

## 1. Emacs crib sheet

A few points first.

- You only need to run one copy of emacs. If you need to look at or edit more than one file then either split the emacs window or start a new frame, or more simply just open another buffer. All these actions are explained below.
- `C-x` means hold the control (Ctrl) key down and press the `x` key
- `C-x C-s` means hold the control (Ctrl) key down and press the `x` key, keep holding the control key down and press the `s`
- `C-x k` means hold the control (Ctrl) key down and press the `x` key, release the control key and then press `k`
- `M-x`, the 'M' refers to the Meta key which does not exist on a standard PC keyboard. It is usually mapped to the `ESC` key, the difference being that you press and then release the `ESC` key, followed by `x`<sup>1</sup>.
- entering a control or meta command will cause the cursor to jump to the mini-buffer at the bottom of the frame.
- **To abandon, or get out of, a command just type `C-g`**
- Emacs creates a buffer for each file you open. By default only the most recently opened buffer will be displayed unless you have more than one window open. You can have many, many buffers open.
- Many of the emacs hot-keys for movement and editing work on the command line of a terminal and some browsers and mail-tools.

## File Management

This is just a small selection of the available commands

Command	Usage	Meaning
<code>C-x C-f</code>	<code>C-x C-f filename</code>	Open or create a file
<code>C-x C-s</code>	<code>C-x C-s</code>	save the current buffer with the current name
<code>C-x C-w</code>	<code>C-x C-w filename</code>	write the buffer with a new name (save as)

### Exercise 1)

Create a new file using the correct Emacs command. Enter some text & save the file using the keyboard. Now save the file under a new name, again using the keyboard commands.

---

<sup>1</sup> You can map the left Alt key to be the Meta key instead, it will then work in the same way as the control key, ie you keep it held down while pressing the associated letter. To try it out, enter `xmodmap -e "keysym Alt_L = Meta_L Alt_L"` in a terminal window and restart emacs from the same terminal window. For a permanent effect next time you log in add the line to your `.xinitrc` file. Note that some window managers may override this.

## Buffer Management

A few more commands

Command	Usage	Meaning
C-x k	C-x k	kill the current buffer (empty it), emacs will prompt you if it thinks it is risky, eg unsaved after changes
C-x o	C-x o	swap to another window if you have more than one window open in emacs, else swap between current window and mini-buffer
C-x b	C-x b	swap to the last previously opened buffer, if it is still open. Emacs will tell which buffer it is about to open
C-x 2	C-x 2	split the current window horizontally to form 2 windows ( on the same buffer )
C-x 1	C-x 1	close all emacs windows except the current one

## Movement

Command	Usage	Meaning
C-f	C-f	move cursor forward one char(f = forward)
M-f	M-f	move cursor forward one word
C-b	C-b	move cursor back one char (b = back)
M-b	M-b	hopefully you guessed this one :)
C-n	C-n	move cursor forward one line(n = next)
M-n	M-n	move cursor forward one paragraph/code block depending on fi le-mode
C-p	C-p	move cursor back one line (p = previous)
M-p	M-p	hopefully you guessed this one :)
C-v	C-v	move cursor down by one screen
M-v	M-v	move cursor up by one screen

### Movement, continued

Command	Usage	Meaning
M-<	M-<	(Esc shift ,) move to start of document
M->	M->	(Esc shift .)move to end of document
C-x C-x	C-x C-x	swap mark and point(cursor)

### Editing & manipulating text

Command	Usage	Meaning
C-d	C-d	delete the char under the cursor
M-d	M-d	delete from the cursor to the end of the word
C-k	C-k	kill from the cursor to the end of the line
M-k	M-k	kill from the cursor to the end of the paragraph or code block
C-y	C-y	paste the last kill or copy from the kill-ring
M-y	M-y	use after C-y to get the previous item from the kill-ring. repeated use will work backwards through the circular cut buffer known as the kill-ring
C-spacebar	C-spacebar	place an invisible mark in the text, used for block delete and block copy to the kill-ring, amongst other things.
C-w	C-w	cut all text/code from point (current cursor position) to mark (see previous entry)
M-w	M-w	copy all text from point to mark into the kill-ring

### Clever stuff!

You will find some of the following very very useful when it comes to editing pattern files for the VHDL module so it is really worth your time playing with these. They cover 'on the fly' macro creation, repeating commands, 'find and replace', storing strings in registers and 'vertical block cut, copy, paste'.

### Creating and using a keyboard macro:

A keyboard macro can contain almost any sequence of characters or commands. It is created on-the-fly and stored by emacs. It is not saved when you exit emacs.

Command	Usage	Meaning
C-x (	C-x (	Start capturing (& executing) keystrokes
C-x )	C-x )	End keyboard macro capture
C-x e	C-x e	Execute (replay) keyboard macro

### Example

Suppose that you have a file containing lines like ...

```
begin
<0 ns> a: 10 0 ***
<+20 ns> a: 10 0 ***
<+20 ns> a: 10 0 ***
<+20 ns> a: 10 1 ***
<+20 ns> a: 10 1 ***
<+20 ns> a: 10 0 ***
<+20 ns> a: 10 0 ***
<+20 ns> a: 10 1 ***
<+20 ns> a: 10 1 ***
```

and that you realise that you need to add 2 more \*\*'s and a ; to each line. You could laboriously type that onto the end of each line but what if there were more than a few lines? The following key sequence will help automate the task. Start by placing the cursor anywhere on the first line to be edited.

**C-x ( <sup>2</sup> C-e \*\*; C-n C-x )**

This will

- "C-x (" start recording a macro,
  - "C-e" move to the end of the current line,
  - "\*\*;" type the characters \*\* ;,
  - "C-n" move down to the next line
  - "C-x )" will end the macro capture
- C-x e** will then repeat the macro for you.

---

<sup>2</sup> Remember don't type the spaces in the C-x cmd sequences.

But what if you need to do it 35 times? That is still a lot of typing ...  $C-n$  (where  $n$  is a number) will repeat the next command  $n$  times so

**C-35 C-x e**

will repeat the macro 35 times.