Installing Linux/Emacs/GCC/R/ESS

Marcus Birkenkrahe

January 18, 2025

Prerequisite

An Internet connection. You need to make sure that you're connected when trying to install, update etc.

Installing Linux

- On MacOS you do not need Linux, because MacOS is a form of Unix.

- On Windows:

  1. From the search field, open the `PowerShell` and enter `WSL --install` at the prompt.
  2. Execute the following command on the Ubuntu shell:

     ```
     sudo apt update -y && sudo apt upgrade -y
     ```

  3. Reboot your computer.
  4. You'll need to pick a username and a (simple) password. You can change the password easily enough later with the `passwd` command.
  5. For the next steps, make sure you're opening the Ubuntu shell, and not the PowerShell. Save the icon to your Taskbar.

- On an Android phone: Get F-droid and download Termux, a full Linux simulator on your phone.

Installing Emacs and GCC

These tests will run provided you have the following applications available (installation instructions work under Linux only):

- Emacs (editor):

    1. Executable: `emacs`
    2. Check with: `emacs --version`
    3. Install with: `sudo apt install emacs`
    4. Configure: `wget -O .emacs tinyurl.com/lyon-emacs`

- GCC (compiler)

    1. Executable: `gcc`
    2. Can install with `sudo apt install gcc`
    3. Check with `gcc --version`

- Note that you can also get Emacs/GCC and bash for the tests below in a Windows environment, but then the installation is rather painful.

C Test

- You should be able to create a code block with the keyboard sequence `<s TAB` - if this does not work, you did not install the configuration file `.emacs` properly:

    - The file may be in the wrong location: It should be at `~/.emacs`
    - The `wget` command may have failed because of a typo (the flag is `-O` and the O is Big-Oh as in "Otto".

- Hello world program:

    `printf("Hello, world!\n");`

    `Hello, world!`

C++ Test

- Hello world program:

    `std::cout << "Hello, world!" << std::endl;`

Shell test

The following program generates an input file as stdout, and the C code reads the file as stdin:

```
echo Marcus > input
cat input
```

```
char name[100];
fgets(name,sizeof(name),stdin);
printf("Hello, %s\n", name);
```

Tangle and weave

- Tangle hello.c with `C-c C-v t` (or `M-x org-babel-tangle`)

- You can now view the file here:

  ```
  cat ./hello.c
  ```

Learning Emacs and Org-mode

- Emacs has an on-board tutorial that takes about 1 hour to complete - you can start it inside the editor with `CTRL + h t`.

- In addition, I have created a shorter tutorial, and made two videos showing how it all works: Part I and Part II. The tutorial itself is here on GitHub.

Installing R and Python (advanced)

You need this for data science courses.

- On Linux, you already have Python! Try `python3 --version` to check.

- Open the Linux terminal

- Enter `sudo apt install r-base -y`

- Test R on the terminal by entering 'R –version'

- Open Emacs

- Open the package list with `M-x list-packages`

- Find ESS with `C-s ess`

- Add it for installation with 'i'

- Install it with 'x'

- Open an R console with `M-x R`

- Create a new file with `C-x C-f Rtest.org`

- Create an R code block with `<s TAB R`

Run these commands in the code block

```
head(mtcars)
```

Create a graphic code block and plot something:

```
#+begin_src R :file test.png :results file graphics output
  plot(mtcars)
#+end_src
```