

Emacs

Living with text



FUKT Computer Society
Teddy Hogeborn

Who am I?

- Emacs user since at least 1994
- FUKT Computer Society
 - Unix system since 1995
- FUKT is a society/club for computer enthusiasts
 - A computer system with many servers
 - Holds lectures 😊
 - Meet, experiment and tinker
 - A lot of (almost 100) members
 - Had some nice rooms until recently

Why Emacs?

- Objections:
 - It's just a large, slow Notepad.
 - UltraEdit/Vim/*whatever* is so much faster, and has so many more features!
 - Surely specialized editors for different tasks is better!

Emacs is not an editor

- Don't think of Emacs as an *editor*
- It is an integrated development and editing environment
- Think of it as a “Desktop”
- Or, like a “Visual Studio for Everything”
 - Everything that is, and everything that will ever be, can or will be used from Emacs

Emacs is integrated

- Unifies *text editing*, replacing the need for:
 - Word processor
 - Text editor
 - Programming environment, “IDE”
- *All* things containing text that you can edit has a small text editor in it
 - Most often a bad one
 - Never as powerful as Emacs

“Emacs can do *anything*”

— often heard quote

- This is no exaggeration:
 - Text editing/word processor
 - Hex editor
 - File comparison
 - Spell checking
 - Personal organizer
 - Calendar
 - To do-lists
 - File manager
- Programming
 - *Every* programming language
 - Debugging
 - Version Control
 - CVS, Subversion, ...
- IRC
- Mail
- RSS feed reader
- Terminal (shell)
- Tetris ☺

Emacs can talk to anything

- Those are just *some* of the things that are *built-in as standard*
 - A vast number of extra modules exist
- Emacs also interfaces to many large external programs, for example:
 - Octave for math
 - R for statistics
 - Maxima for symbolic algebra
- In many cases, the programs themselves *recommend* using Emacs as an interface

Emacs is future-proof

- It's Free Software, extremely popular and well-known, so it's not going away
 - First ever Free Software package
 - Can be considered to be the Free Software Foundation's flagship product
- Ported to every OS and machine that has the resources to contain it
- About new programming languages, the first question asked is always:
 - “Is there an Emacs mode for it?”
 - These days, the Emacs mode is included

Emacs *is* everything

- For every new operating system, program, technology and service, either:
 - Emacs has support for it built-in
 - There exists an Emacs “mode” for it
 - Someone, somewhere, is writing Emacs support for it *right now*.
 - Extending is easy – it has its own internal scripting language

Emacs is also small

- Can perfectly well be used as a large, slow notepad
 - Has standard menus: File, Edit, etc.
 - Can be run as a terminal application from a text terminal
 - No understanding of any special features is *required*

Emacs *is* bad at being some things:

- Being a small editor
 - Slow to start
 - Hard to find and learn the more advanced features
- Being a specialized program for domain-specific tasks
 - Hard to find options and facilities
 - Not immediately adapted for the task you want to do

Emacs is not an editor

- It is not *meant* to be used this way!
- Remember: it is an *Environment*, a *Desktop*.
- Emacs has so many features and facilities, it would be *madness* to try and fit them all into the menu bar/tool bar
 - Instead, Emacs has modes which you must switch to
 - Easiest way is to open a file of the that type

How *should* Emacs be used?

- Just like an IDE, started *once* when logging in, and never closed
 - Work with many files open at once
 - Many, many files... – too many for tabs
- Seldom, if ever, switch to other applications when Emacs will do
- Learn to use the keyboard a lot
 - Emacs' key bindings adapted to US keyboard layout, so using it helps a *little*
 - Can be used in many other applications too

Why use the keyboard?

- Emacs has so many quick commands for marking, changing and moving text
 - Using the mouse for these would be much slower
 - All of them would not fit into the menus in a million years, anyway
 - If you aren't learning and using these commands to work faster, you might as well be using Notepad
 - I'm *still* finding useful features in Emacs to do faster what I once did by hand

About keyboards

- Keys like Ctrl, Alt and Shift are called *modifier keys*
- Emacs was written many ages ago, when keyboards were strange and different
 - Might not have:
 - Function keys, or only F1-F4
 - Alt key
 - Cursor keys
 - Home, End, Page Up, Page Down
 - etc.

Standard keys

Esc

~ `	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9) 0	- _	+ =	← Backspace
Tab ↔	Q	W	E	R	T	Y	U	I	O	P	{ [}]	 \
Caps Lock ↑	A	S	D	F	G	H	J	K	L	: ;	" '	↵ Enter	
Shift ↑	Z	X	C	V	B	N	M	< ,	> .	? /	Shift ↑		
Ctrl	Win Key	Alt								Alt Gr	Win Key	Menu	Ctrl

Keyboards

- Emacs does not require the use of any other keys – so it uses Ctrl+*key* a lot for doing things.
 - Example:
 - Page Down works fine, but Emacs also does the same thing when pressing Ctrl-V
 - “But that’s paste!”
 - You will have to unlearn *some* habits
 - Or, of course, reconfigure Emacs
 - See “pc-selection-mode”

Meta

- Emacs hopes that keyboards have *some* sort of extra modifier key besides Ctrl and Shift – It calls this key “Meta”
- On PCs, this is most often Alt, but some prefer it to be the Windows key.
 - Many non-PC computers like Sun have *both* Alt and Meta as separate keys
 - People used to this object to considering Meta as another name for Alt
 - Alt was chosen as Meta on PCs before Windows keys existed

Keyboard abbreviations

- Single modifiers:
 - Ctrl-x is called C-x
 - Alt-x, or rather Meta-x, is called M-x
 - Shift-x is called S-x
- Combinations of modifiers:
 - Ctrl+Meta-v is called C-M-v
- Key Sequences:
 - Ctrl-x followed by Ctrl-s is called C-x C-s
 - Ctrl-x followed by s is called C-x s
 - These do related, but different, things

Marking text

- Marking text is different
 - Can, of course, be changed to PC/Windows-style
 - Can be done using the mouse, but you should be using the keyboard, remember?
- When operating on a section of the text, you mark it – this is called the *region*
 - Cut, Copy, etc. operate on the region
- You set the “mark” at one end of the region with Ctrl-Space
 - The region is always the area between the (invisible) mark and the cursor

Buffers

- Text in Emacs does not have to be a file – it could just be some text
- All text in Emacs is contained in a *buffer*
 - Which might or might not be tied to a file
- Some commands are not “Save *file* as”, but “Save *buffer* as”, and so on
- A buffer can be thought of as a Tab
 - To *show* them as tabs would be crazy, since you normally work with many, many, files at once

Extended commands

- Keyboard too small to contain all commands
- Sometimes easier to remember the name than the key
- All commands (even those that have keys) can be run as extended commands
- Extended commands are run by M-x *command-name* RET
 - RET is the Emacs name for Enter/Return

Common extended commands

- compile
 - Runs “make”, shows compiler errors and warnings, and can jump to those lines
- goto-line
 - If you want to go to a line number manually
 - Does not, strangely enough, have a key
 - In the new unreleased Emacs 22, it is on M-g g
 - You could very easily “bind” it to any key you like
 - In fact, any command can be bound to any key

Some keyboard commands (of those not also in the menus)

C-g Break / interrupt / abort

Use this when you have done something strange to Emacs

C-z Put away Emacs

In a terminal, this suspends Emacs. In a window, it makes Emacs an icon.

- Use this instead of exiting Emacs

C-x 3 Split window vertically

(In contrast to C-x 2, which splits it horizontally)

*There are really **way** too many good commands and keys to list.*

Too many keyboard commands?

- I recommend starting with using the menus; explore them and learn new commands as fast as comfortable
 - Use the Emacs reference card to explore further: https://www.fukt.bsnet.se/lectures/2007-03-22_Emacs/
- Very many keyboard commands only become *really* handy once you know enough other commands
 - Examples: C-f, M-f and C-M-f,
C-x C-x and C-u C-SPC

Learn the keyboard commands!

- For example, the commands just to *move around* include:
 - Moving forward and backward by word, sentence, paragraph, expression, and page
 - Moving up and deeper into and out of expressions, functions and brackets
- Once you learn to use these, using the mouse to move will seem *painfully slow*
- Again, these are just the *move* commands.

Tutorial

- Emacs has a built-in tutorial you can go through, which is started by C-h t
 - C-h is in general the help key
- The tutorial is a bit strange to start with
 - It teaches using *nothing* but the standard keys, moving the cursor *without* the cursor keys, etc.
- It quickly goes on to teach more advanced and useful stuff

Further reading

- Emacs manual
 - Emacs has the whole manual built in
 - Is a bit large to start with unless you plan to read large bits of it
 - Also on the web: <http://www.gnu.org/software/emacs/manual/emacs.html>
- Emacs Wiki
 - <http://www.emacswiki.org/>
- Emacs home page
 - <http://www.gnu.org/software/emacs/>