

Emacs Hands On Session - Some Libraries

Marco Hassan

February 17, 2021

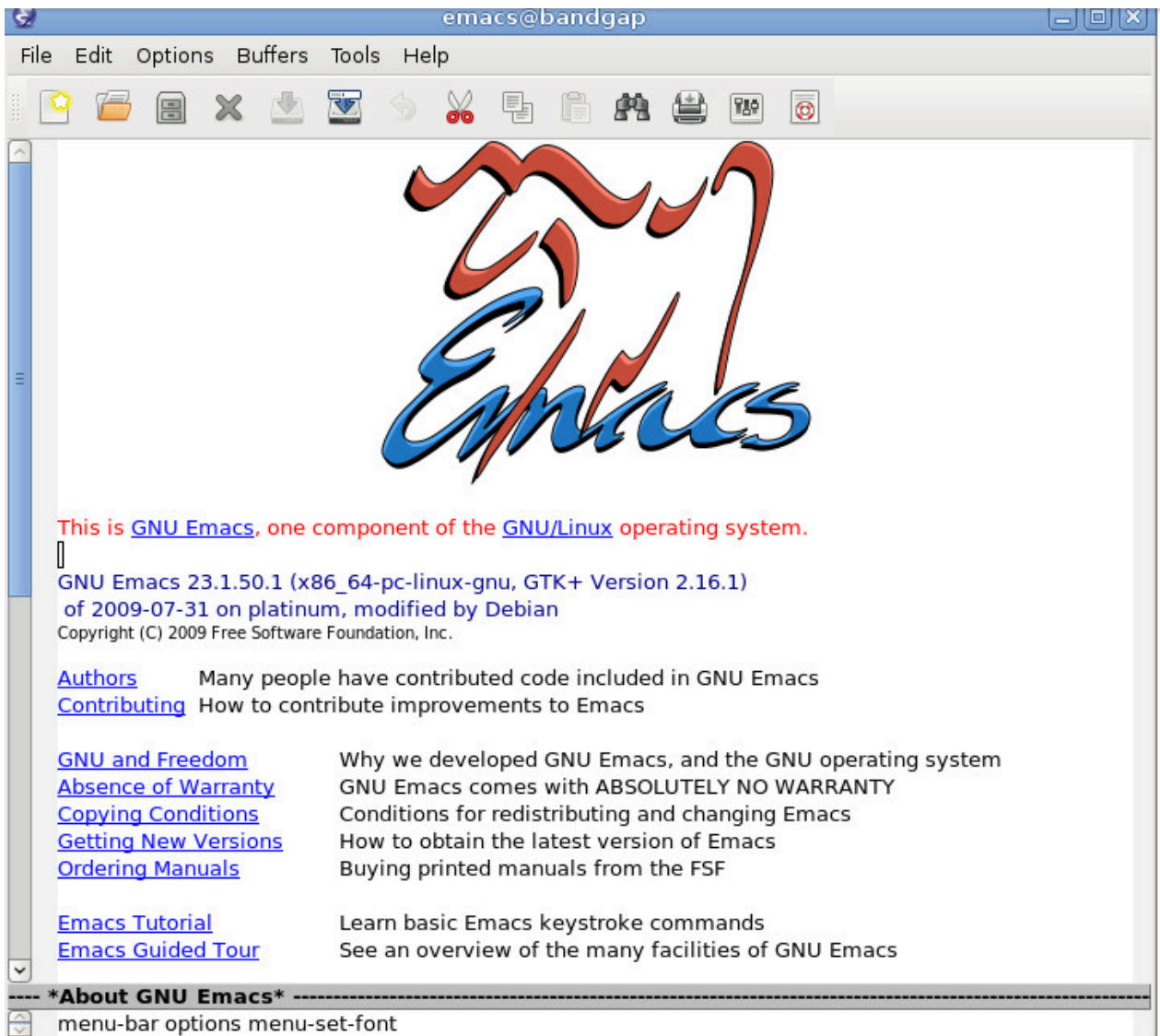
Contents

1	After Installation	1
2	Other Way - Centaur and possibly similar	2
3	My Emacs Configuration and some Packages I like	3
3.1	Task Management and Agenda	3
3.1.1	Have a word on Org-Mode	3
3.1.2	Explain the task management	3
3.1.3	Explain how you can view todos - for one file or project	3
3.1.4	Explain how you can decide what tasks status you can cycle with	3
3.1.5	Show the org capture	3
3.1.6	Show agenda	3
3.1.7	Recall the org journal some people work with it	3
3.1.8	Talk about tags	3
3.2	Org babel and literate Programming	3
3.2.1	Python - Example	3
3.2.2	Tell that this is just a dummy and fraction of possibilities	3
3.2.3	Can pipe results from one org block to another; mix languages and much more	3
3.2.4	On Compiler - based Languages	3
3.3	PlatUML - For Architects / Software Engineers	4
3.3.1	Show this dummy example	4
3.4	Scientific Documentation	4
3.4.1	Show Export in Latex	4
3.4.2	Show Export in HTML	4
3.5	Ein - Emacs Jupyter Notebook	4
3.6	Magit	4
3.6.1	Show magit Example	4
4	Warning	4

1 After Installation

Notice that after the installation your Emacs will not look like mine. It will be an ugly white program with little functionality.

2021-02-17 um 10.52.06.bb



The key idea is the **customization** idea of Emacs that we previously discussed.

Emacs is shipped as almost nothing at the beginning - i.e. all of the cool packages are not included. It is up to you to make sense of it and customize your emacs however you want.

To do that you have essentially to modify your `.emacs` file. This is usually in the `~` folder. If it does not exists create it.

You also will have an `.emacs.d` directory in the `~` where all of the downloaded packages will live. If it does not exists create it.

Then enable all of the different package managers. MELPA, Marmalade, ELPA, mirrors etc.

You are good to go and start to download the packages you want. Check at the `M-x list-packages` function.

2 Other Way - Centaur and possibly similar

The other way is not to download emacs from the official source and start from scratch at your speed.

There are people that went extremely harsh into emacs and created crazy emacs configurations.

They dockerized everything such that you should theoretically be able to run the docker image for such emacs, with all of the dependencies on the 1000s packages you will use next to your emacs.

You will then be blown at the beginning with a crazy out of the box emacs. I know one such option is centaur emacs. You can find it here.

I have no experience with it. I started two years ago with emacs and coming from economics I had no great idea of the software world. I started from 0 and by googling made my way. Might be that this is a very viable option. I will not switch as everything I have of my emacs comes from me and I know my emacs like my pockets.

This is the decision you will have to make. If you go with centaur and similar it might be tough to jump in such a huge sea without having any clue on how to navigate it. Maybe it is better to go slower but build your own emacs where you understand all of the config and all what you have in there.

You can then spend time, from time to time to check what others are doing. I choose this road, coping from here and there but always integrating everything myself such that I know what it is in my emacs. Up to you there is no right or wrong.

3 My Emacs Configuration and some Packages I like

3.1 Task Management and Agenda

3.1.1 Have a word on Org-Mode

3.1.2 Explain the task management

3.1.3 Explain how you can view todos - for one file or project

3.1.4 Explain how you can decide what tasks status you can cycle with

3.1.5 Show the org capture

3.1.6 Show agenda

3.1.7 Recall the org journal some people work with it

I personally never took the time to check at this into detail

3.1.8 Talk about tags

3.2 Org babel and literate Programming

Notice this is the best for product documentation or for projects where you have total ownership of something. As soon as you start to work in big projects of course there are limits to the extent to which you can use it. As soon as you tangle you tangle it all and overwrite the buffer.

I do not know, and did not check to this point if there is some conflict resolution scheme similar to git that you can integrate. That would be the endgame.

Also for working and learning by your own this is great.

3.2.1 Python - Example

Hello I write my documentation here. I do literate programming. Write what I am doing both for myself and for the friends I have to work with such that they will love me.

```
1 print ("Hello world")
```

Above I have written hello world. Next I will write this very complex function that is key to my project.

```
1 def win_1mio ():
2     return 10**6
```

Etc. etc. You got the idea.

Notice that through the session argument above babel understand that all of the blocks in the subtree belongs to the same runtime

```
1 a = 2
```

```
1 a
```

3.2.2 Tell that this is just a dummy and fraction of possibilities

3.2.3 Can pipe results from one org block to another; mix languages and much more

3.2.4 On Compiler - based Languages

Also other languages work. Even languages that need a compiler! Did not use this myself and have no idea how the workflow for such languages is but I know for sure that experts Software Engineers and Developers use it - so you might want to check online if that is anything interesting for your.

3.3 PlatUML - For Architects / Software Engineers

3.3.1 Show this dummy example

Once you set it up you can create your UML diagrams directly into Emacs. You can check at the PlantUML documentation.

Example:

[width=.9]/Users/marcohassan/Desktop/emacs_{presentation}/HandsOn/images/uml_{example}

3.4 Scientific Documentation

3.4.1 Show Export in Latex

3.4.2 Show Export in HTML

3.5 Ein - Emacs Jupyter Notebook

Show then how you can program in emacs and how also all the matplotlib stuff.

3.6 Magit

3.6.1 Show magit Example

4 Warning

You live in your own bubble once you start with emacs. Consider it well if that is something you want to do. There are plenty of other workflow options. This is a rather complex and involved one.

I hope that I made some of the advantages clear. Notice that on the downside is that it is something that you have to invest lots of time in. Continuously. Software evolves and so does your emacs.

I learned it when I was not working and just studying during my Masters and had quite a bit of spare time.

I love it and now that I did the effort will never go back. The majority of the people around you will not understand what you are doing with your emacs and so you will not be appreciated for being an emacs master by itself.

You have to do it for your own and because you believe that with it you can reach a higher productivity. And this is why eventually people will appreciate you at work even if they have no clue about why your screen always looks like you are still in the 90s.