How to install WSL (Windows Subsystem Linux)

How to install WSL (Windows Subsystem Linux) and MCCE dev tools

WSL

WSL Installation

The official WSL installation instruction can be found on MIcrosoft website.

https://learn.microsoft.com/en-us/windows/wsl/install

The following is the my WSL installation on WIndows 10 Pro.

1. Open Powershell as Administrator, run
   `wsl --install`
   This installation will install Ubuntu under WSL.
2. Setting up user for Linux. After the reboot, WSL will ask for setting up the first user.
3. While wsl is open, right click on the penguine icon on task bar to pin it so that it is easy to start wsl linux next time.

From WSL, to access the files on the host system (Windows 10), use the path **/mnt/c** for the C drive.

From the host (Windows), to access the files on WSL, use the path **\\wsl.localhost\Ubuntu**

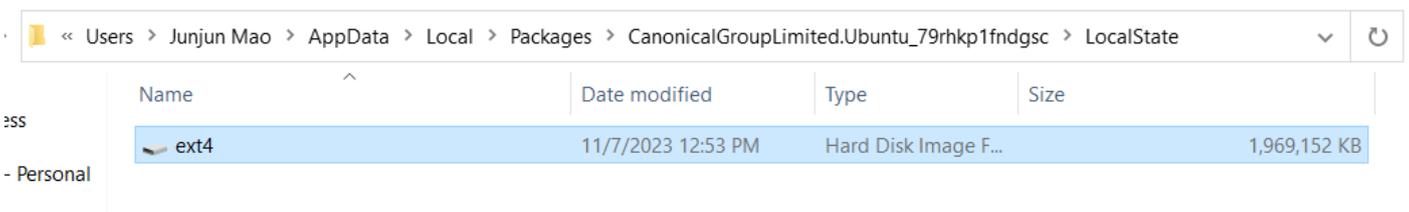To check WSL version, run `wsl -l -v` under PowerShell.

To check Linux version, run `cat /etc/issue` under Linux

WSL Virtual disk optimization

Linux under WSL uses a dynamic virtual disk. The virtual disk may be even larger than the physical disk on the host. In my case, the virtual disk is 1TB while I only have a 500 GB disk:

```
jmao@Junjun-Laptop:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            3.4G  4.0K  3.4G   1% /mnt/wsl
none            466G   46G  420G  10% /usr/lib/wsl/drivers
none            3.4G     0  3.4G   0% /usr/lib/wsl/lib
/dev/sdc       1007G  1.4G  955G   1% /
none            3.4G   80K  3.4G   1% /mnt/wslg
rootfs          3.4G  1.9M  3.4G   1% /init
none            3.4G  832K  3.4G   1% /run
none            3.4G     0  3.4G   0% /run/lock
none            3.4G     0  3.4G   0% /run/shm
none            3.4G     0  3.4G   0% /run/user
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
none            3.4G   72K  3.4G   1% /mnt/wslg/versions.txt
none            3.4G   72K  3.4G   1% /mnt/wslg/doc
drvfs           466G   46G  420G  10% /mnt/c
snapfuse         92M   92M     0 100% /snap/gtk-common-themes/1535
snapfuse        128K  128K     0 100% /snap/bare/5
snapfuse        151M  151M     0 100% /snap/ubuntu-desktop-installer/967
snapfuse         54M   54M     0 100% /snap/snapd/18933
snapfuse         73M   73M     0 100% /snap/core22/607
```

The actual usage after initial installation is about 2 GB, and the host will host will allocate more physical disk to it as needed.

« Users > Junjun Mao > AppData > Local > Packages > CanonicalGroupLimited.Ubuntu_79rhkp1fndgsc > LocalState			
Name	Date modified	Type	Size
ext4	11/7/2023 12:53 PM	Hard Disk Image F...	1,969,152 KB

However, once allocated, the virtual disk never shrinks. To shrink the virtual disk, I will need to do it through PowerShell.

1. Open PowerShell as Adminostrator.
2. Shutdown wsl instance: `wsl --shutdown`
3. Run command
   `Optimize-VHD -Path  C:\Users\Junjun\Mao\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\ext4.vhdx  -Mode Full`

Linux customization

- Home directory: Add `cd` to the user .bashrc so the terminal window starts at the home directory.
- Update Linux:

  `sudo apt update`

  `sudo apt upgrade`

  `sudo apt install vim aptitude`
- Build essential

  `sudo aptitude install build-essential`

Conda Python and Modules on WSL Ubuntu

Miniconda

1. Download miniconda Linux installer to wls ubuntu instance:

   `wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`
2. Run the downloaded script to install miniconda, yes to initilization set up.

   `bash Miniconda3-latest-Linux-x86_64.sh`

   Exit and start terminal after installation to enter the conda environment.

Python modules

`conda install numpy scipy matplotlib pygraphviz pandas xlrd openpyxl`

Compilers:

This is a little complicated. Currently PB solver delphi requires an old version of gfortran. That's the reason we wanted to install a conda version of compiler. Since we will adopt the new delphi which is compilable in C++, I decided to stick with the gcc from ubuntu OS.

Conda management

Revert base to clean state:

```
conda install --rev 0 --name base
conda clean --all
```

Microsoft Visual Studio Code

I am moving from Pycharm to Code due to

- Pycharm crashes at the start on my desktop.
- The ability of Code to code remotely vs ssh and wsl.

VS Code installation

Under WIndows (not under WSL), install VS Code.

1. Download VS Code from https://code.visualstudio.com/
2. Run the installer, select default set up options.
3. If WSL was installed before VS Code, the first launch of VS Code will prompt for WSL extension installation. Install this extension.

VS Code WSL extension

WSL extension can be installed from extension market, which can be activated by CTRL+Shift+X.

To connect to WSL, click the remote icon at the bottom left screen, choose connect to WSL.

Other software under WSL

Install Xserver for Graphic User Interface

In order to run X11 applications, Python plot and ssh -X for example, you need an X11 library under WSL.

Install X11 and test apps

`sudo apt install x11-apps`

Run `xclock` to test.

Pymol

Pymol is a molecular structure viewer. I believe one can install Pymol either from conda or from Ubuntu apt package.

To install under conda:

```
conda install -c conda-forge -c schrodinger pymol-bundle
```

I experienced conda version Pymol crashes, so I installed a Ubuntu Pymol package. Since conda apps precedes ubuntu apps under conda environment, the two Pymols can be both installed.

```
conda deactivate
```
```
sudo aptitude install pymol
```