Lab 1:

Setting up Windows Subsystem for Linux

In principle you can probably not use any of this, and just write your code on windows, but all supercomputers are Linux, so you should know how to run your code on Linux at least a little bit.

Based on

<https://wiki.ubuntu.com/WSL>

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

<https://www.youtube.com/watch?v=nKCe9UE-quA>

1. You will need to make sure Virtualization is enabled in your bios/uefi. For some god forsaken reason this is called SVM on my computer, so you may need to poke around for how to do it on your particular computer if it’s not already enabled. Basically you enter the bios when the machine boots up, and find the setting for virtualization and make sure it’s on. You can just try and do the other stuff without it first, and it just won’t work if this isn’t turned on. Note this directly conflicts with the other lab1/C++ in this which uses virtualbox so you can only do one or the other.
2. Make sure you are on Windows 10 2004 (may 2020 update), (19041) or later. You can check this in “System”
3. Turn on Windows subsystem for linux, you need to turn windows features on or off (start -> turn windows features on or off)

Graphical user interface, text, application

Description automatically generated

Probably you’ll need to reboot

Start a Powershell with admin privileges (start menu type powershell, then either right click on it or select the one that says “Run as a administrator”  
Graphical user interface, application

Description automatically generated

In powershell type (copy paste this whole thing).

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

1. Open the windows store. Download ubuntu (I suggest Ubuntu, not ubuntu 18.04 or the like, as the default “ubuntu” will keep you up to date.

Graphical user interface, application

Description automatically generated

While you’re at the window store I suggest you grab the Windows Terminal:

Graphical user interface, text, application, email

Description automatically generated

1. After you launch ubuntu you need to create a new username and password

(Make sure you don’t lose those), since the only one ever logging into this will be you, don’t worry too much about security.

1. At the command prompt type sudo apt update

A screenshot of a computer

Description automatically generated with medium confidence

1. If you are doing this really recently after the update to 2004 you will need:

<https://docs.microsoft.com/en-us/windows/wsl/wsl2-kernel>

you’ll know you need that if in Linux you type

cat /proc/version and it says 4.4 not 4.19 or later

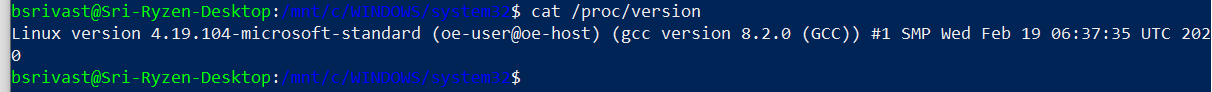
Old:

Text

Description automatically generated

New:

(you don’t need to do this in powershell I was just playing with terminals (windows) run): wsl.exe --set-version Ubuntu 2



In any of windows terminal, powershell or the like type:

wsl.exe --set-default-version 2

Windows updates \*should\* handle that for you by the middle of June/July 2020, but before that, there will be some fiddling

Lab hand in

Just show me a screenshot of Linux running.

E.g. Start windows terminal (Start menu type terminal). Then type “wsl” Show me the output of the command: cat /proc/version



And take a screenshot of some of the output of the command “ls”

# Setting up windowing.

(DO NOT TRY THIS IN SU 2020, I’ve left this here so I can update it in future years. you may want to install vcxsvr so you can run windowed apps from linux, but trying to run a full windows linux desktop doesn’t work yet without serious errors and it’s not worth the trouble),

On windows:

<https://sourceforge.net/projects/vcxsrv/>

Properties in that you need to let it accept sessions from anywhere

From window you can start this with: Start button - > Xlaunch

(Generally select one large window).

export DISPLAY=$(cat /etc/resolv.conf | grep nameserver | awk '{print $2}'):0.0

sudo -H DISPLAY=:0 XDG\_SESSION\_TYPE=x11 gnome-session

sudo apt install xubuntu-desktop

XLaunch option -swcursor