do not need to worry about hitting the Escape key to enter a command. Figure 7.1 shows a partial Emacs screen. Depending on the Emacs version, your screen may look different, but in all versions the first screen provides general information about Emacs version you are using.

Figure 7.1
Partial Emacs Screen

```
Welcome to GNU Emacs, one component of a Linux-based GNU system.

Get help          C-h (Hold down CTRL and press h)
Undo changes     C-x u   Exit Emacs          C-x C-c
Get a tutorial   C-h t   Use Info to read docs  C-h i
Activate menubar F10 or ESC' or M-'
('C-' means use the CTRL key. 'M-' means use the Meta (or Alt) key.
If you have no Meta key, you may instead type ESC followed by the character.)

GNU Emacs comes with ABSOLUTELY NO WARRANTY; type C-h C-w for full details.
Emacs is Free Software—Free as in Freedom—so you can redistribute copies
of Emacs and modify it; type C-h C-c to see the conditions.
Type C-h C-d for information on getting the latest version.
```



1. In Emacs documentation, *C* stands for [Ctrl]. For example, **C-x** means press and hold [Ctrl] and press **x**. The [Ctrl] key is labeled CTRL or CTL on your keyboard.
2. In Emacs documentation, *M* stands for [Meta], which is the [Alt] key in most systems. For example, **M-x** means press and hold [Alt] and press **x**.
3. Notice the command to exit Emacs is C-x C-c. That means you type **C-x** followed by **C-c**. Using this book's notation, the exit command is presented as [Ctrl-x] followed by [Ctrl-c]

You can start Emacs with a file name specified on the command line. The format is

```
emacs filename
```

where *filename* is the name of a file you want to create or edit. For example, suppose you type the following command:

```
$ emacs Example [Return] . . . . . Start Emacs with the Example file
```

If `Example` is a new file, Emacs clears the screen and the cursor is positioned at the upper-left corner of the screen. If an `Example` file already exists, Emacs displays the first page (one screenful) of the file. Figure 7.2 is the Emacs screen showing the `Example` file. You can create the `Example` file by invoking Emacs and typing the text shown in Figure 7.2.

Figure 7.2

Emacs Screen Showing the `Example` File

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

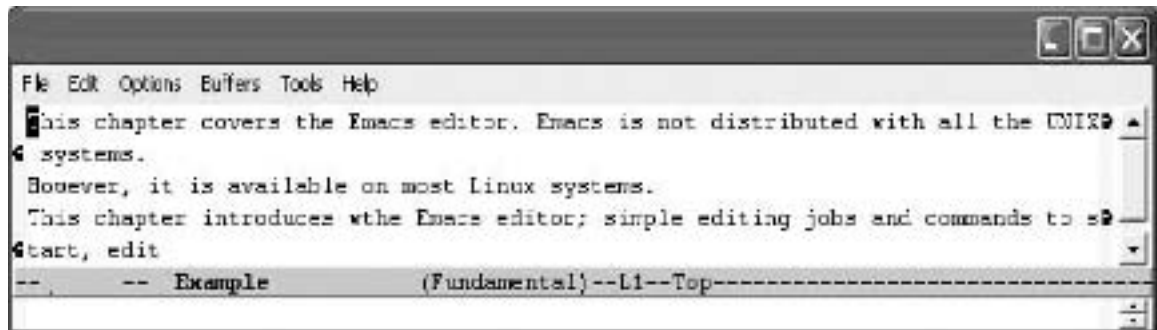
```
--:-- Example                                <Fundamental>--L1---ALL-----
```



When a line of text gets longer than one line on the screen, the line of text is continued onto a second line. A backslash (\) or, if you are using a windowed display, a little curved arrow at the margins indicates a line which has been continued. See Figure 7.3.

Figure 7.3

Screen Capture Showing Curved Arrows at the Margins



7.3 Emacs SCREEN

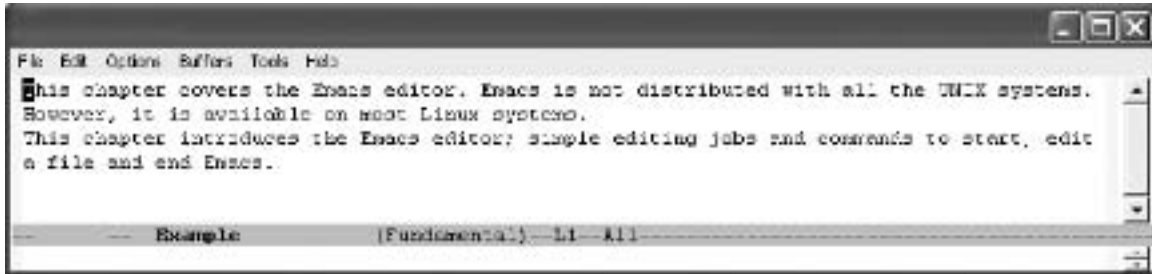
When you start Emacs, the entire screen except for the top and bottom lines is available for you to edit or enter text. This area is called the *window*. At the top there is normally a menu bar, where you can access a series of menus and submenus that provide different editing functions using a mouse.

7.3.1 Menu Bar

Each Emacs window normally has a menu bar at the top of the window. When you are using a window system, you can use the mouse to choose a command from the menu bar. An arrow pointing right, after the menu item, indicates that item has a submenu.

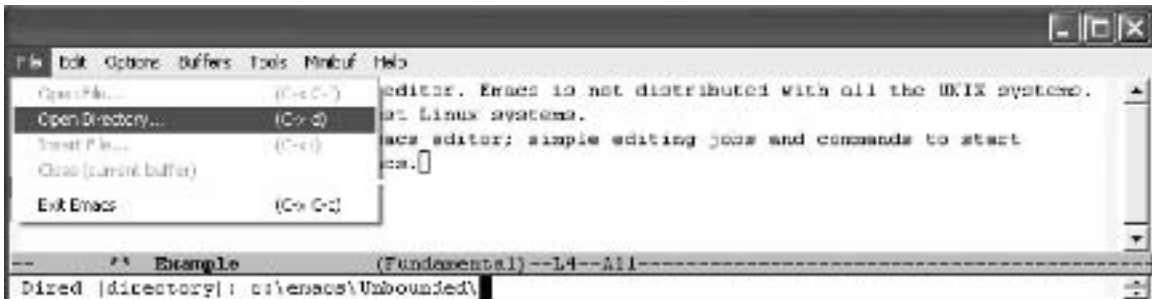
The . . . at the end means that the command requires arguments and will read arguments from the keyboard before it actually does anything. Figure 7.4 shows an Emacs screen capture in a window system.

Figure 7.4
The Emacs Screen Showing the Menu Bar



If your UNIX system supports a mouse (either in the terminal mode or X window), then you can use the mouse to scroll through a document and make selection from a pull-down menu. However, the menus do not exactly behave as you might expect from working with Microsoft Windows. You expect a command to be executed when you select it from the menu. This does not happen with Emacs. The selection is loaded at the bottom of the screen. You can run the command directly, or else you can enter additional information, for example, specifying the filename. Figure 7.5 shows a screen capture for selecting a menu item in a window system.

Figure 7.5
The Emacs Screen Showing Submenu Selection



7.3.2 Text Mode Menu

On text-only terminals with no mouse, you can use the menu bar by pressing the **[F10]** key. This command lets you to select a menu item from the keyboard. A prompt appears on the echo line at the bottom of the screen. Use the left and right arrow keys to move through the menu to different choices. When you have found your choice, press **[Return]** to select your choice. Figure 7.6 shows the *echo line*, also called *echo area* and *minibuffer*, after **[F10]** is pressed.



Please read the information on the screen. Essential keys to be used to move through the menu are mentioned and explained, including important keys such as those that cancel the menu screen.

Figure 7.6

Screen Showing First-Level Menu Using the Keyboard

Press PageUp Key to reach this buffer from the minibuffer. Alternatively, you can use Up/Down keys (or your History keys) to change the item in the minibuffer, and press RET when you are done, or press the marked letters to pick up your choice. Type C-g or ESC ESC ESC to cancel. In this buffer, type RET to select the completion near point.

Possible completions are:

b==>Buffers	f==>Files
t==>Tools	e==>Edit
s==>Search	m==>Mule
h==>Help	

```
--11:**-F1 *Completions*      (Completion List)--L1--A11---
```

```
Menu bar (up/down to change, PgUp to menu): b==>Buffers
```

Use the up and down arrow keys to move through the menu to different choices. When you have found your choice, press **[Return]** to select it.

The current menu choice in Figure 7.6 is `b==>Buffers`, indicating your choice is the first item on the menu. Each time you press the **up** or **down arrow key**, the menu item changes from one menu item to another.

```
Menu bar (up/down to change, PgUp to menu): b==>Buffers
```

- Press the **down arrow key** one time.

This changes the menu choice on the prompt line to `f==>File`.

```
Menu bar (up/down to change, PgUp to menu): f==>File
```

- Press **[Return]**

This indicates your choice is menu item File.

Figure 7.7 is the screen showing your choices for the File menu. The prompt at the bottom of screen shows the current menu choice, `o==>Open File... (C-x C-f)`.

Notice the format of the menu items on the bottom of the screen, for example:

```
o==>Open File... (C-x C-f)
```

Each menu item is identified by a letter or digit. This letter or digit is separated from the item name by `= =>`. You can enter the item's letter or digit to select the item.

`Open File` is the name of the command, and the letter `o` is the abbreviation for the Open File command. You can type the command abbreviation directly at the prompt instead of searching for a menu item.

`. . .` (`dot dot dot`) means this menu choice requires an argument, in this case the name of the file to be opened.

Figure 7.7

Screen Showing the File Menu Using the Keyboard

Press PageUp Key to reach this buffer from the minibuffer. Alternatively, you can use Up/Down keys (or your History keys) to change the item in the minibuffer, and press RET when you are done, or press the marked letters to pick up your choice. Type C-g or ESC ESC ESC to cancel. In this buffer, type RET to select the completion near point.

Possible completions are:

o==>Open File... (C-x C-f)	O==>Open Directory... (C-x d)
s==>Save Buffer As... (C-x C-w)	r==>Recover Session...
i==>Insert File... (C-x i)	k==>Kill Current Buffer
m==>Make New Frame (C-x 5 2)	n==>Open New Display...
S==>Split Window (C-x 2)	w==>One Window (C-x 1)
e==>Exit Emacs (C-x C-c)	

```
--11:**-F1 *Completions*
```

```
(Completion List)--L1--A11-----
```

```
Files (up/down to change, PgUp to menu): o==>Open File... (C-x C-f)
```

(C-x C-f) indicates this command is also mapped to a sequence of keys. You can use the key sequence to open a file instead of using the menu options.

You can use **[Page Up]** to move the cursor to the menu screen.

Press **[Page Up]**

This moves the cursor to letter **o** of the menu's first item:

```
o==>Open File... (C-x C-f)
```

You can use the arrow keys to move the cursor from one menu item to another and use **[Page Up]** and **[Page Down]** to move from one page to another. You press **[Return]** to select your choice. If the item you have selected requires an argument, the cursor is placed on the *echo line* and you are prompted to enter the necessary argument(s).

Use the arrow keys to place the cursor on the first menu item:

```
o==>Open File... (C-x C-f)
```

Press **[Return]**

This indicates your selection from the menu. You will be prompted to enter the filename.

```
Find file: /student/home/david/█
```

Your Emacs prompt will be similar to this, but your pathname is going to be different. Enter a filename such as *Example* or any other text file in your directory, followed by **[Return]**. The specified file will be opened in Emacs.



1. You can invoke the menu screen anytime in your Emacs session. The menu screen appears in its own buffer on the screen and not in the file buffer.
2. You can press the Escape key three times, **[Esc][Esc][Esc]**, to cancel the menu.

7.3.3 Mode Line

The *mode line* appears right above the *echo line* and is the next-to-last line on the screen. When Emacs starts, the mode line displays status information, such as what buffer is being displayed above it in the window, what major and minor modes are in use, and whether the buffer contains unsaved changes. The mode line starts and ends with dashes. On a text-mode display, the mode line is in inverse video if the terminal supports that. Normally, the mode line looks like the following line and consists of several fields:

```
--:-- buf (major minor) -- line ----- pos -----
```

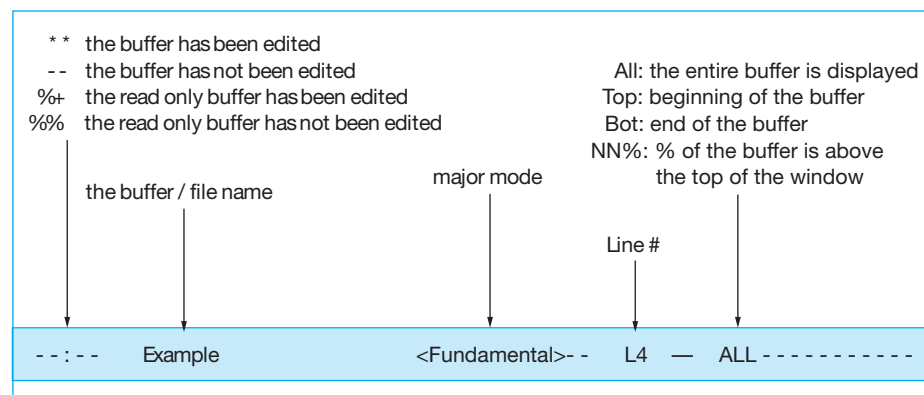
The two characters after the `:` (colon) indicates the status of the text in the buffer.

- The dashes (- -) after the `:` indicates that the buffer has not been edited.
- ** The stars (**) after the `:` indicates the text in the buffer has been modified (edited).
- %% The percentages (%%) after the `:` indicates that a read-only buffer is not modified.
- %* The percentage star (%*) after the `:` indicates that a read-only buffer is modified.
- buf** This is the name of the window's buffer. In most cases this is the same as the name of the file you are editing.
- line** This is the letter **L** followed by the current line number of the cursor position. This is present when Line Number mode is enabled (which is normally the case).
- pos** This indicates whether there is additional text above the top of the window, or below the bottom. If your file is small it is all visible in the window. **pos** could indicate one of the following:
 - ALL** This indicates the entire buffer (file) is displayed.
 - Top** This indicates the beginning of the buffer (file) is displayed.
 - Bot** This indicates the end of the buffer (file) is displayed.
 - NN%** This indicates the percentage of the buffer (file) above the top of the window.

Major This indicates the name of the *major mode* in effect in the buffer. The *major modes* available include *Fundamental mode* (the least specialized), *Text mode*, *Lisp mode*, *C mode*, and many others.

Figure 7.8 shows an example of mode line's fields.

Figure 7.8
Emacs Mode Line Fields





Make sure the mode line shows *<Fundamental>* in the Major/Minor mode field. Other modes are for programming languages such as Lisp interaction mode. In any case, pressing [Ctrl-x] [Ctrl-f] one after another will create a text file buffer and changes the Major mode to *<Fundamental>*.

7.3.4 Echo Line

The last line on the screen (below the mode line) is a special *echo area* where prompts appear and where you can enter information such as a filename. Echoing means what you type will be displayed on this line. Commands are echoed only if you pause while typing them. Also, if a command cannot be executed, it may display an error message on the echo line. The commands that take a long time to execute often display messages ending in . . . (dot dot dot) while they are working, and show the word *done* at the end when they are finished.

minibuffer The *echo line* is also used for reading arguments to the commands, such as the name of a file to be opened. The bottom line is called a *minibuffer* when it is used for this sort of input. In this case, the *echo line* begins with a prompt string followed by the cursor. For example:

```
Find file: C:\emacs\Unbounded/█
```



You can always get out of the echo area by typing [Ctrl-g].

7.4 ENDING Emacs

In order to save the file you have created, or edited in Emacs, you type **Ctrl-x Ctrl-s**. The Emacs editor asks you for the name of the file. Observe how the command is displayed in the echo line as you type it. Table 7.1 lists the sequence of keys to save files and quit Emacs.

Table 7.1
Emacs Save and Quit Commands

Key	Operation
Ctrl-x Ctrl-s	Saves the file (contents of the current buffer) and exits Emacs.
Ctrl-x Ctrl-c	Quits the Emacs editor and abandons the contents of the file.
Ctrl-x Ctrl-w	Saves the file (contents of the current buffer) into file <i>filename</i> .



Scenario: Assuming that you have launched Emacs by specifying the filename Example on the command line, your screen would be similar to the following partial screen capture.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

```
--:**Example                               <Fundamental>--L4--ALL-----
```

The two ****** on the *mode line* indicates that this file was changed.

Press **[Ctrl-x] [Ctrl-s]**

This displays a confirmation message on the *echo line* confirming the file was saved.

```
--:**Example                               <Fundamental>--L4--ALL-----
```

```
Wrote file /emacs/Unbounded/Example
```



Another scenario: Assuming you have modified the file `Example`, as indicated by two ****** on the *mode line*. Now you want to exit Emacs.

Press **[Ctrl-x] [Ctrl-c]**

This prompts you on the *echo line* asking for confirmation to save the file.

```
--:**Example                               <Fundamental>--L4--ALL-----
```

```
Save file /emacs/Unbounded/Example? (y, n, !, ., q, C-r or C-h)_
```

If you want to save the file and exit Emacs:

Press **[y]**

This saves the file `Example` and `$` prompt indicates you are out of Emacs.

If you want to exit Emacs and not saving the file:

Press **[n]**

This indicates you do not want to save the file. The prompt appears on the *echo line* asking you to confirm your intention.

```
--:**Example                               <Fundamental>--L4--ALL-----
```

```
Modified buffer exist; exit anyway? (yes or no) _
```

You have to answer **yes** or **no** followed by **[Return]**. Entering any other word is not acceptable. For example, if you enter **n** instead of **no**:

```
--:**Example      <Fundamental>--L4--ALL-----
Modified buffer exist; exit anyway? (yes or no) n [Return]
```

Emacs will complain. Observe the message on the *echo line*.

```
--:**Example      <Fundamental>--L4--ALL-----
Please enter yes or no.
```

The message “Please enter yes or no” remains on the screen for few seconds and then is replaced by the previous prompt, asking you to enter yes or no. Entering **no** will put the cursor back in the file window and you can go on editing your file. Entering **yes** will quit Emacs without saving the file `Example`. There are other options on the prompt line when you want to save a file and then exit Emacs.

```
Save file /emacs/Unbounded/Example? (y, n, !, ., q, C-r or C-h) █
```

The easiest way to find out what the other options mean is to enter **C-h** for help. That is, press [**Ctrl-h**]. Figure 7.9 shows the help view after entering [**Ctrl-h**].

Figure 7.9

The Emacs Help View

```
Type SPC or 'y' to save the current buffer;
DEL or 'n' to skip the current buffer;
RET or 'q' to exit (skip all remaining buffers);
! to save all remaining buffers;
ESC or 'q' to exit;
^R to display the current buffer;
or . (period) to save the current buffer and exit.
```

```
--11:%%-F1 *Help*                                (Help View)--L1--A11-----
```

```
Save file /emacs/Unbounded/Example? (y, n, !, ., q, C-r or C-h)
```



1. Notice that there are many ways to exit Emacs and to save or not save the file.
2. In this help explanation, the symbol **^** (caret) means you use the [**Ctrl**] key.



Another scenario: Assuming that you have launched Emacs by specifying the filename `Example` on the command line, your screen would be similar to the following partial screen capture.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.■

```
--:**Example                <Fundamental>--L4--ALL-----
```

The two ** on the *mode line* indicates that this file was changed.

□ Press [Ctrl-x] [Ctrl-w]

This prompts you to enter the filename.

```
--:**Example                <Fundamental>--L4--ALL-----
```

```
Write file: /emacs/Unbounded/■
```

□ Type **junk** [Return]

This displays a confirmation message on the echo line confirming the file was saved with the name junk.

```
--:**Junk                  <Fundamental>--L4--ALL-----
```

```
Wrote file /emacs/Unbounded/junk
```



Notice the filename was changed from Example to junk, so your editing will be done in current buffer named junk. The Example file will remain intact.

7.5 HELP IN Emacs

The Emacs editor includes a help system that can be invoked if you need something explained or do not remember the many Emacs commands. Depending on the version of Emacs, one of the following keys or key sequences can be used to invoke the help system:

```
[Ctrl-h]   [F1]           [Esc-?]
[Meta-?]   [Meta-x]
```

However, [Ctrl-h] is the most common command entered for help. Unlike all other UNIX application, Emacs does not rely on online manual pages (the **man** command) for describing its functionality. The Free Software Foundation provides an **info** format for the Emacs online help. Table 7.2 lists some of the keys that can be used to get help for different topics in Emacs.

Table 7.2
Emacs Help Commands

Key	Operation
Ctrl-h	Invokes the emacs help (Control key is held while entering h).
Ctrl-h t	Invokes a short emacs tutorial. (Control key is held while entering h followed by t).
Ctrl-h k	Explains the function of a particular key.
Ctrl-h i	Loads the info documentation reader.
Ctrl-h Ctrl-c	Displays the Emacs General Public License.
Ctrl-h Ctrl-d	Displays information on ordering Emacs from FSF.

7.5.1 Using Help: Ctrl-h



This command invokes the Emacs help. First open a file in Emacs. For example:

```
$ emacs Example [Return]..... Open Example in Emacs
```

The GNU Project was launched in 1984 to develop a complete UNIX style operating system. Variants of the GNU (pronounced guh-noo) operating system, called Linux, are now widely used.

Free Software Foundation (FSF) is the principal organizational sponsor of the GNU Project.

Emacs is a popular screen-oriented text editor that is distributed by FSF.

```
--1-:F1 Example <Fundamental>--L1--A11-----
```

- Press **[Ctrl-h]** (Hold the Control key while entering **h**.)

This displays the following prompt line:

```
-- Emacs is a popular screen-oriented text editor that is distributed by FSF.
```

```
--1-:F1 Example <Fundamental>--L1--A11-----
```

```
C-h (Type ? for further options)█
```

- Press **[?]**

This displays the help options in the help window (view). It also prompts you on the echo line to enter your selection.

```
--
Emacs is a popular screen-oriented text editor that is distributed by FSF.
```

```
--1-:F1 Example <Fundamental>--L1--A11-----
```

```
You have typed C-h, the help character. Type a Help option:
(Use SPC or DEL to scroll through this text. Type q to exit the Help command.)
```

```
a command-apropos. Give a substring, and see a list of commands
  (functions interactively callable) that contain
  that substring. See also the apropos command.
b describe-bindings. Display table of all key bindings.
c describe-key-briefly. Type a command key sequence;
  it prints the function name that sequence runs.
C describe-coding-system. This describes either a specific coding system
  (if you type its name) or the coding systems currently in use
```

```
--11:%*-F1 *Help* (Help View)--L1--Top-----
```

```
Type one of the options listed, or SPACE or DEL to scroll:█
```

- Press the [Space] or [Del] key at the prompt to scroll up or down in the help window.
- Press the [q] key to exit the help window.

7.5.2 Using Help: Ctrl-h t



This command invokes the Emacs short tutorial help. First open a file in Emacs. For example:

```
$ emacs Example [Return] . . . . Invoke Emacs using the filename
Example
```

- Press [Ctrl-h] t

This displays the Emacs tutorial screen.

Using the following keys, you can move forward to the next screen or backward to the previous screen.

- Press [Ctrl-v] (Hold down [Ctrl] and type v)

This takes you to the next tutorial screen.

- Press [Alt-v] (Hold down [Meta], in this case [Alt], and type v)

This takes you to the previous screen.



[Meta] can be the [Esc] or [Alt] key on your keyboard.

Copyright (c) 1985 Free Software Foundation, Inc; See end for conditions.
You are looking at the Emacs tutorial.

Emacs commands generally involve the CONTROL key (sometimes labeled CTRL or CTL) or the META key (sometimes labeled EDIT or ALT). Rather than write that in full each time, we'll use the following abbreviations:

C-<chr> means hold the CONTROL key while typing the character <chr>
Thus, C-f would be: hold the CONTROL key and type f.
M-<chr> means hold the META or EDIT or ALT key down while typing <chr>.
If there is no META, EDIT or ALT key, instead press and release the ESC key and then type <chr>. We write <ESC> for the ESC key.

Important note: to end the Emacs session, type C-x C-c. (Two characters.)
The characters ">>" at the left margin indicate directions for you to try using a command. For instance:

>> Now type C-v (View next screen) to move to the next screen.
(go ahead, do it by holding down the control key while typing v).
From now on, you should do this again whenever you finish reading the screen.

```
--1-:F1 TUTORIAL      <Fundamental>--L1--Top-----
```

Use [Alt-v] and [Ctrl-v] to browse and read the Emacs tutorial. This tutorial covers the Emacs fundamental commands and practices.

You can ask for a description of a specific key or command. For example:

Press [Ctrl-x] h (Hold down [Ctrl] and type x followed by h.)

This displays the following prompt message on the echo line.

```
--1-:F1 TUTORIAL      <Fundamental>--L1--Top-----
```

Describe key:

Press [Ctrl-v]

This shows the description for the [Ctrl-v] (scroll up) command Figure 7.10 Shows the partial screen after you have entered the command.

The echo line shows the commands to remove the help screen and how to scroll the help screen.

Press [Ctrl-x] 1 (Hold down [Ctrl] and type x; then enter 1)

This removes the help screen.

Figure 7.10

Partial Help Screen Describing the [Ctrl-v] Command

```

--1-:**-F1 TUTORIAL                                (Fundamental)--L22--2%-----
C-v runs the command scroll-up
      which is an interactive built-in function.

Scroll text of current window upward ARG lines; or near full screen if no ARG.
A near full screen is 'next-screen-context-lines' less than a full screen.
Negative ARG means scroll downward.
If ARG is the atom '-', scroll downward by nearly full screen.
When calling from a program, supply as argument a number, nil, or '-'.

(scroll-up &optional ARG)

--11:%%-F1 *Help*                                  (Help View)--L1--A11-----
Type C-x 1 to remove help window. M-C-v to scroll the help.

```

7.5.3 Using Help: Ctrl-h i



The *info* mode is a hypertext-based system that covers many commands. As hypertext it allows you to move down layer by layer from general commands and concepts to more specific information. Figure 7.11 shows the *info* mode of the Emacs.

Figure 7.11

Partial Screen Showing the Emacs info Mode

```

File: dir  Node: Top  This is the top of the INFO tree

This (the Directory node) gives a menu of major topics.
Typing "q" exits, "?" lists all Info commands, "d" returns here,
"h" gives a primer for first-timers,
"mEmacs<Return>" visits the Emacs topic, etc.

In Emacs, you can click mouse button 2 on a menu item or cross reference
to select it.

* Menu:

Texinfo documentation system
* Standalone info program: (info-stand). Standalone Info-reading program.
* Texinfo: (texinfo). The GNU documentation format.
* install-info: (texinfo)Invoking install-info. Update info/dir entries.
* makeinfo: (texinfo)makeinfo Preferred. Translate Texinfo source.
* texi2dvi: (texinfo)Format with texi2dvi. Print Texinfo documents.

```

```
* texindex: (texinfo)Format with tex/texindex. Sort Texinfo index files.
```

```
Miscellaneous
```

```
* Am-utils: (am-utils). The Amd automounter suite of utilities
```

```
--11:%-F1 Info: (dir)Top
```

```
(Info Narrow)--L1--Top-----
```

```
Composing main Info directory...done
```

The *info* section of the **Help** command brings up a general listing of software packages and commands. You scroll down the list and find the entry for Emacs to get the Emacs specific information. Once you start getting to the deeper levels, the information becomes more specific, and at some point you will see a listing of commands and short descriptions of what they do. You can select any of the commands to get into more detailed explanation of the command and its functionality.

7.6 CURSOR MOVEMENT KEYS

Similar to the vi editor, Emacs features many commands designed to navigate around in the text on the screen. The basic cursor movement commands are listed in Table 7.3. Most implementations of Emacs conveniently map the first four movement commands to the arrow keys on the keyboard.

Table 7.3
Emacs Cursor Movement Keys

Key	Operation
Ctrl-f or [→]	Moves the cursor forward one character.
Ctrl-b or [←]	Moves the cursor back one character.
Ctrl-p or [↑]	Moves the cursor to the previous line.
Ctrl-n or [↓]	Moves the cursor to the next line.
Ctrl-a	Moves the cursor to the beginning of the line.
Ctrl-e	Moves the cursor to the end of the line.
Ctrl-v	Moves the cursor forward one screen.
Meta-v	Moves the cursor backward one screen.
Meta-f	Moves the cursor forward one word.
Meta-b	Moves the cursor back one word.
Meta-<	Moves the cursor to the beginning of the text.
Meta->	Moves the cursor to the end of the text.

7.7 DELETING TEXT

You can delete text in several ways. **[Backspace]** or **[Del]** is used to erase the character before the cursor. The command **[Ctrl-d]** deletes the character under the cursor and **[Ctrl-k]** deletes (kills) all characters from the cursor current position to the end of the line. Table 7.4 lists the keys used for deleting text in Emacs.

Table 7.4
Keys for Deleting Text

Key	Operation
Backspace or Delete	Deletes the character before the cursor.
Ctrl-d	Deletes the character under the cursor.
Ctrl-k	Kills all characters from cursor to the end of the line.
Meta-d	Kills the next word after the cursor.
Meta-Del (Delete key)	Kills the word before the cursor.
Meta-k	Kills the sentence that cursor is on.
Ctrl-x Del	Kills the previous sentence.
Ctrl-w	Kills all the text between the two positions.



1. Note there are “delete” and “kill” operations.
2. **[Del]** or **[Backspace]** may not work in your system as you expect, that is, deleting a character before the cursor. Instead, it may invoke a help prompt. This problem is caused by the terminals’ incompatibility.

7.7.1 Killing Vs Deleting

There is a difference between killing and deleting text. Except for character deletion, the text you kill will be saved in a section of memory reserved for storage of deleted text. This storage is called the *kill buffer*, or *kill ring*. The killed text that was saved in the *kill buffer* can be reinserted (yanked), whereas deleted characters cannot be reinserted. In general, when you use the commands to remove a lot of text, the *kill buffer* is available so that you can yank the text. When you use commands that remove just one character or remove only blank lines and spaces, the deleted text is not saved, and you cannot yank that text. You don’t need to set up a *kill buffer*. It already exists, and its default size is 30 deleted items. The size of the *kill buffer* can be changed in the Emacs configuration file. Table 7.5 lists the recovery keys.

Table 7.5
Recovery Keys

Key	Operation
Ctrl-y	Restores (yanks) what was killed.
Meta-y	After Ctrl-y , inserts the previously killed section of text.
Ctrl-x u	Undoes the previous editing changes.

7.7.2 Using Delete Operators/Keys

For practicing delete commands, first create a file named `Example` or open the one you have already created. The following command opens the file `Example` in Emacs.

```
$ emacs Example [Return]....Invoke Emacs using the filename Example
Use the arrow keys or other cursor movement keys to place the cursor on letter E of
Emacs on line 1.
```

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

```
--:--Example <Fundamental>--L1--ALL-----
```

□ Press **[Ctrl-d]**

This deletes the letter **E**, and the cursor moves to letter **m**, the next letter.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

□ Press **[Ctrl-x] u**

This undoes the last changes, in this case the deletion of the letter **E**.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

```
--:--Example <Fundamental>--L1--ALL-----
```

Undo!

Notice the commands you type are echoed on the echo line at the bottom of the screen. This is true for most of the commands; in this case, as soon as the command is executed, it shows “Undo!”

7.7.3 Using the Numeric Argument Command: Ctrl-u

You can use numeric arguments to further control the commands’ operations. For example, using the numeric argument, you can specify the number of characters to be deleted. The

command for specifying the argument is **[Ctrl-u]**. This begins a numeric argument for the command that follows it. For example, the sequence of keystrokes for deleting the word Emacs using the command argument is:

❑ Press **[Ctrl-u] 5 [Ctrl-d]**

This deletes five characters, starting from the cursor position. Thus, **[Ctrl-u] 5** provides the numeric argument for the command **[Ctrl-d]**.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

❑ Press **[Ctrl-x] u**

This undoes the last changes—in this case, the deletion of the word Emacs—and you are back to where you started.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

❑ Press **[Ctrl-k]**

This kills all characters from the cursor position (cursor is on E) to the end of the line. Next, move the cursor to the beginning of the line 2 on the letter H of the word However.

❑ Press **[Ctrl-k]**

This kills the whole line. Observe that the line is deleted but Newline remains.

This chapter covers the Emacs editor.

This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

❑ Press **[Ctrl-k]** a second time.

This kills the Newline that follows the line you have just killed. The first **[Ctrl-k]** killed the contents of the line, and a second **[Ctrl-k]** killed the line itself. Observe that the rest of the lines are all moved up.

This chapter covers the Emacs editor. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.

Place the cursor on the beginning of the last line on the letter a.

❑ Press **[Ctrl-k]**.

This kills the last line. Observe that the line is deleted.

This chapter covers the Emacs editor.
This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit.

- Press **[Ctrl-x] u** four times.

This undoes the changes. In this case, the last four kills are recovered and the text looks the same as before the deletion operations.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start, edit a file and end Emacs.█



1. You can undo deletion of text just as you can undo killing of text. The distinction between killing and deleting affects the **[Ctrl-y]** to yank text but makes no difference when using **[Ctrl-x] u**, the undo command.
2. If you use **[Ctrl-k]** several times in a row, all the killed text is saved together, so that one **[Ctrl-y]** will yank all the lines at once.

However, the killed texts were copied into the *kill buffer*. That means you can bring back the killed texts from the *kill buffer* (this is called yanking) and paste it to any place in your file. After the three kill operations of the previous exercise, the kill buffer contains three items. Figure 7.12 represents the *kill buffer* and the order of the saved text.

Figure 7.12

The Order of the Deleted Items in the *Kill Buffer*

↑	1	a file and end Emacs. <i>(the most recent kill)</i>
	2	However, it is available on most Linux systems. <i>(the second kill)</i>
	3	Emacs is not distributed with all the UNIX systems. <i>(the first kill)</i>

7.7.4

Using the Reinserting Text Command: **Ctrl-y**

The command for reinserting (yanking) text is **[Ctrl-y]**. It reinserts the most recent kill text at the current cursor position. For example, to reinsert the last kill text from the kill buffer:

- Press **[Alt->]**

This places the cursor at the end of the file.

- Press **[Ctrl-y]**

This yanks the most recent kill text. In this case, item number 1 in our kill buffer.



1. Every time you use **[Ctrl-y]**, the yank command, you get the same text, the most recent text from saved in the *Kill Buffer*.

2. You can yank the same text several times to make multiple copies of it.
3. You use **[Meta-y]** access the rest of the items in the kill buffer.
4. We use **[Alt]** for **[Meta]**. Your system might have other keys mapped to the **[Meta]** key.



Scenario: What if you want a copy of some other items from the *kill buffer*? For example, suppose you want to yank item 2 and place it at the last line of the text. The **[Ctrl-y]** commands insert the most recent killed text at the cursor position. The following sequence of commands will do the job:

- Press **[Alt->]**

This places the cursor at the end of the text.

- Press **[Ctrl-y]**

This places item 3, the most recently killed text, at the cursor position.

- Press **[Alt-y]**

This replaces the previously yanked text and places item 2 (the second killed text) at the cursor position.

- Press **[Alt-y]** again

This replaces the previously yanked text and places item 3 at the cursor position. You can use numeric argument with **[Ctrl-k]** to kill many lines with one command. For example:

- Press **[Alt-<]**

This places the cursor at the beginning of the text. In this case, the letter **T** is placed on line 1.

- Press **[Ctrl-u] 3 [Ctrl-k]**

This kills three lines, starting from the cursor position.



1. Observe the echo line. It shows the commands you are typing.
2. The three deleted lines are copied into the kill buffer as one item.

Next, let us undo the last delete operation.

- Press **[Ctrl-x] u**

This undoes the last changes, in this case, the deletion of three lines. Observe the three lines are recovered with one command.

- Press **[Alt->]**

This places the cursor at the end of the buffer.

- Press **[Ctrl-y]**

This yanks the three lines and places them at the current cursor position.

7.8 REARRANGING TEXT

Deleting, copying, moving, and changing text are collectively referred to as cut-and-paste operations. Emacs provides easier commands than the vi editor to rearrange the text in your file, but you have to know about *point* and *mark* before doing the cut-and-paste operations.

Point is the location of the cursor in the text.

Mark is a remembered location in the text.

You have to indicate the boundaries of the section of the text that you intend to rearrange in the file. The mark is used to indicate the beginning, and the cursor location indicates the end of the selected section of the text. Newer versions of Emacs allow you to select a section of the text by moving the cursor to the beginning of the section and dragging the mouse to the end of the text. The selected section is highlighted. Table 7.6 lists the mark and point keys.

Table 7.6
Set Mark Keys

Key	Operation
Ctrl-SPC	Sets the mark at the current point location (Control key followed by Space key).
Ctrl-x Ctrl-x	Interchanges point and mark locations. This allows showing the mark location.



Scenario: We want to move a selection from one part of the file to another. Open the *Example* file in Emacs by typing:

```
$ emacs Example [Return] .....Invoke Emacs using the filename
Example
```

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.■

```
--:--Example <Fundamental>--L4--ALL-----
```

Move the cursor to the letter **T** at the beginning of line four.

□ Press [CtIs-SPC]

This displays the message *Mark set* at the bottom of the window on the echo line.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.■

```
--:--Example <Fundamental>--L3--ALL-----
```

Mark set

Move the cursor to the end of the file to indicate the *point*.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.■

--:--Example

<Fundamental>--L4--ALL-----

Notice the *mark* (location of the letter **T**) is invisible unless you are using Emacs with X11 windows. Just to make sure the *mark* is set, do the following:

- Press [CtIs-x] [CtIs-x].

This interchanges *point* and *mark* and shows where the location of *mark* is.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.■

--:--Example

<Fundamental>--L3--ALL-----

C-x C-x

Notice that the cursor is back on the letter **T**, indicating your *mark* location.

- Press [Ctrl-x] [Ctrl-x].

This returns your selected section of the text to its previous state. The end result is that you have selected (blocked) a section of the text and indicated the boundaries by setting *mark* on the letter **T** and *point* at the end of the file. Now, you can use commands to delete, copy, or rearranged the selected text.



Scenario: Copying the selected section of the text to another location in the file:

- Press [Ctrl-w]

This kills the selected text (between the *mark* and *point* boundaries). The deleted text is in *kill buffer* and can be reinserted in other parts of the file.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems.■

--:**Example

<Fundamental>--L3--ALL-----

- Press [Alt-<]

This places the cursor at the beginning of the file on the letter **T**.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems.

```
--:**Example          <Fundamental>--L1--ALL-----
```

- Press [Ctrl-y]

This inserts (yanks) the deleted text on cursor position.

This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs. This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems.

```
--:--Example          <Fundamental>--L2--ALL-----
```

Mark set

The \ (backslash) at the end of line 2 on the right margin indicates continuation of line 2 onto the next line.

7.9 CASE CONVERSION COMMANDS

Emacs provides commands to capitalize a word, change a word or a selected part of text from uppercase to lowercase or visa versa. Table 7.7 lists the keys that are used for case conversions.

Table 7.7
Emacs Case Conversion Keys

Key	Operation
Meta-u	Converts the following word to uppercase.
Meta-l	Converts the following word to lowercase.
Meta-c	Capitalizes the following word.
Ctrl-x Ctrl-u	Converts the region to uppercase.
Ctrl-x Ctrl-l	Converts the region to lowercase.

Notice, the **[Ctrl-x][Ctrl-u]** command operates on a specified region. The region is the text between a *point* and a *mark*.



Scenario: Let's assume you have the `Example` file opened in Emacs and you intend to change a region, in this case the last line in the file, to uppercase. To do so, you must specify the region. The following command sequences show how to identify the region and change it to uppercase.

❑ Press **[Alt->]**

This places the cursor at the end of the file.

❑ Press **[Ctrl-SPC]**

This sets the mark at the end of the last line. The message *mark set* is displayed on the echo line. Remember, mark location is invisible on the screen.

❑ Press **[Ctrl-a]**

This places the cursor to the beginning of the last line. Now we have the region boundaries identified. The region is the last line (the text between *mark* and *point*).

❑ Press **[Ctrl-x] [Ctrl-u]**

This changes the selected region (the last line) to uppercase.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems.

However, it is available on most Linux systems.

This chapter introduces the Emacs editor; simple editing jobs and commands to start EMACS, EDIT A FILE AND END EMACS.

7.10 FILE MANIPULATION

When you read a file in for editing, Emacs creates a buffer for it, and the content of the specified file is stored in that buffer. Any editing is done on the text in the Emacs buffer. However, you must “save” this buffer to make your changes permanent. Otherwise, your changes will be lost when you end your Emacs session.

You start your Emacs editing session by opening a file. For example, you can open a file by typing the filename on the command line:

```
$ emacs Example [Return] . . . . . Invoke Emacs using the filename
                                Example
```

This command will look for the file named `Example` in your current directory. If a file named `Example` exists, Emacs opens the file. That means it copies the content of the `Example` to its buffer and you will see the first screen full of the text. If `Example` doesn't exist, Emacs creates the file and associates a buffer to `Example`, and you will see the Emacs default (empty) screen. When Emacs opens a file, that file's name will appear on the Mode line.

However, the text you type and the changes you make using Emacs do not become permanent until you save the file. Even when you save, Emacs leaves the original file under a changed name. This helps in case you want to go back to the original file or keep track of the changes.

7.10.1 Finding Files

You can also start your editing session by invoking Emacs without typing the filename:

```
$ emacs [return] ..... Invoke Emacs without filename
                           argument
```

Emacs will show the default screen and now you can use the commands for opening a file while you are in Emacs. The following partial screen capture shows the Emacs first screen. Your system will show this or something similar to this.

```
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.█
```

```
-1--"Scratch"                (Lisp Interaction)--L3--ALL-----
```

7.10.2 Using Finding File Command: [Ctrl-x] [Ctrl-f]

In order to open a file while in Emacs, you type the “Find File” command. For example,

▣ Press [Ctrl-x] [Ctrl-f]

This prompts you for the filename.

```
-1--*Scratch*                (Lisp Interaction)--L3--ALL-----
```

```
Find file: C:\emacs\Unbounded/█
```

Emacs waits for you to enter the filename. The filename you type appears on the bottom line of the screen. Recall that the bottom line is called *minibuffer* when it is used for this sort of input. You can use Emacs editing commands to edit the file name in *minibuffer*. You enter a filename for example you type Example and press the [Return] key to signal the end of your input.



1. UNIX is case sensitive. That is, for UNIX, example and Example are two different filenames.
2. In our example, the current directory is Unbounded. Your current directory will be different.
3. You can change the directory pathname by retyping your desired directory. (Use the **left/right** and [Del] keys to go back and forth over the text or delete text in the *minibuffer*).

```
-1--*Scratch*                (Lisp Interaction)--L3--ALL-----
```

```
Find file: C:\emacs\Unbounded/Example
```

If `Example` exists in the current directory, Emacs will open it for your editing. If `Example` doesn't exist in the current directory, Emacs creates a new file named `Example`, places the cursor at the top left corner of the window and lets you to type in the text.

```

■
-1:-Example          (Fundamental)--L1--ALL-----
(New file)

```

If you don't remember the filename or want to look at the list of the files in your current directory, you can't just type the filename and press the **[Return]** key. The following screen capture shows a sample of a directory listing. Your directory listing will probably be different.

```

c:/emacs/Unbounded:
total 1 free 66232348
-rw-rw-rw-  1 Student  root  257   May  3  00:40 #Example#
drwxrwxrwx  2 Student  root   0   May 19  00:06 .
drwxrwxrwx  2 Student  root   0   May 18  23:28 ..
-rw-rw-rw-  1 Student  root  257   May  2  21:07 Example
-rw-rw-rw-  1 Student  root  250   May  2  20:57 Example~

-1\%%  Unbounded      (Directory by name)--L6--ALL-----
Reading directory \emacs\Unbounded/... Done

```

To open any of the listed files, place the cursor on the filename that you want to open and press the **[Return]** key. When you wish to make your changes permanent, type the save command.

- Press **[Ctrl-x]** **[Ctrl-s]**.

This copies the text within Emacs buffer into the file. When saving is finished, Emacs displays a confirmation message on the echo line. For example:

```
"Wrote . . . Example"
```

The first time you open an existing file, Emacs renames the original file to a new name so that a copy of the original file is saved. The new name is made by adding the `~` (tilde) symbol to the end of the original filename. For example, notice the filename `Example~` in the directory listing.

7.10.3 Canceling a Command: **Ctrl-g**

While you are entering the filename (or any *minibuffer* input), you can cancel the command using the **[Ctrl-g]** keys. For example:

- Press **[Ctrl-x]** **[Ctrl-f]**

This will prompt you for the filename.

- Press **[Ctrl-g]**

This cancels the *minibuffer* and also cancels the **[Ctrl-x]** **[Ctrl-f]** command.

7.11 Emacs BUFFERS

You can open more than one file while in Emacs. When you open a second file, the first file remains inside Emacs. Emacs assigns a buffer to each file you open and stores each file's text inside the corresponding buffer. You can switch back and forth among files (buffers) and edit any one of them. When you have several buffers, only one of them is the current buffer and that is the one you edit. If you want to edit another buffer, you need to switch to it.



The following set of command sequences shows the Emacs multiple-file opening feature.

```
$ emacs test_1 [return]
```

This creates a file named `test_1`. Type in some text, such as the following.

```
1- This is test_1 line 1.
2- This is test_1 line 2.
3- This is test 1 line 3.
```

```
--:** test_1          <Fundamental>--L4--ALL-----
```

❑ Press **[Ctrl-x] [Ctrl-s]**.

A confirmation message appears on the echo line confirming that the file was saved.

```
--:** test_1          <Fundamental>--L4--ALL-----
```

```
Wrote file /student/home/david/test_1
```

Now you want to open another file, let's say `Example`.

❑ Press **[Ctrl-x] [Ctrl-f]**

This prompts you to enter the filename. Enter `Example` and press the **[Return]** key.

This opens the file `Example` and makes it the current file. Now you have two files opened in Emacs. Each file is associated with its own buffer. You can switch from one buffer to another by using the **[Ctrl-x] [Ctrl-f]** command and typing the desired filename in response to the prompt. For example:

❑ Press **[Ctrl-x] [Ctrl-f]**

This prompts you to enter the filename.

❑ Type `test_1` **[Return]**

This switches to the `test_1` buffer and it becomes the current file.

There is another and easier way to switch among the buffers. You can use the **[Ctrl-x] [Ctrl-b]** command followed by the filename. For example, in order to switch to the file `test_1` you type:

❑ Type **[Ctrl-x] [Ctrl-b] test_1 [Return]**

This switches to the `test_1` buffer and `Test_1` becomes your current file.

You can open many files in Emacs and sometimes you will want to see a list of them. To see a list of the buffers that currently exist in your Emacs job:

□ Type **[Ctrl-x] [Ctrl-b]**

This splits the file window into two windows, and the buffer's list appears in the bottom window.

```
1- This is test_1 line 1.
2- This is test_1 line 2.
3- This is test 1 line 3.
```

```
--:** test_1          <Fundamental>--L1--ALL-----
```

MR	Buffer	Size	Mode	File
.	Example	257	Fundamental	/student/home/david/Example
.	test_1	78	Fundamental	/student/home/david/test_1
*	*scratch*	191	Lisp Interaction	
*	*Messages*	244	Fundamental	

```
--:** *Buffer List*   <Buffer Menu>--L7--ALL-----
```

Each buffer has a name, usually the filename for the file whose contents it holds. Notice there are other buffer names. In fact, any text you see in Emacs windows is in some buffer. Some of these buffers are created when you open a file in Emacs, and some are created by Emacs to hold messages and other Emacs-related texts. For example, the buffer named `*Messages*` contains the messages that have appeared on the bottom line during your Emacs session and is not associated with any file. In fact, you can look at the messages in this buffer by switching to the `*Messages*` buffer. For example:

□ Type **[Ctrl-x] b *Messages* [Return]**

This displays the contents of the `*Messages*` buffer. Most probably, the next thing you want to do after listing the buffers and choosing the buffer name is to cancel the `Buffer List` and close its window. The command is:

□ Press **[Ctrl-x] 1**

This closes the unwanted windows. In this case it cancels the `Buffer List` and closes its window. Make sure the cursor is in the window that you want to keep on the screen. This command closes all the other windows.

7.11.1

Saving Buffers

When you switch from one buffer to another and make changes to the text of files' buffers, files' changes are not saved. The changes remain inside Emacs in the corresponding buffers. If you want to save any of the files' buffers, you must visit that buffer by using **[Ctrl-x] [Ctrl-b]** followed by **[Ctrl-x] [Ctrl-s]** to save the changes in that buffer. This is inconvenient if you have many buffers to save. But Emacs provides the **[Ctrl-x] s** command, which asks you about each buffer containing changes that you have not saved. It asks you, for each such buffer, whether to save the buffer. For

example, let's say you have two files: `Example` and `test_1` open in Emacs. You have changed text in each of them and now you want to save your changes.

□ Press **[Ctrl-x] s**

This displays the following prompt in the minibuffer.

```
1- This is test_1 line 1.
2- This is test_1 line 2.
3- This is test_1 line 3.
4- This is test_1 line 4.
█
```

```
--:** test_1                                <Fundamental>--L4--ALL-----
```

```
Save file /emacs/Unbounded/test_1? (y, n, !, ., q, C-r or C-h) y
```



1. Notice we have changed text in the `test_1` buffer by adding the fourth line. If there are no changes in `test_1`, you are not prompted to save this file or any other file's buffer with no changes.
2. As before, you can type **[Ctrl-h]** to display the help window that explains the different save options.

Let's say you typed **y**. This saves the buffer of the file `test_1`, and you are immediately prompted to save `Example`, the next file's buffer.

```
1- This is test_1 line 1.
2- This is test_1 line 2.
3- This is test_1 line 3.
4- This is test_1 line 4.
█
```

```
--:** test_1                                <Fundamental>--L4--ALL-----
```

```
Save file /emacs/Unbounded/Example? (y, n, !, ., q, C-r or C-h) █
```

Again, Emacs prompts you to save `Example` only if you have changed the text in `Example`.

Notice the prompt is asking to save `Example`, but `test_1` remains on the screen. You can display the text from the current buffer, in this case the file `Example`:

□ Press **[Ctrl-r]**

This displays the file `Example` in the current window.

7.12 FILE RECOVERY OPTION

Your computer or Emacs might crash while you are editing a file. In this case, your changes could be lost if you have not saved them. Fortunately, Emacs periodically saves each file that you are editing. This “autosave” feature saves the file in the current directory by adding

a # sign at the beginning and the end of the filename. For example, if you are editing a file named `Example`, the autosave file's name will be `#Example#`. When you save your file in the normal way, Emacs deletes the autosave file. The autosave mechanism informs you each time it saves a file by display the message “*Auto-saving . . . done*” on the echo line.

7.12.1 Using the File Recovery Command: `Alt-x`

If the computer crashes, you can recover your autosaved editing by using the recovery file command: `[Alt-x]`



Scenario: Assuming that you were editing the file named `Example` when your computer crashed, then you will have the autosaved file named `#Example#` in your directory.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start emacs, edit a file and end Emacs.■

```
--:** Example <Fundamental>--L4--ALL-----
```

When your computer is running again, you can recover your typing by using the following:

- Type `Alt-x recover-file` [`Return`].

```
--:-- Example <Fundamental>--L4--ALL-----
```

```
M-x recover-file
```

After you press [`Return`], the following prompt is displayed on the echo line.

```
--:-- Example <Fundamental>--L4--ALL-----
```

```
Recover file: C:\emacs\emacs-21.3\bin/
```

Type your file's name—in this case, `Example`—and press the [`Return`] key.

```
--:-- Example <Fundamental>--L4--ALL-----
```

```
Recover file: C:\emacs\emacs-21.3\bin/example
```

You are prompted for confirmation:

```
--:-- Example <Fundamental>--L4--ALL-----
```

```
Recover auto save file h:\usr\students\mydir\#Example#? (yes or no)
```

❑ Type **yes** [Return]

You will see the recovered copy of the file `Example`.

You must type **yes** or **no**. If you enter any other word, Emacs prompts you again to enter **yes** or **no**.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.

--:-- **Example**

<Fundamental>--L1--ALL-----

If you were editing more than one file, you can use the command

M-x recover-all-files

to recover all current autosaved files. If you do not enter the filename and just press [Return], you will have the list of the current directory, and you can select the file you want.

7.13 SEARCHING AND REPLACING

Searching for occurrences of a specific string throughout your text or finding and replacing one specified string with another is a cumbersome task, particularly when you are working with a large file. Emacs provides searching and replacing commands for these occasions.

7.13.1 Using the Search Commands: [Ctrl-s] and [Ctrl-r]

Emacs provides commands to do searches for strings forward or backward throughout the text. A string of text can be described as groups of contiguous characters or words. The Emacs search command is different from the search commands of most editors, such as vi. Emacs search is *incremental*. That means Emacs begins searching as soon as you type the first character of the search string and shows you the places of the search string (that you have typed so far) in the text. Table 7.8 summarizes the Emacs search commands.

Table 7.8
Emacs Search Commands

Key	Operation
Ctrl-s	Incremental forward search.
Ctrl-r	Incremental backward search.

The command to initiate a forward search is **[Ctrl-s]**, for a backward search, it is **[Ctrl-r]**. The following command sequences show how these commands work.

Open the `Example` file in Emacs using the following command:

```
$ emacs Example [return] . . . . . Open Example in Emacs
```

You want to search for the word Emacs in file `Example`:

▣ Press **[Ctrl-s]**

This prompts you to enter the search string.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.

```
--:-- Example <Fundamental Isearch>--L1--ALL-----
I-search: E
```

Start typing the search string (**Emacs**). Notice as soon as you type in the letter **E**, Emacs searches for all occurrences of the letter *E*, and they are highlighted to show their locations in the text. Then continue with your typing and type in letter the letter **m**. Now all the *Em* characters are highlighted. As you type the rest of the search string, the rest of the characters will be highlighted. This is how the incremental search works. Eventually, you type in the word **Emacs** and *Emacs* occurrences are found and highlighted.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.

```
--:-- Example <Fundamental Isearch>--L1--ALL-----
I-search: Emacs
```

Typing subsequent **[Ctrl-s]** commands makes the cursor to go to the next occurrence of the word Emacs. Press **[Return]** or use the **[Ctrl-g]** command to terminate the search.

You can use the **[Ctrl-r]** command for a backward search. That means the search starts at the cursor position and goes backward:

I-search backward:



1. An uppercase letter in the search string makes the search case sensitive.
2. The **[Ctrl-s]** starts the search by looking for any occurrence of the search string after the current cursor position.
3. Everything that you have learned about **[Ctrl-s]** also applies to **[Ctrl-r]**, except that the direction of the search is reversed.

7.13.2 Using Replace String Command: Meta-%

The command to find and replace text is `[Meta-%]`. This command requires two arguments: the search string and the string with which to replace the search string. Each argument is ended with `[Return]`. The format is:

Meta-% search-string [Return] replace-string [return]

The following command sequences show how the replace command works and describe the command's options. You want to replace all occurrences of the word Emacs (*search-string*) with emacs (*replace-string*) in the Example file. Open the Example file in Emacs using the following command.

\$ emacs Example [Return] Open Example in Emacs

The partial screen captures show the different prompts you receive to enter the command's arguments. We use the `[Alt]` key for the `[Meta]` key in our examples.

Press `[Alt-%]`

This prompts you to enter the *search-string*, in this case, Emacs.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.

--:-- Example <Fundamental>--L1--ALL-----

Query replace: Emacs

Type Emacs [Return]

This prompts you to enter the replace-string, in this case, emacs.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.

--:-- Example <Fundamental>--L1--ALL-----

Query replace Emacs with: emacs

Type emacs [Return]

This highlights the first occurrence of Emacs and prompts you to enter your intention.

This chapter covers the Emacs editor. Emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the Emacs editor; simple editing jobs and commands to start Emacs, edit a file and end Emacs.

--:-- Example <Fundamental>--L1--ALL-----

Query replacing Emacs with emacs: (? for help)

Typing in the `?` mark opens the *help view*. The *help view* explains the available options for text replacement. For example, pressing the **[Space]** bar changes the current highlighted *Emacs* to *emacs*; the next occurrence of the word *Emacs* is then highlighted and you are prompted again for your intention. Table 7.9 summarizes the available options for the replace command.

Table 7.9
The Replace Command Confirmation Options

Key	Operation
SPC (Space bar) or y	Confirms to replace the <i>search-string</i> with <i>replace-string</i> .
Del (delete key) or n	Skips to the next occurrence of the <i>search-string</i> .
, (comma)	Displays the results of the replacement.
[Return] or q	Quits the replacements.
. (period)	Replaces the current occurrence and quits.
!	Replaces all the remaining occurrences without asking.
^	Goes back to the previous occurrence.

Let's say you want to replace all occurrences without being prompted.

Press [!]

This changes all *Emacs* occurrences in the text to *emacs* and displays total number of replacements.

This highlights the first occurrence of *Emacs* and prompts you to enter your intention.

This chapter covers the emacs editor. emacs is not distributed with all the UNIX systems. However, it is available on most Linux systems. This chapter introduces the emacs editor; simple editing jobs and commands to start emacs, edit a file and end emacs.

```
--:-- Example <Fundamental>--L1--ALL-----
```

ReplacEd 5 occurrences

7.14 Emacs WINDOWS

You have already seen multiple Emacs windows, for example, the window for the *help view*. You can also use this feature to display multiple windows on the screen at the same time. Multiple windows can display parts of different buffers, or different parts of the same buffer. Table 7.10 lists the window commands.



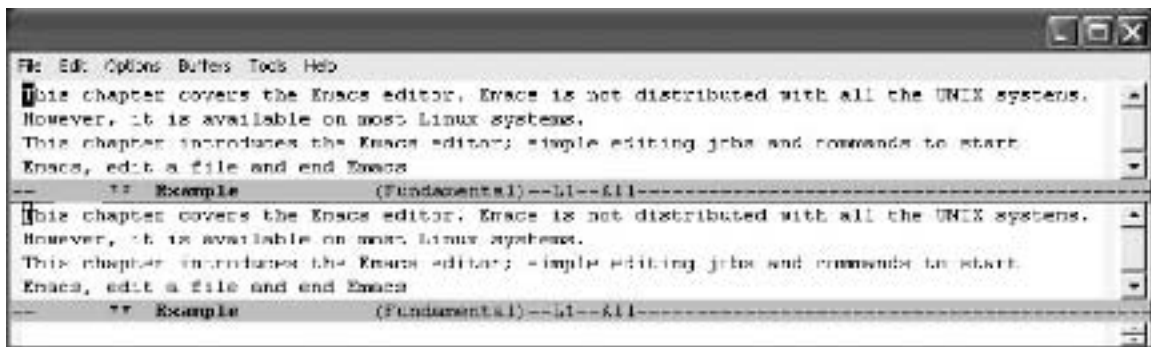
Scenario: Assume you have opened the file `Example` in Emacs. You want to split the `Example` window into multiple windows and open other files in each window. The following command sequences show how to split, display, and manipulate multiple windows.

Press **[Ctrl-x] 2**

Table 7.10
Window Commands

Key	Operation
Ctrl-x 2	Divides the current window horizontally into two windows.
Ctrl-x 3	Divides the current window vertically into two windows.
Ctrl-x >	Scrolls the current window to the right.
Ctrl-x <	Scrolls the current window to the left.
Ctrl-x o (letter o)	Places the cursor to the other window.
Ctrl-x 0 (zero)	Delete the current window.
Ctrl-x 1	Deletes all windows except the current window.

This splits the screen into two horizontal windows. Both windows display the file `Example`, and the cursor stays in the top window. The following screen capture shows the split windows in a window system. Notice that the cursor is in the top window, and the top window is your current, or active, window. All the ordinary editing commands apply to the active window.



❑ Press **[Ctrl][Alt] v**

This scrolls the bottom window. The command **[Ctrl][Alt] v** is useful when you are editing text in one window while using the other window for reference. In here, the content of the file `Example` is less than a full screen, and the whole file is visible in the window. If you have a large file, you might need to use the scroll command to see the hidden part of the file.

❑ Press **[Ctrl-x] o**

This moves the cursor to the bottom window. Now the bottom window is the active window. You can keep using **[Ctrl-x] o** to switch between the windows.

❑ Press **[Ctrl-v]**

This scrolls the active window (where the cursor is) up.

❑ Press **[Alt-v]**

This scrolls the active window down.

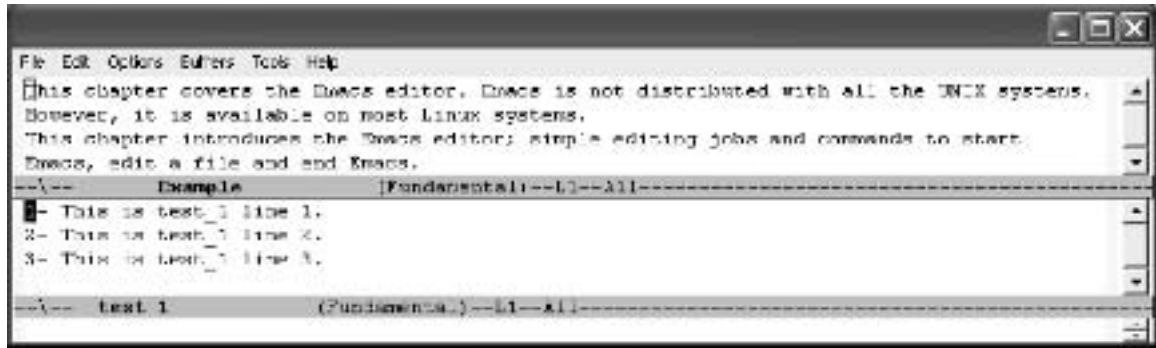
You do not have to display the `Example` buffer in both windows. You can use the **[Ctrl-x][Ctrl-f]** to find a file in any of the windows. Make sure the cursor is on the bottom window and then use the “Find file” command.

- Press **[Ctrl-x] o**

This switches the cursor to the next window.

- Press **[Ctrl-x][Ctrl-f]**

This will prompt you for the filename. Enter the filename, for example, `test_1`, and press **[Return]**. Now the bottom window shows the file `test_1`.



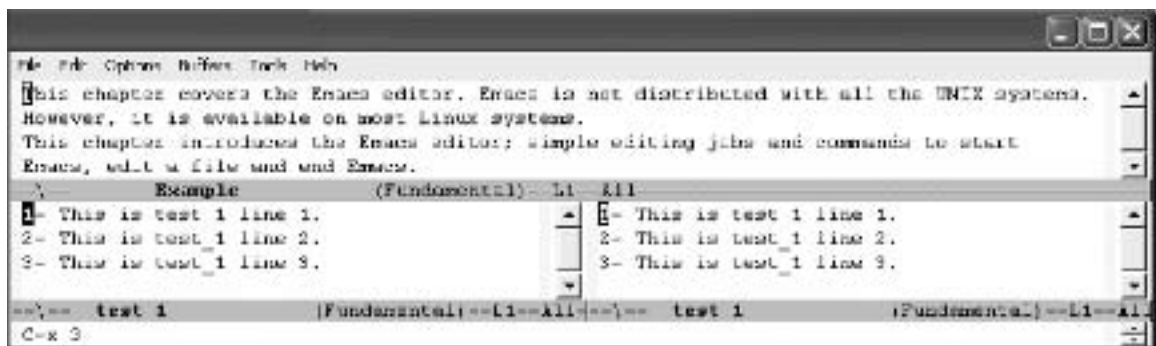
Scenario: Assuming you have the cursor in the bottom window. You want to split the bottom window vertically.

- Press **[Ctrl-x] 3**.

This splits the selected window (the bottom window) into two vertical windows.



1. *The active window is the window with the cursor in it.*
2. *All the ordinary editing commands apply to the active window.*
3. *Any active window can be saved using the save command **[Ctrl-x] s**.*



- Press **[Ctrl-x] 1**.

This command keeps the active window and gets rid of the rest.

7.15 THE `.emacs` FILE

When you start the Emacs editor, it automatically checks in your home directory for the existence of a file named `.emacs` and sets up the Emacs's editing environment according to the commands and setups in this file. You can customize the Emacs editing

environment by creating this file or editing the existing one. When Emacs starts up, it creates a buffer named `*scratch*`. The `*scratch*` buffer uses Lisp (programming language) Interaction mode and is used to type Lisp expressions. You can specify a different *major mode* for this buffer by setting the variable `initial-major-mode` in your `.emacs` file. The `.emacs` file contains Lisp function call expressions. Each function call consists of a function name followed by arguments, all inside parentheses:

```
(setq default-major-mode 'text-mode')
```

Placing this line in `.emacs` makes *Text Mode* the default mode for new buffers.

The `.emacs` is not the only startup file that Emacs looks for when is invoked. There is `.emacs.el` in your home directory and your site may also have files named `.default.el` and `site-start.el` that could be created for local customizations. If these files exist, Emacs finds and loads them before it loads your startup file.

7.16 COMMAND LINE OPTIONS

Emacs supports command line options. These options are to request various actions when invoking Emacs. For example,

```
$ emacs +3 Example [Return] . . . . . Open Example using line
                                     number option
```

This opens the file `Example` in Emacs and places the cursor on line 3. Table 7.11 lists some more command line options.

Table 7.11
Emacs Command Line Options

Key	Operation
<code>--no-init-file</code>	Starts Emacs without loading the customized initialization file.
<code>--user=user</code>	Starts Emacs by loading another <i>user's</i> initialization file.
<code>+linenum file</code>	Opens <i>file</i> in Emacs and then places the cursor on the line number <i>linenum</i> .

COMMAND SUMMARY

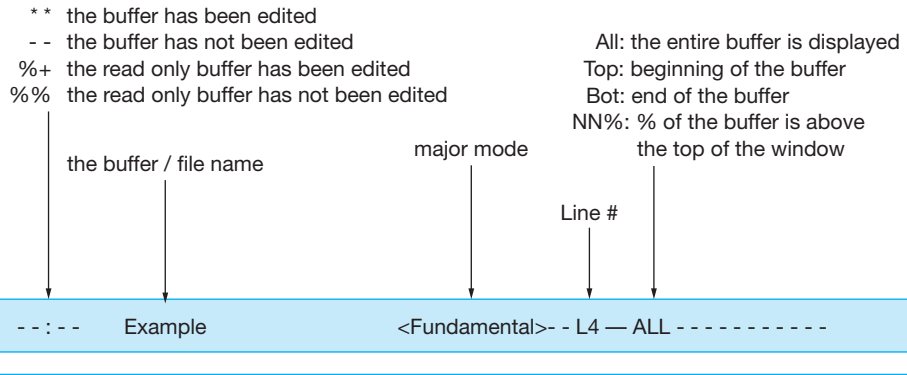
The following Emacs editor commands and operators were covered in this chapter.

The Emacs editor

Emacs is a popular screen-oriented text editor that is not shipped with every version of UNIX. Several versions of Emacs are available, including one distributed by the Free Software Foundation (FSF).

Emacs Mode Line Fields

The *mode line* appears right above the *echo line* and is the next-to-last line on the screen. The mode line displays status information such as what buffer is being displayed, what major and minor modes are in use, and whether the buffer contains unsaved changes. Normally, the mode line looks like the following line and consists of several fields:



Emacs Save and Quit Commands

Key	Operation
Ctrl-x Ctrl-s	Saves the file (contents of the current buffer) and exits Emacs.
Ctrl-x Ctrl-c	Quits the Emacs editor and abandons the contents of the file.
Ctrl-x Ctrl-w	Saves the file (contents of the current buffer) into file <i>filename</i> .

Help Commands

The Emacs editor includes a help system that can be invoked if you need explanation or help remembering the Emacs many commands.

Key	Operation
Ctrl-h	Invokes the emacs help (hold [Ctrl] while entering h).
Ctrl-h t	Invokes a short Emacs tutorial (hold [Ctrl] while entering h followed by t).
Ctrl-h k	Explains the function of a particular key.
Ctrl-h i	Loads the info documentation reader.
Ctrl-h Ctrl-c	Displays the Emacs General Public License.
Ctrl-h Ctrl-d	Displays information on ordering Emacs from FSF.

Cursor Movement keys

Emacs features many commands designed to navigate around in the text on the screen. Most implementation of Emacs conveniently maps the first four movement commands to the arrow keys on the keyboard.

Key	Operation
Ctrl-f or [→]	Moves the cursor forward one character.
Ctrl-b or [←]	Moves the cursor back one character.
Ctrl-p or [↑]	Moves the cursor to the previous line.
Ctrl-n or [↓]	Moves the cursor to the next line.
Ctrl-a	Moves the cursor to the beginning of the line.
Ctrl-e	Moves the cursor to the end of the line.
Ctrl-v	Moves the cursor forward one screen.
Meta-v	Moves the cursor backward one screen.
Meta-f	Moves the cursor forward one word.
Meta-b	Moves the cursor back one word.
Meta-<	Moves the cursor to the beginning of the text.
Meta->	Moves the cursor to the end of the text.

Keys for Deleting Text

Key	Operation
Backspace or Delete key	Deletes the character before the cursor.
Ctrl-d	Deletes the character under the cursor.
Ctrl-k	Kills all characters from cursor to the end of the line.
Meta-d	Kills the next word after the cursor.
Meta-Del (Delete key)	Kills the word before the cursor.
Meta-k	Kills the sentence that cursor is on.
Ctrl-x Del	Kills the previous sentence.
Ctrl-w	Kills all the text between the two positions.

Recovery Keys

In general, when you use the commands to remove a lot of text, the *kill buffer* is available so that the text can be recovered.

Key	Operation
Ctrl-y	Restores (yanks) what was killed.
Meta-y	After Ctrl-y , inserts the previously killed section of text.
Ctrl-x u	Undoes the previous editing changes.

Set Mark keys

You have to indicate the boundaries of the section of the text that you intend to rearrange in the file. The mark is used to indicate the beginning, and the cursor location indicates the end of the selected section of the text.

Point is the location of the cursor in the text.

Mark is a remembered location in the text.

Key	Operation
Ctrl-SPC	Sets the mark at the current point location ([Ctrl] key followed by [Space] key).
Ctrl-x Ctrl-x	Interchanges <i>point</i> and <i>mark</i> locations. This allows showing the mark location.

Case Conversion keys

Emacs provides commands to capitalize a word, change a word or a selected part of text from uppercase to lowercase or visa versa.

Key	Operation
Meta-u	Converts the following word to uppercase.
Meta-l	Converts the following word to lowercase.
Meta-c	Capitalizes the following word.
Ctrl-x Ctrl-u	Converts the region to uppercase.
Ctrl-x Ctrl-l	Converts the region to lowercase.

Search Commands

Emacs provides commands to do searches for strings forward or backward throughout the text. Emacs search is *incremental*. That means Emacs begins searching as soon as you type the first character of the search string and shows you the places of the search string (that you have typed so far) in the text.

Key	Operation
Ctrl-s	Incremental forward search.
Ctrl-r	Incremental backward search.

The Replace Command and Confirmation Options

The command to find and replace text is **[Meta-%]**. This command requires two arguments; the search-string and the string with which to replace the search-string. The format is:

Meta-% search-string [Return] replace-string [return]

Key	Operation
SPC (Space bar) or y	Confirms replacing the <i>search-string</i> with <i>replace-string</i> .
Del (delete key) or n	Skips to the next occurrence of the <i>search-string</i> .
, (comma)	Displays the results of the replacement.
[Return] or q	Quits the replacements.
. (period)	Replaces the current occurrence and quits.
!	Replaces all the remaining occurrences without asking.
^	Goes back to the previous occurrence.

Window Commands

Emacs provides command to divide the window into multiple windows. Multiple windows can display parts of different buffers, or different parts of the same buffer.

Key	Operation
Ctrl-x 2	Divides the current window horizontally into two windows.
Ctrl-x 3	Divides the current window vertically into two windows.
Ctrl-x >	Scrolls the current window to the right.
Ctrl-x <	Scrolls the current window to the left.
Ctrl-x o (letter o)	Places the cursor to the other window.
Ctrl-x 0 (zero)	Deletes the current window.
Ctrl-x 1	Deletes all windows except the current window.

REVIEW EXERCISES

1. What are the two keys that usually used in Emacs commands?
2. What is the Meta key on your keyboard?
3. What is command line to start Emacs using a filename?
4. What are the commands that invoke Emacs help?
5. Explain the following:
 - a. What is the *echo line*?
 - b. What is the *echo area*?
 - c. What is the *minibuffer*?

6. Explain the meaning of the following fields and symbols on the *echo line*:
 - a. ******
 - b. **- -**
 - c. **% %**
 - d. **buf**
 - e. **line**
 - f. **pos**
7. Explain the meaning of the following words and symbols in the *pos* field:
 - a. **All**
 - b. **Top**
 - c. **Bot**
 - d. **NN%**
8. What is the command to save the current file (buffer) and end Emacs?
9. What is the command to quit Emacs without saving the file (buffer)?
10. What do the following delete commands delete?
 - a. **Ctrl-d**
 - b. **Ctrl-k**
 - c. **Ctrl-w**
11. What is the difference between kill and delete?
12. What are the following commands?
 - a. **Ctrl-y**
 - b. **Ctrl-x u**
13. Explain *point* and *mark*.
14. What are the command to do the following:
 - a. Convert a word to uppercase.
 - b. Convert a word to lowercase.
 - c. Convert a region to uppercase.
15. Explain the Emacs buffers.
16. What are the Emacs search commands?
17. What is the Emacs replace command?
18. What is the command to split the Emacs window to two windows.
19. What is the `.emacs` file?
20. What are the Emacs command line options?



Terminal Session

In this terminal session, you will create a small text file and practice using the editing keys that Emacs provides. Use your imagination. Do not limit yourself to the small file

in this exercise and try to use all the keys covered and explained in this chapter to practice Emacs commands.

Use the Emacs editor to create a file named `Exercise` and type the text shown on Screen 1, exactly as is shown. Save this file.

Open the file `Exercise` and use cut-and-paste, cursor positioning, and other commands to correct and format the file to look like Screen 2. At completion save the file as `Exercise_Done` for later references.

Screen 1

```
Everywhere the trend is toward a simpler, and easy to care garden. Few
advises might help you to have less trouble with your gardening. I am sure
you have heard them before, but listen once more. Gardening: The easy
approach visit the plant nurseries, it is good for your soul. Let me tell
you that there is no easy to care garden. Use plants that are suitable for
your climate. Native plants are good CHOICE. Before planting, choose the
right site. Use your imagination, plants grow faster than what you think.
Gardening can be made easier and more enjoyable if you hire a gardener to
do the job. Use mulches to reduce weeds and save time in watering the
plants. Do not use too much chemicals to kill every weed insight. You are
the only one who sees the weeds, let them grow. They keep the moisture and
prevent soil erosion.
```

Screen 2

```
Gardening: The easy approach
Everywhere the trend is toward making simpler and easier-to-care-for
gardens.
However, let me tell you that there are no easy-to-care-for gardens.
Gardening can be made easier and more enjoyable if you hire a gardener to
do the job.
Some advice might help you to have less trouble with your gardening.
I am sure you have heard it before, but listen once more:
1. Before planting, choose the right site.
   Use your imagination. Plants grow faster than you think.
2. Visit the plant nurseries; it is good for your soul.
3. Use mulch to reduce weeds and save time in watering.
4. Use plants that are suitable for your climate.
   Native plants are a good choice.
5. Do not use too many chemicals to kill every weed in sight.
   You are probably the only one who sees the weeds.
   Let them grow.
   They keep moisture and prevent soil erosion.
```

Open `Exercise_Done` and apply the following commands.

1. Split the screen into two horizontal windows.
2. Save the top window in a file named `TopWindow`.
3. Save the bottom window in a file named `BottomWindow`.

4. Change the text in the top window to lowercase.
5. Change the text in the bottom window to uppercase.
6. Use the scroll commands to see different parts of the text in the windows.
7. In the bottom window, indicate a region by *point* and *mark* and delete that region.
8. Undo the deletion.
9. Place the cursor at the top of the screen. Delete the word *gardening*.
10. Undo the deletion.
11. Capitalize the word *gardening*.
12. Use the Help command: **Ctrl-h i**.
13. Remove bottom window.
14. Split the remaining window into two vertical windows.
15. In the left window, search for the word *weed*.
16. Remove the left window.

