Installation

CSC100 Introduction to programming in C/C++

Marcus Birkenkrahe

January 13, 2024

Contents

1 What are you going to learn?

- Why we don't just get coding
- Setting up infrastructure
- Install a C compiler
- Install the Emacs editor and IDE
- Customize the IDE
- Understand and learn to use GitHub
- Summary of concepts/code
- What's next

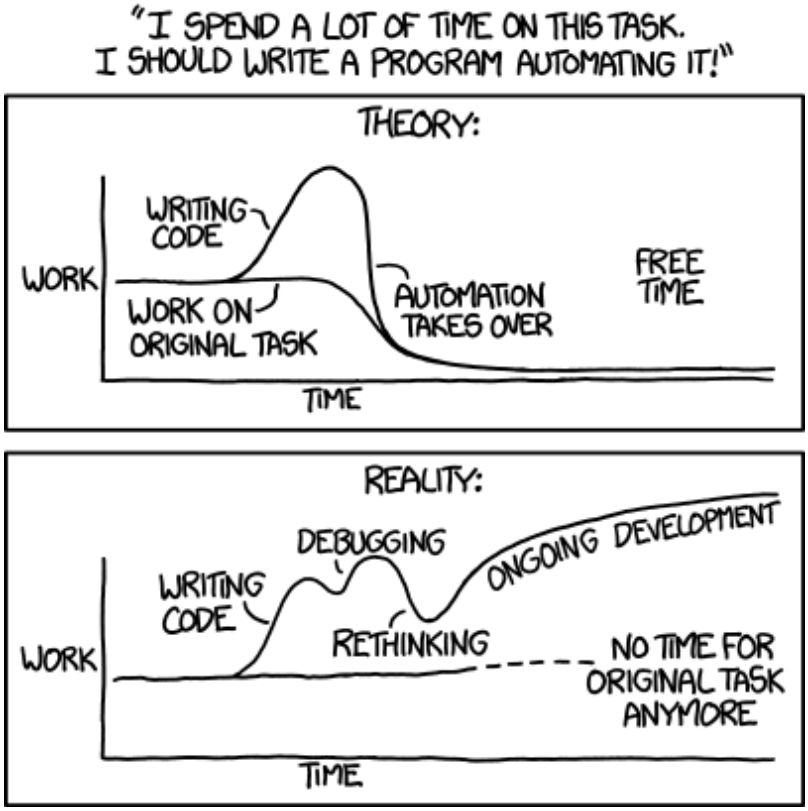


Figure 1: Automation by xkcd.

2 Why we don't just get coding

There are two BIG reasons - one is philosophical, the other is technical.

3 Philosophy

> "Understanding underlying technologies helps you develop a sense of what can go wrong. Knowing just high-level tools makes it easy to ask the wrong questions. It's worth learning to use a hammer before graduating to a nail gun. Learning underlying systems and tools gives you the power to build new tools, which is important because there will always be a need for tool builders, even if tool users are more common. Learning about computers so that the behavior of programs isn't a mystery enables you to craft better code." (Steinhart, 2019)

- Understanding technologies makes them safer
- Knowing low-level tools helps asking the right questions
- Learning low-level tools helps you build new tools
- Learning about computers enables you to craft better code

4 Technology

The other reason has to do with C itself:

TASK	PROGRAM	EXAMPLE
C source code is written and edited	editor	GNU Emacs
C source code is compiled, linked, debugged	compiler	GNU CC
C object code is run	shell	GNU bash

There are different routes to get C running on your computer, depending on your operating system (Linux, MacOS or Windows)[1].

The problem with most IDEs is that they're clunky. They take some time getting used to, have many functions that you'll never need, and any time spent on learning the IDE is wasted unless you keep working with it. Graphic especially IDEs do not teach you transferable skills.

[1] Provided the block has been formatted correctly.

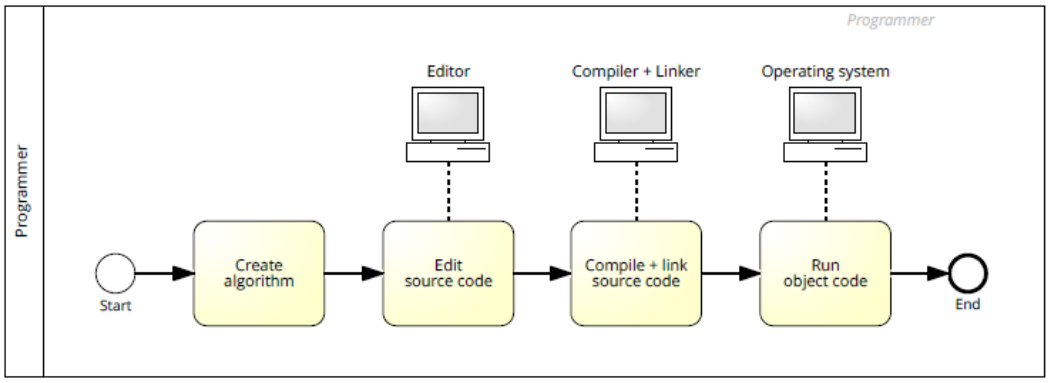


Figure 2: C programming workflow (BPMN model)

5 Infrastructure Setup

The sections below cover most of the infrastructure used in my courses. They are also available as FAQs on GitHub (not updated in 2024).

PLATFORM	SOFTWARE	PURPOSE	Meaning
GitHub	Git	Hosting	Global Information Tracker
MinGW	GCC	Compiler	Minimal GNU for Windows
GNU	Emacs	Editor	GNU's Not Unix

6 GitHub - What is it?

- Largest software development platform (2023: 100 mio developers)
- Built around the version control system Git by Linus Torvalds
- Bought by Microsoft in 2018
- AI support (e.g. GitHub Copilot) for students for free!

  Watch: "What is GitHub?" (GitHub, 2016)

  `../img/2_github.gif`

7 GitHub - Why are we using it?

Image: Org-mode file in GitHub

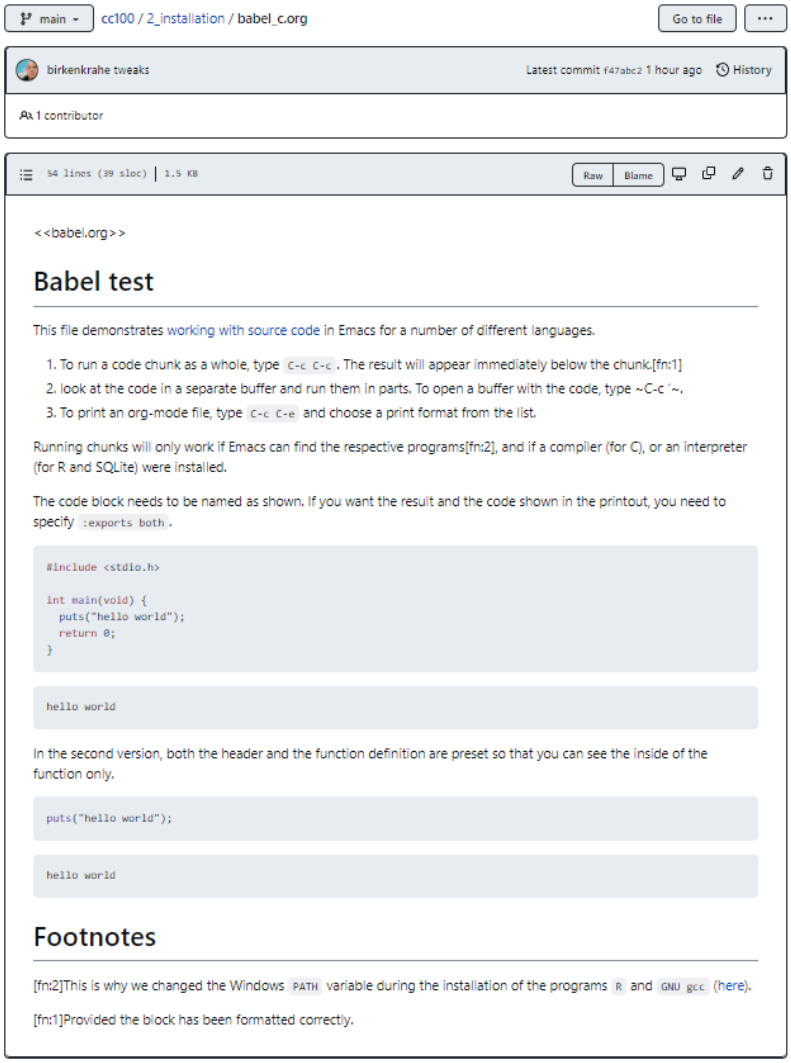


Figure 3: Screenshot of an Org-file rendered in GitHub

- It's free
- To host course materials
- Upload assignments (esp. Org-files)
- Discussion
- Wiki for collaboration
- Complements the learning management system[2]

8 GitHub - What will you have to do?

- Optional: Sign up with GitHub - use Lyon Email
- Pick an available username **using your own first and last name**, e.g. `MarcusBirkenkrahe`, or `DonaldTrump`
- Optional: Complete the "Hello World" exercise (FAQ) for bonus points.

  > If you do have a GitHub account already, do the exercise anyway using your existing account (it takes 10 min)!

9 GitHub - What else can you do?

- You can fork the cpp repository
- You can watch the cpp repository - and set Notifications to `Participating and @mentions` so that you see my comments (see image below). *Image: Notifications settings when watching a repository*
  - You can submit issues from the repository (e.g. if you notice mistakes or if you want extra information, or to share a link)
  - You can participate in discussions (sometimes I will make you)
  - You can add to the wiki (e.g. comments and links to interesting resources)

[2] This is why we changed the Windows `PATH` variable during the installation of the programs `R` and `GNU gcc` (here).

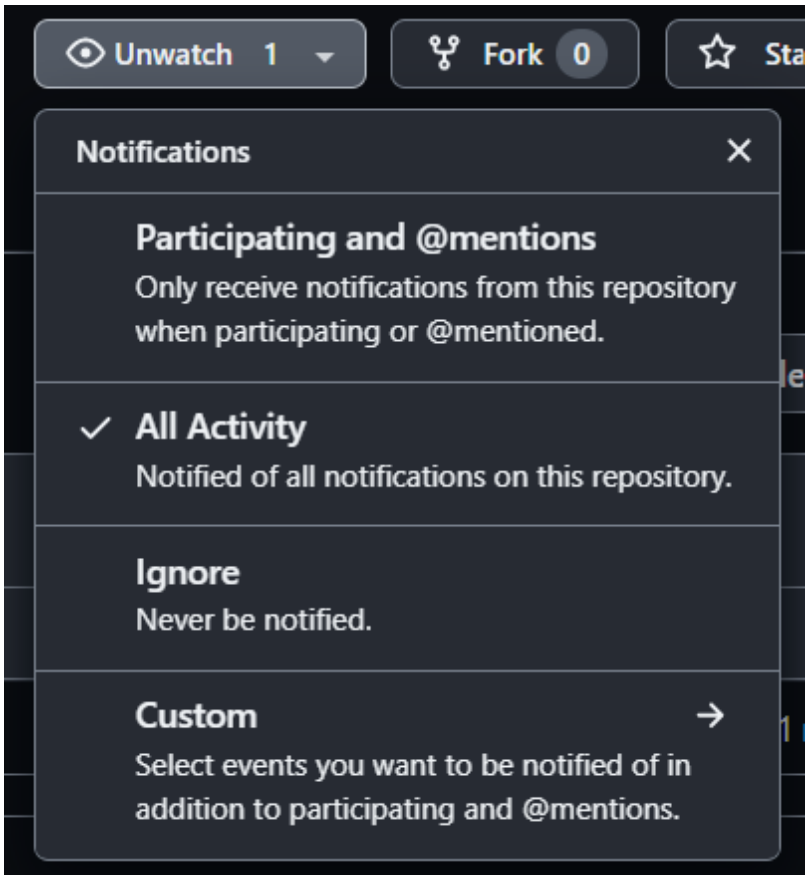


Figure 4: GitHub notifications setting

- You can install the mobile app on your smartphone[3]
- You can use it as a platform for projects or coding
- You can download the desktop client to manage repos on your PC (see image below).

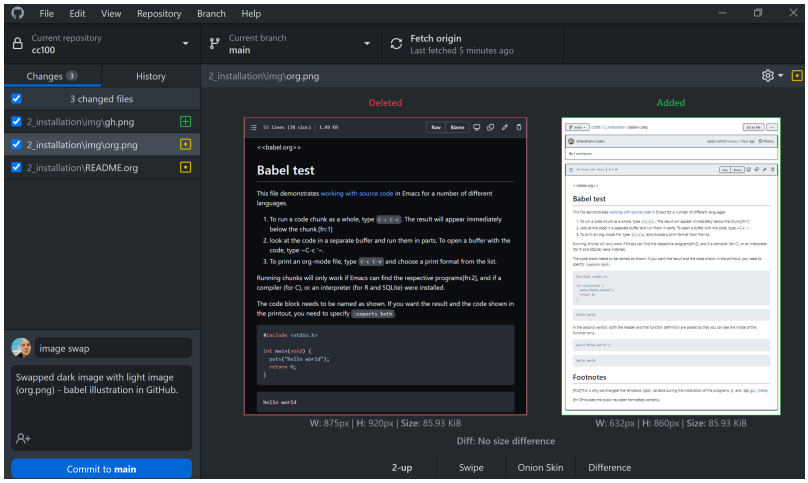


Figure 5: GitHub desktop client commit

10 Install GNU CC

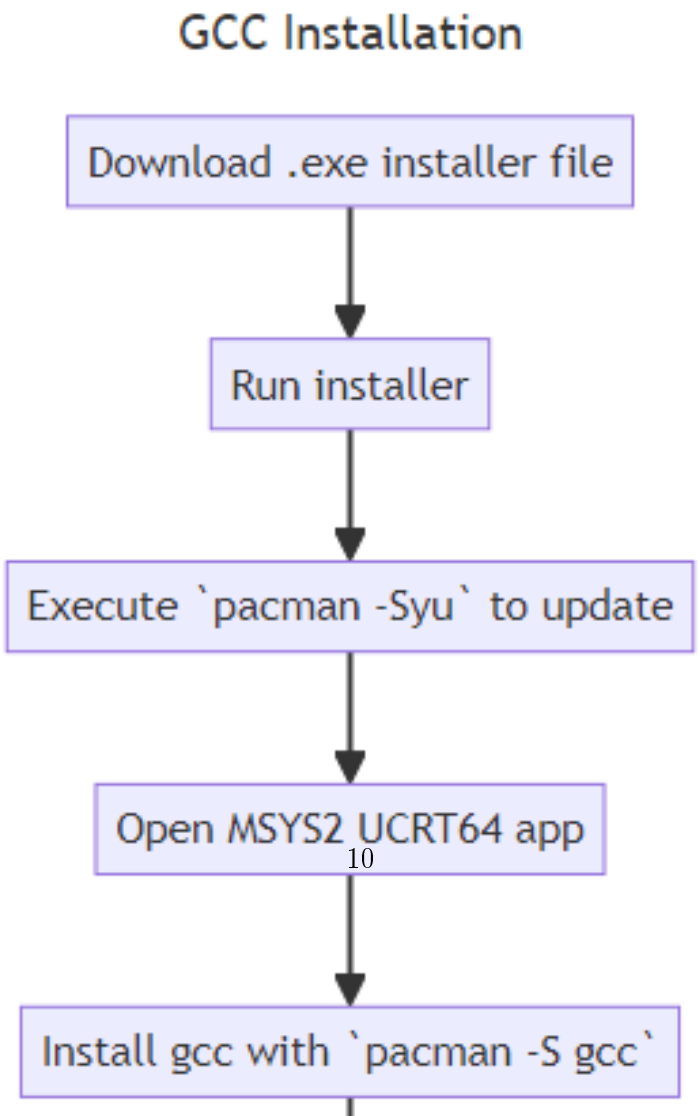
- Compiler & linker are usually bundled
- GCC ("GNU cc") is a popular C/C++ compiler
- MinGW ("Minimalist GNU for Windows") is a port of `GNU gcc` to Windows

11 Install GCC on Windows

Best way to install `gcc` is by installing `MSYS2` from msys2.org

[3]Only Markdown (`.md`) files are rendering in the mobile app. Org-mode files (`.org`) do not. Since you have Emacs, feel free to add a Markdown version of an Org file if you want one because you use the mobile version a lot.

Figure 6: Shameless plug for the Free Software Foundation - fsf.org



12 Install Clang on MacOS

Apple no longer allows GNU tools - instead, you can install the free Clang compiler as part of the Xcode development suite.

- Check if you already have a C compiler: open a terminal (search for `terminal` app) and enter `cc -v`.
- If no compiler is found, download it by entering `xcode-select --install`
- Test it by typing `cc -v`.

Figure 7: Checking Clang compiler exists on a Mac

13 Install GCC on Linux

Under Linux, the installation of a binary (executable) for that distribution and computer architecture, is done with a simple command from the command line, usually using the package manager, e.g. `apt` for Ubuntu 20.04: `sudo apt install build-essential`. This will install `gcc`, `g++`, and the powerful GNU `make` program.

14 Emacs for C programming

With the Emacs editor + Org-mode, you can almost program interactively (live code) with C - akin to Python or R. Org-mode inside Emacs works like a REPL (Read-Evaluate-Print-Loop).

15 Free online C coding

A resource to look at, and use (for free, at first) that uses the REPL concept, is repl.it (aka replit.com) See image below for the "hello world" program in C.

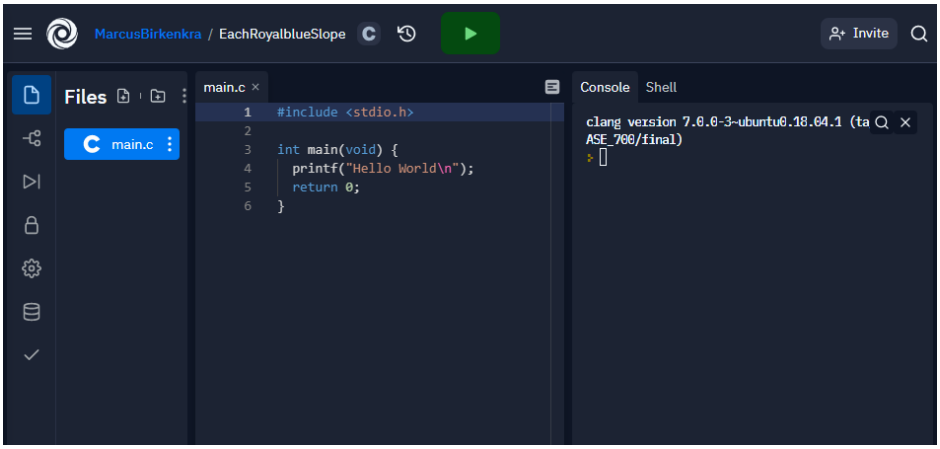


Figure 8: Replit.com C template

16 What is Emacs ?

PROPERTY	WHAT THIS MEANS
Extensible editor	You can adapt it to your needs[4]
Written in C with Emacs Lisp	It's fast and smart (via Lisp[5])
Ancient software	Written 1976, released in 1985[6]
Ca. 1.5M lines of code	By comparison: Windows ca. 50M; Linux kernel ca. 30M

[4]Here is an example from my `/.emacs` file: I defined the function `iwb` to indent a whole buffer according to the buffer's mode - something that can also be done with the key sequence `C-x h C-M-\` (`mark-whole-buffer + indent-region`).

```
;; re-indenting of whole buffer according to mode
(defun iwb ()
  "indent whole buffer"
  (interactive)
  (delete-trailing-whitespace)
  (indent-region (point-min) (point-max) nil)
  (untabify (point-min) (point-max))
  )
```

[5]Emacs Lisp is a Lisp dialect. Lisp was one of the first languages used for Artificial Intelligence research (cp. SHRDLU, an early natural language processing system).

[6]Written in 1976 by Richard Stallman, who then tinkered with it for ten years before releasing it. Emacs is also one of the two contenders, along with `vi`, of the famous editor wars of the UNIX culture. UNIX is the "mother" of all operating systems, the systems

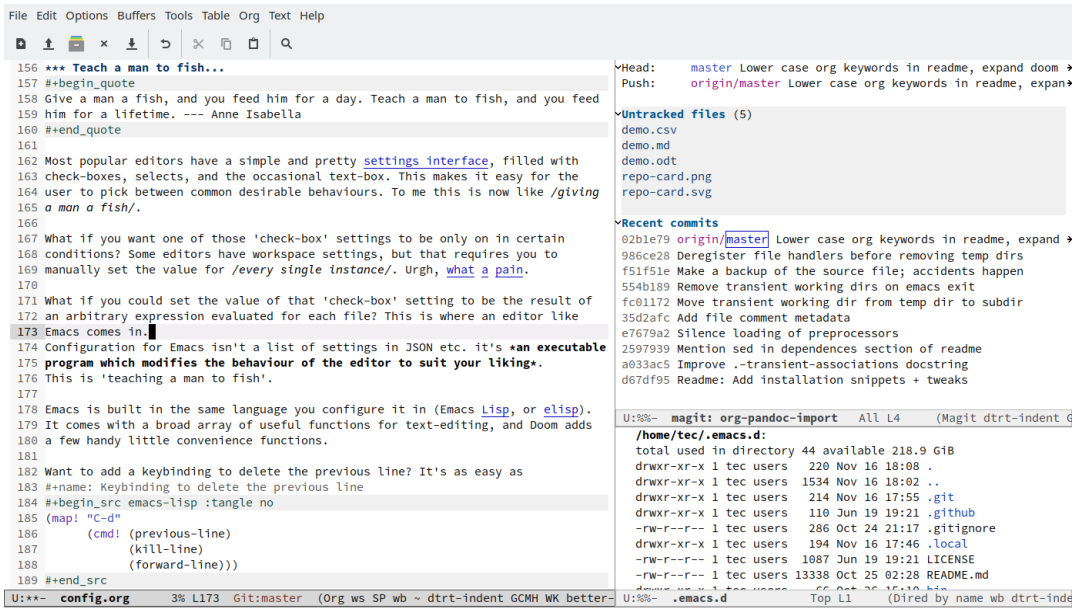


Figure 9: Emacs 27.1 showing Org, Magit and Dired

Challenge: which Emacs properties can you deduce from this image alone?[7]

17 How do you use Emacs?

See FAQ. I use Emacs for most of my computing needs:

- Writing (teaching, research)
- Planning (Calendar, ToDo)
- Organizing (Files)

See also:

that make computer run and do stuff.

[7](1) Emacs has versions (at the time the screenshot was taken: 27.1); (2) Emacs has named "buffers", and you can open several simultaneously [the names correspond to Emacs plugins or packages for organization (org), Git (magit) and file management (dired); (3) Emacs has layout themes with title and borders. (4) Each buffer is accompanied by a status line at the bottom [modeline].

- Article: "Getting started with Emacs" (Kenlon, 2020)
- Video: "The Absolute Beginner's Guide to Emacs" (System Crafters, 2020) with my notes.

This is how I looked when I first began to use Emacs:

Figure 10: DESY APE research group (1994). Can you find me?

Other uses:

- As window manager (only under Linux)
- As email client
- Remote access (with GNU Tramp)

18 How will we use Emacs?

We'll use it as:

- EDITOR to write source code,
- NOTEBOOK to write literate programs, and
- SHELL to build and run code.

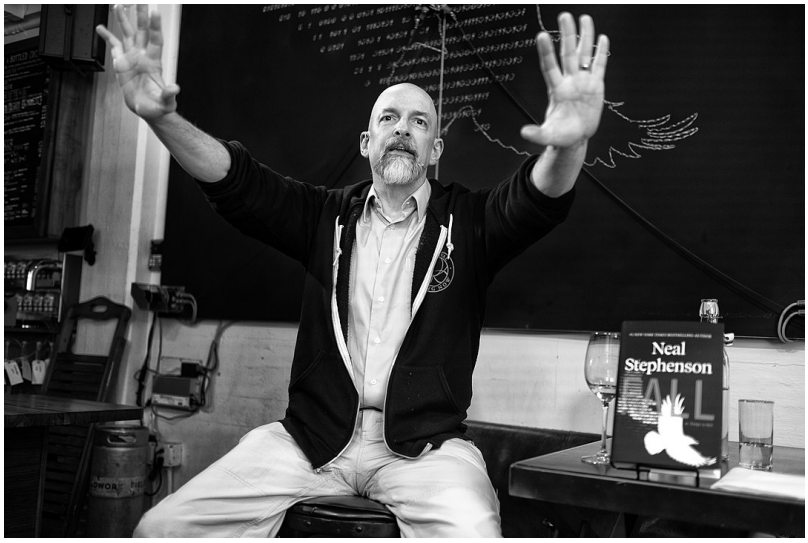


Figure 11: Neal Stephenson (Cyberpunk)

> "Emacs outshines all other editing software in approximately the same way that the noonday sun does the stars. It is not just bigger and brighter; it simply makes everything else vanish." – Neal Stephenson, In the Beginning was the Command Line (1998)[8]

We will not use Emacs as a substitute for religion even though there is a "Church of Emacs" (EmacsWiki)! Huh?! What?!

[8] Neal Stephenson is a sci-fi author who also coined the term "cyberspace", and developed a spacecraft and launch system for Bezos' Blue Origin.

Figure 12: The real Church: Notre Dame de Paris. Source: Wikipedia.

19 Does it really have to be Emacs?

You'll handle it. Keep calm and carry on coding.

If you look around, you'll see a lot of discussion on different source code editors and IDEs. Currently Microsoft's Visual Studio (VS) Code seems to

be the most popular contender. However, as one developer said:

> "One thing that cannot be replaced by any extension in VS code, VIM or any other editor: Emacs' Org mode. Org mode is for sure one of the most amazing pieces of software I have ever seen or worked with. It does things that no other text-based word processor can do, even if you are writing complex scientific reports. VS code has an extension which brings less than 5% of Org mode functionality, tops and that is mostly the code highlighting." (Timachi, 2019)

20 What about Emacs' famously "steep learning curve" ?

> "Emacs can be a challenge if you are used to using mouse pointer. One should be willing to leave the mouse and stick with the keyboard." (Timachi, 2019)

Using the keyboard for everything is much faster (than mouse-only, or mouse + keyboard) but takes getting used to. During the writing of this paragraph, I used the following keystrokes (with the command behind the keys, which your fingers will learn):

KEY	COMMAND
<q RET	`org-self-insert-command`
C-M-\	`indent-region`
M-q	`org-fill-paragraph`
C-a	`org-beginning of line`
C-e	`org-end-of-line`
C-x C-s	`save-buffer`

I'll show you!

Computer science, and IT, are largely about mastering, and creating new tools. Therefore, almost any effort is justified that goes into improving your **meta skills**[9] in this area.

[9] "Meta skills" are transferable skills that you learn, or improve, while you learn something specific (like Emacs or Org-mode). While the special skills might become obsolete or less important to you over time (because of a change of job, interest, or the market), meta skills stay important and fresh forever, because you can use them for every new special skill learning project.

21 Install GNU Emacs on Windows or Mac

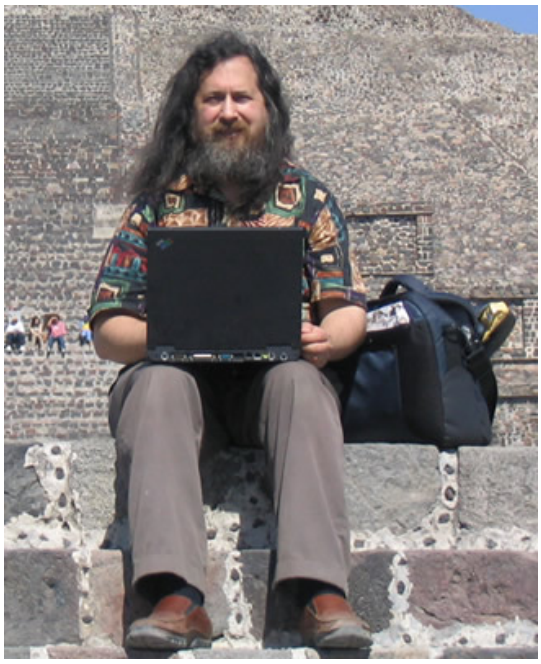


Figure 13: GNU Emacs creator, Richard M Stallman (MIT)

You won't have to do it but you can simply download an installer from here and install it using the installer. You need admin rights.

- Check out my new tutorial at GitHub (with videos)

22 Customize GNU Emacs

- GNU Emacs is much more than a text editor and an IDE. It's more like an operating system inside your operating system. Among the many things that Emacs is capable of, we only need one for this class: the ability to create and run interactive notebooks.
- This will give you the power of Jupyter notebooks or Colaboratory on your computer, without language limitations, and you can share notebooks with anyone, who has Emacs (or Markdown, for reading only).

- The central package for many day to day tasks is `Org-mode`. Here is a set of Org-mode tutorials (with videos) covering many interesting applications. Org-mode is especially popular among scientists, and among these, physicists (my original tribe), who developed it.
- And here is an excellent video tutorial by someone who is also getting started with Emacs for the first time like you: The Absolute Beginner's Guide to Emacs (System Crafters, 2021) - 1hr11min long - time well invested (I made some notes).

23 Create Emacs configuration file (`.emacs`)

- To create interactive computing notebooks in Emacs, we use the Org-mode and Babel packages. Both are already installed in your version of Emacs, but you have to tell Babel, which languages you want to work with.
- Customization like this is done with a configuration file `.emacs`, which is placed in your home directory (`$HOME`). Where this folder is actually located on your computer depends on your operating system[10].
- Download the configuration file from GitHub (tinyurl.com/EmacsLyon) and copy and paste it into a `.emacs` file or save it as `emacs.txt` and rename it to `.emacs`.
- Once you've created the `.emacs` file, you can start Emacs and code away "literarily". The customizations below are optional. But even just by using Emacs as your editor for assignments, you'll become quite an expert, almost a "hacker" (Wulff, 2021).

24 Create sample notebook for C

- To create a notebook using Org-mode, create an `.org` file. Then type `C-c C-,` and select your chunk from the list. You can also abbreviate this by entering `<s` on any line.
- You can work with source code in Emacs for a number of different languages:

[10] On my Windows machine, `$HOME` is `C:\Users\birkenkrahe\`. On my Linux box, it is `/home/marcus/`.

1. To run a code chunk as a whole, type `C-c C-c`. The result will appear immediately below the chunk.[1]
2. look at the code in a separate buffer and run them in parts. To open a buffer with the code, type `C-c '`.
3. To print an org-mode file, type `C-c C-e` and choose a print format from the list.

- Running chunks will only work if Emacs can find the respective programs[2], and if a compiler (for C), or an interpreter (for R and SQLite) were installed.
- The code block needs to be named as shown. If you want the result and the code shown in the printout, you need to specify `:exports both`.

```
#include <stdio.h>

int main(void) {
  puts("hello world");
  return 0;
}
```

```
hello world
```

In the second version, both the header and the function definition are preset so that you can see the inside of the function only.

```
puts("hello world");
```

```
hello world
```

25 Summary

- To program in C, we need a computer, a compiler, and an editor
- You'll have to download the compiler for Windows or MacOS
- You can download and install Emacs (ready for data science)
- Emacs is a highly customizable editor (using Emacs Lisp)
- Org-mode is a literate programming environment

26 Jargon

CONCEPT	EXPLANATION
Source code	Human-readable program
Compiling	Translating source
Linking	Linking compiled program to libraries
Library	Bundle of reusable macros or functions
Object code	Code ready for execution by a machine
Execution	Running object code on a machine
Interpreter	Machine that interprets and executes source code
Script	Source code for an interpreter
Emacs	Extensible text editor (via Emacs Lisp)
Literate Program	Readable code - expands into doc + executable
GNU	"GNU's not UNIX"
GNU/Linux	Free, open source operating system
Richard Stallman	Creator of the GNU project and Emacs
Org-mode	Emacs package for literate programming (and more)

27 References

- Biggs/Donovan (November 9, 2020). Modern IDEs are magic. Why are so many coders still using Vim and Emacs? [Blog]. URL: stackoverflow.org.
- DistroTube (October 4, 2019). Switching to GNU Emacs [video]. URL: youtu.be/Y8koAgkBEnM.
- Galov (August 9, 2021). 111+ Linux Statistics and Facts - Linux Rocks! [blog]. URL: hostingtribunal.com.
- GCC, the GNU Compiler Collection. URL: gcc.gnu.org.
- GitHub (Dec 19, 2016). What is GitHub? [video]. URL: youtu.be/w3jLJU7DT5E.
- GNU Emacs, an extensible, customizable, free/libre text editor. URL: gnu.org/software/emacs.
- Kenlon (March 10, 2020). Getting started with Emacs [blog]. URL: opensource.com.
- MinGW-w64 - Minimal GCC for Windows. A complete runtime environment for GCC & LLVM for 32 and 64 bit Windows. URL: mingw-w64.org.

- Steinhart (2019). The Secret Life of Programs. NoStarch Press. URL: nostarch.com.
- System Crafters (March 8, 2021). The Absolute Beginner's Guide to Emacs [video]. URL: youtu.be/48JlgiBpw$_{\text{I}}$.
- System Crafters (November 28, 2021). M-x Forever: Why Emacs will outlast text editor trends. Emacs conference 2021 [video]. URL: youtu.be/9ahR5K$_{\text{wkNQ}}$.
- Timachi (Dec 7, 2019). Why I switched from VScode to Emacs | Why I switched from VScode to Emacs [blog]. URL: hadi.timachi.com.
- Wulff (Jul 27, 2021). 8 Reasons Why Emacs is the Best Text Editor for Programming [blog]. URL: hackernoon.com.
- xkcd (n.d.). A webcomic of romance, sarcasm, math, and language [website]. URL: xkcd.com.
- Zamboni (March 21, 2018). Beautifying Org Mode in Emacs [blog]. URL: zzamboni.org.