IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Setting up your environment

Integration week – TAF DCL

1 Objective

The goal of this lab session is to begin to set up a (UNIX/Linux) work environment adapted to DCL and to software development.

Depending on your situation (having or not having a laptop in class, being a Linux/MacOS/other UNIX system user or a Microsoft Windows user, feeling comfortable or not with a computer, ...) the lab session will be more or less useful.

The main content of this lab session aims Windows users and users who do not feel comfortable with their computer. If you are already a *UNIX/Linux power user*, feel free to help your fellow students.

Note: in this documents, `$>` denotes the prompt of your terminal. You never type the `$>` itself, but the characters that follow it.

2 Setting up a UNIX/Linux environment

Lab sessions will take place in rooms equipped with Linux computer. All necessary tools are installed on them. If you prefer to use your own computer, it is necessary to install all required software.

There are two main situations:

- If you already are a UNIX/Linux user, there is nothing to do in this section
- If your are a Microsoft user, we invite you to install and to configure a tool in order to have a UNIX/Linux environment within Windows.

2.1 Windows Subsystem for Linux (WSL)

WSL[1] is a way to benefit from advantages of a Linux system while keeping a Microsoft Windows environment. Basically, it runs a virtual machine with a Linux OS. WSL installation is straightforward when following the official documentation: `https://learn.microsoft.com/en-us/windows/wsl/install`.

By default, the installed Linux distribution is Ubuntu.

3 Installing tools

3.1 Linux

Usually, Linux distributions come with a package manager. Debian and Ubuntu use `apt`, but there are other managers, depending on your distribution. For the examples, we consider the Ubuntu distribution.

Regularly, update the list of available packages:

```
$> apt-get update
```

If you want to install the package named `foobar`, then type:

```
$> apt-get install foobar
```

But how to know the package you want is named `foobar`? By searching on the internet or directly with the `apt` command. For instance, the software is called `foo`, therefore the package probably contains its name. Thus, a search could be:

```
$> apt-cache search foo
```

We also advise you to regularly upgrade your installed packages. First, see what will be upgraded (without doing anything):

```
$> apt-get upgrade --dry-run
```

Then, do it:

```
$> apt-get upgrade
```

3.2 MacOS

The easiest and safest way to install software on MacOS is to use `brew`: `https://brew.sh/`

The installation instruction is on the main page: it consists in typing a single command in the terminal, then in reading the explanations.

[1] WSL documentation: `https://learn.microsoft.com/en-us/windows/wsl/`

3.3 Useful tools for DCL

Here is a non-exhaustive list of tools which are useful and/or necessary for DCL:

- Java
- Git
- Eclipse
- vscode
- Gradle

Try to install them on your system.

If you are a Windows user, you should have a functional Linux OS running on WSL. However, you probably also have already installed vscode on your Windows system. You don't need to reinstall vscode on your Linux system: you can specify to vscode to work with your WSL environment by installing the dedicated WSL extension (`https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-wsl`)

The full explanation/tutorial can be found here: `https://code.visualstudio.com/docs/remote/wsl`

To configure the default user when you sign in, just edit your `/etc/wsl.conf` by adding the following lines:

```
[user]
default=username
```

(where username is replaced by the username you want to use)