## *Linux-like environment on windows without VM*

This tutorial concerns people that have a windows laptop and don't want to install linux on their machine or use a virtual machine (VM).

There exists a tool called *Windows Subsystem for Linux* (WSL) created by Microsoft than can run native linux applications (with only few limitations [1]). However WSL can be used for this course.

This tutorial is based on this page and personal experience[2].

### Requirements

- up-to-date *Windows 10* installed on your laptop

- administrator rights on your computer

### Installing ubuntu

Open a Windows Powershell (search in the windows menu) and open it with administrator rights. Then execute the following command :

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

This will take a minute or so, and then it will prompt to restart your computer.

Once this is done, you can open the Microsoft Store in your programs and search for "Ubuntu". Install the official version of Microsoft.

Then you can open your Ubuntu terminal like any other program. The first time you open it, the terminal will prompt you to define a username and a password. Don't forget the password as you will need it later each time you want to install a program or do execute any command that starts with `sudo`.

### Finding the windows files from your terminal

The linux environment is installed somewhere in a hidden folder of windows (in `AppData`), so you probably want to work in your usual windows folder with the linux terminal. The root of your current disk (not the root of linux) can be accessed with the `cd` command :

```
cd /mnt/c
```

In fact, every disk (usb key, other disk, external hard-drive, ...) that is connected to your computer will appear in `/mnt/<letter of disk>`.

You will want to make a symbolic link to your usual working folder with the following command (be sure that your are in the home folder of linux with "`cd ~`") :

```
ln -s /mnt/c/Users/myusername/working_directory pcsc
```

You have to replace 'myusername' and the work folder 'working_directory with yours. This command will create in the home folder of the terminal a link named "pcsc", which you can access via `cd pcsc`.

---

[1]Mainly it is not possible to run applications with hardware acceleration (e.g. CUDA). You will have to wait until WSL2 to use these kind of features

[2]If you have any questions, feel free to write me an email at joachim.koerfer@epfl.ch

## Installing the compiling tools

In the linux terminal, execute :

```
sudo apt-get install cmake gcc clang gdb build-essential
```

It may not work (*"unable to retrieve packages"*), because sometimes the repositories of your Ubuntu are not up-to-date. If that happens, write `sudo apt update` before relaunching the previous command.

## Making CLion work with WSL

The last part is for making a connection between the windows installation of *CLion* and the Linux terminal. *JetBrains* have made a script that automatically configures your linux installation to work with *CLion*. Execute the following command[3] :

```
wget https://raw.githubusercontent.com/JetBrains/clion-wsl/master/ubuntu_setup_env.sh
bash ubuntu_setup_env.sh
```

Now you have to set-up the local `ssh` server that was created thanks to the previous command (be sure to change `username` to the username you have chosen when lauching ubuntu for the first time) :

```
ssh username@localhost -p2222
```

The last part is to specify in CLion the correct toolchains. Just following the instructions here. Be sure that you set the correct `ssh` credentials (figure 1).
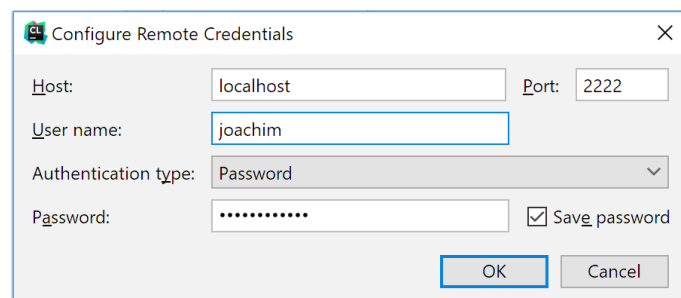


Figure 1: Setup correct credentials for the `ssh` connection, with the correct username, password and port.

## Using graphic application with the WSL

Sometimes, you may want to launch programs on linux that have a graphical user interface. You cannot do this without some manipulations. For this course, normally you wouldn't need to follow this part, except if you want to use a library that needs graphical output (like *Gnuplot*).

First, you need to install a program that can execute graphical requests (X11 protocol). Follow this link https://sourceforge.net/projects/vcxsrv/ and install the program 'VcXsrv' on windows.

Now, go on your linux terminal and type :

```
cd ~
nano .bashrc
```

Find the last line of the file and write in a new line : `export DISPLAY=:0`. Save the file. Reopen the linux terminal.

Each time you want to launch a linux graphical program, first open VcXsrv on windows (it launches in the background), then on the terminal launch the desired program.

---

[3]If you read this tutorial in the future, be sure to check if the command is still the same in this page