Installing Root on WSL

Mithil Vakde

August 17, 2020

Before you start

Hi. Welcome to this guide. These instructions are for installing Root on windows through WSL. There are other ways to install on Windows, but this has the added benefit of allowing you to use Linux on your Windows PC without dual booting. Linux is a powerful family of operating systems, used extensively by programmers. This guide will take you around 2 hours to complete. Please follow the instructions **exactly** if you're new to using command line interfaces.

All the instructions in this guide are just detailed versions of guides available online (linked in the documents). If you're comfortable with command line interfaces and installations, I suggest you to follow those guides and refer to this when you're confused. I do not take responsibility if you mess up while following this guide. If unsure while running any code, cross check it with the online guide linked. All [variable names] have been enclosed by square brackets.

While this guide might seem trivial or superfluous to those who are comfortable with Linux, I have deliberately made it so detailed to avoid any ambiguity. I was bombarded with too many doubts last time. I recommend using WSL 1 for this installation process if this is your first time, You can upgrade to WSL 2 anytime later. However, Root will work on either if installed correctly.

---

1 Installing Linux

1.1 Warning

The following steps include running *Powershell* as an administrator and using Linux as a super user. These give you extraordinary rights that let you modify your computer to a large extent. You can accidentally do a lot of damage to your computer if you're not careful. But if you're careful, it is perfectly safe. Please do not make any mistakes while running these codes. Once you're done, immediately close *Powershell*. The online guide for this part:

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

1.2 Steps

Step 1: Disable legacy console (if enabled)
Open *Windows Powershell* as an administrator (right-click app → run as administrator). Then right-click title bar → Properties → Uncheck "Use legacy console" (only if it is checked) → Click OK

Step 2: Enabling WSL on Windows
Enter the following command on *Powershell* (still running as administrator):

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

Step 3: Restart your PC

Step 4: Install a Linux Distribution
Open Microsoft Store and install your favourite Linux distribution: (For this guide, I have installed Ubuntu 20.04 LTS)

Step 5: Open the Ubuntu application
This opens up a black-coloured terminal, which is also called console, shell, bash, etc. This is the command line interface of Ubuntu, which is what you'll be working on for Root. The first time you launch Ubuntu, you'll have to wait for a minute or two for files to de-compress and be stored on your PC. All future launches should take less than a second. Now Ubuntu will ask you to create your account and password.

Step 6: Creating your default user
Enter in any username and set your password. Note here that the password will not be visible when you type it. Even the cursor does not move as you type. The username has no relation to your Windows user. This password will also be the *sudo* (Super User Do) password since your default user has superuser privileges. However, anytime you attempt a task requiring those privileges, you'll have to type in this password (only once per session). Once you create the username and password, the account will automatically sign-in on launching Ubuntu.

If all the steps were done correctly, a prompt shows up in green like this:

```
[username]@[Desktop ID]:~$
```

Step 7 Update Ubuntu
There might be some updates needed for Ubuntu after the install. Enter the following command in this terminal to do so:

```
sudo apt update && sudo apt upgrade
```

It will ask for the *sudo* password. This is the same password you set a few minutes ago.

Congratulations! You have installed Linux as a subsystem on Windows. Before we install Root, we will install a package manager - Conda - that makes installing Root (and any other package) a simple one line affair.

2 Installing Miniconda

Miniconda is a minimal installer for Conda that does not install the monstrous amount of excess packages that Anaconda does. It is smaller, faster and lesser buggy than Anaconda. Hence we will be installing it. The online guide for this is here:

`https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html`

2.1 Steps

Step 1: Download the Installer
Using a normal browser on Windows, Go to the following page and download the latest version of Miniconda:

`https://docs.conda.io/en/latest/miniconda.html#linux-installers`

Step 2: Find directory path
Find the path of the directory into which the Installer was downloaded. Let this path be known as [directorypath]. For me, the [directorypath] of the downloaded installer was:

`C:\Users\mithil\Downloads`

Step 3: Change directory
Open Ubuntu and enter the following command to change the working director to the one which has the Installer:

```
cd [directorypath]
```

However, [directorypath] has to be in WSL format. Here is how the command looked like for me:

```
cd /mnt/c/Users/mithil/Downloads
```

The rules for converting the directory path from Windows to WSL format:

- Backslash → Forward-slash

- "C:" → "/mnt/c" (Note that c is not capital in the converted pathname)
- Path is case sensitive (unlike Windows)

If successful, the directory will be visible in the prompt (it will replace the ∼ that is currently visible)

Step 4: Verify Hashes (optional, but recommended)
Enter the following command on the Ubuntu terminal (you should still be in the downloaded file directory)

```
sha256sum [filename]
```

Where [filename] is replaced by the name of the installer file (case sensitive). For me, the command was:

```
sha256sum Miniconda3-latest-Linux-x86_64.sh
```

This will output a long string (the SHA256 hash). Compare it to the hash given in the table from the website where you downloaded it (Link above). If it is the same, proceed to the next step, else repeat from step 1 with a different version of the installer.

Step 7: Execute the installer
Making sure your active directory is still the one where the downloaded file exists, enter the following command on your Ubuntu Terminal:

```
bash [filename]
```

Where [filename] is the name of the file of the Miniconda installer. For me, the command was:

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

A set of prompts will now appear on your screen. Accept the default settings. Specifically, let it be installed into the default location

```
/home/[username]/miniconda3
```

When prompted whether to let the installer initialize Miniconda by running *conda init*, say yes.

Step 7: Checking
Your installation process is done. Restart Ubuntu. If you did everything correctly, "(base)" will appear on the leftside of your prompt.

```
(base)[username]@[Desktop ID]~$
```

Irrespective of that, enter the following command to check if Miniconda works:

```
conda
```

If successful, lots of text will be displayed on screen, else it will display errors like "command not found: conda".

3 Installing Root

This is the easiest part of the Installation process. The online guide is here:

`https://root.cern/install/#conda`

3.1 Three Parts

Step 1: The Install
Once Miniconda is working, enter the following command on the Ubuntu terminal:

```
conda create -c conda-forge --name [environmentname] root
```

Replace [environmentname] with any valid variable name. This creates a virtual environment with that name and installs Root on that environment (downloaded from the Conda Forge channel).

Step 2: The Activate
Enter the following command on the Ubuntu Terminal:

```
conda activate [environmentname]
```

The active environment is changed to the one where Root is installed. "(base)" should change to "([environmentname])". You are required to do this everytime you open Ubuntu to run Root.

Step 3: The ~~Prestige~~ Test

```
root
```

If this command runs successfully, congratulations, you have installed Root! However, WSL cannot handle graphics on its own (yet). To run graphics, you need to install and configure an X server. In this GUide, we will be installing Xming.

4 Installing Xming

The guide for this part is here:

https://virtualizationreview.com/articles/2017/02/08/graphical-programs-on-windows-subsystem-on-linux.aspx

The guide starts from the heading "Running Graphical Programs" (Ignore the SSH part).

4.1 Steps

Step 1: Install Xming

Download Xming from the following website normally through a browser on Windows:

`https://sourceforge.net/projects/xming/`

Once downloaded, launch the setup file and use the default configurations. Then launch the Xming application once the install is done. It will run in the background and will be visible in the System Tray. You will need to launch Xming before starting an Ubuntu session everytime you wish to use graphics (once a session).

Step 2: Exporting DISPLAY

Open Ubuntu and enter the following command on the terminal:

```
1    export DISPLAY=:0
```

This is a step you need to do everytime you open an Ubuntu session when you want to use graphics (once a session).

Step 3: Checking

To check if graphics works properly, you can try out any functions on Root that has a graphical output. Otherwise, you can install X11-apps package which is a small graphical apps package. Enter the following commands:

```
1    sudo apt-get install x11-apps
2    xeyes
```

If you see a program with a pair of eyes on screen, congratulations! You are done with this guide.

4.2 Running graphics

If you want to run graphics during an Ubuntu session, here are the steps: (You will need to do these steps once during every session)

Step 1: Launch Xming if not already running.

Step 2: Export the DISPLAY variable:

```
1        export DISPLAY=:0
```

5 Troubleshooting

Most installation errors are well documented online. You could refer to Github, Stax Exchange, etc. in general. For WSL installation, refer to

`https://docs.microsoft.com/en-us/windows/wsl/install-win10#troubleshooting-installation`

6 Epilogue

In this guide, I have installed WSL1. You can upgrade to WSL2 and it will still work fine (albeit faster). Instead of Ubuntu, you can install other Linux distributions. Instead of Miniconda, you can install Anaconda (will take much more space and a much longer install time). You can also install Root without Miniconda by following the steps in the Cern website (although this is complicated for beginners). Instead of Xming, you can install any other X server like vcXsrv.