Windows 10 and the Ubuntu Subsystem

Jared Bennett

Table of contents

Windows 10 has a powerful new feature that allows a full Linux system to be installed and run from within Windows. This is incredibly useful for building/testing code in Linux, without having a dedicated Linux machine, but it poses strange new behaviors as two very different operating systems coexist in one place. Initially, this document mirrors the Windows Install tutorial, showing you how to install Ubuntu and setting up R, RStudio, and LaTex. Then, we cover some of the issues of running two systems together, starting with finding files, finding the Ubuntu subsystem, and file modifications.

Installing Ubuntu

There are 2 parts to installing a Linux subsystem in Windows. I will write this using Ubuntu as the example, as it is my preferred Linux distro, but several others are provided by Windows.

Sources:

- Official Windows Instructions
- Ubuntu Update Instructions

1) Enable Linux Subsystem

By default, the Linux subsystem is an optional addition in Windows. This feature has to be enabled prior to installing Linux. There are two ways to do it.

- **CMD Line**
  The simplest way to enable the Linux subsystem is through PowerShell.
  - Open PowerShell as Administrator

- Run the following (on one line):

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

- Restart the computer

- **GUI**
  If you don't wish to use PowerShell, you can work your way through the control panel and turn on the Linux subsystem.
  1) Open the **settings** page through the search bar
  2) Go to **Programs and Features**
  3) Find **Turn Windows Features on or off** on the right side
  4) Enable the **Windows Subsystem for Linux** option
  5) Restart the computer

2) Install Linux Subsystem

Once the Linux subsystem feature has been enabled, there are multiple methods to download and install the Linux distro you want. I highly recommend installing Ubuntu from the Microsoft store. There are several other flavors available as well, but Ubuntu is generally the easiest to learn and the most well supported.

1. Open the **Microsoft Store**
2. Search for **Ubuntu**
   - You're looking for the highest number followed by LTS, currently **20.04 LTS** (or 18.04 LTS is fine too). This is the current long-term-release, meaning it will be supported for the next 5 years.
3. Click on the tile, then click **Get**, and this should start the installation.
4. Follow the prompts to install Ubuntu.

After installing Ubuntu, it is advisable to update it. This is something you should do on a regular basis.

1. Open a Bash terminal.
2. Type `sudo apt update` to update your local package database.
3. Type `sudo apt upgrade` to upgrade your installed packages.

Using the Linux Terminal from R in Windows

To get all the functionality of a UNIX-style commandline from within R (e.g., for bash code chunks), you should set the terminal under R in Windows to be the Linux subsystem.

A Note on File Modification

DO NOT MODIFY LINUX FILES FROM WINDOWS

It is highly recommended that you never modify Linux files from Windows because of metadata corruption issues. Any files created under the Linux subsystem, only modify them with Linux tools. In contrast, you can create files in the Windows system and modify them with both Windows or Linux tools. There could be file permission issues because Windows doesn't have the same concept of file permissions as Linux. So, if you intend to work on files using both Linux and Windows, create the files in the `C` drive under Windows, and you should be safe to edit them with either OS.

Finding Windows from Linux

Once you have some flavor of Linux installed, you need to be able to navigate from your Linux home directory to wherever Windows stores files. This is relatively simple, as the Windows Subsystem shows Windows to Linux as a mounted drive.

- Source

1. Open a Bash terminal.
2. Type `cd /` to get to the root directory.
3. In root, type `cd /mnt`. This gets you to the mount point for your drives.
4. Type `ls` to see what drives are available (you should see a `c`, and maybe `d` as well).
5. Type `cd c` to get into the Windows `C` drive. This is the root of the `C` directory for Windows.
6. To find your files, change directy into the users folder, then into your username.
   - `cd Users/<your-user-name>`
   - This is your home directory in Windows. If you type `ls` here, you should see things like
     - `Documents`
     - `Downloads`
     - `Pictures`
     - `Videos`
     - etc…

Finding Linux from Windows

This is slightly more tricky than getting from Linux to Windows. Windows stores the Linux files in a hidden subfolder so that you don't mess with them from Windows. However, you can find them, and then the easiest way (note, do not read as safest or smartest) to find those files in the future is by creating a desktop shortcut.

- Source

1. Open File Explorer
2. In the address bar, type `%userprofile%\AppData\Local\Packages`
   - `%userprofile%` will expand to something like `C:\Users\<your-user-name>`
3. Look for a folder related to the Linux distro that you installed
   - These names will change slightly over time, but look for something similar-ish.
   - For Ubuntu, look for something with *CanonicalGroupLimited.UbuntuonWindows* in it.
     - Canonical is the creator/distributor of Ubuntu.
4. Click **LocalState**
5. Click **rootfs**
   - This is the root of your Linux distro.
6. Click into **home** and then into your user name.
   - This is your home directory under Linux.
7. **DO NOT MODIFY THESE FILES FROM WINDOWS**
   - Data corruption is a possibility.

So, the final path to find your home directory from windows will look like:

`%userprofile%\AppData\Local\Packages\<Distro-Folder>\LocalStat\rootfs\home\<your-user-name>\`

Installing R on the Linux Subsystem

IMPORTANT: This section is only if you'd like to try using R under Linux. For class, using R under Windows should be fine.

The Linux Subsystem behaves exactly like a regular Linux installation, but for completeness, I will provide instructions here for people new to Linux. These instructions are written from the perspective of Ubuntu, but will be similar for other repos.

R is not a part of the standard Ubuntu installation. So, we have to add the repository manually to our repository list. This is relatively straightforward, and R supports several versions of Ubuntu.

Sources:

- CRAN guide for Ubuntu
- Digital Ocean quick tutorial

1. In a bash window, type:

   ```
   sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619
   ```

   - This adds the key to "sign", or validate, the R repository

2. Then, type:

   ```
   sudo add-apt-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/'
   ```

   - cloud.r-project.org is the default mirror, however, it is prudent to connect to the mirror closest to you geographically. Berkeley has it's own mirror, so the command with the Berkeley mirror would look like

   ```
   sudo add-apt-repository 'deb https://cran.r-project.org/bin/linux/ubuntu/bionic-cran40/'
   ```

3. Finally, type `sudo apt install r-base`, and press `y` to confirm installation

4. To test that it worked, type `R` into the console, and an R session should begin

   - Type `q()` to quit the R session

Installing Rstudio on the Linux Subsystem

⚠ Warning

THIS NO LONGER WORKS

As of Rstudio 1.5.x, it does not run on WSL. Link
Also possible issues, WSL has no GUI, and therefore can't support anything that uses a GUI.
These instructions work, but Rstudio doesn't run.

Sources:

- Rstudio
- Source

1. Go the the Rstudio website (link above) and download the appropriate Rstudio Desktop version.
   - For most people, this is the `Ubuntu 18 (64-bit)` installer.
   - Save it somewhere that you can find it.
   - You should have a file similar to `rstudio-<version number>-amd64.deb`
2. Open a terminal window and navigate to wherever you saved the rstudio install file.
3. Type the command `sudo dpkg -i ./rstudio-<version number>-amd64.deb`
   - This tells the package installer (`dpkg`) to install (`-i`) the file specified (`./thing.deb`)
4. Type the command `sudo apt-get install -f`
   - This tells the package manager (`apt-get`) to fix (`-f`) any dependency issues that may have arisen when installing the package.
5. Type the command `which rstudio` to make sure the system can find it.
   - Output should be similar to `/usr/bin/rstudio`
6. Run rstudio from linux by typing `rstudio &`
   - The `&` runs it in the background, allowing you to close the terminal window.

Installing LaTeX on the Linux Subsystem

! Important

This section is only if you'd like to try using LaTeX under Linux. For class, using LaTeX (or R Markdown) under Windows should be fine.

LaTeX is a text-markup language used when generating documents from .Rmd files.

Source LaTeX

1. Type `sudo apt-get install texlive-full`, press `y` to confirm installation

Generally, if you want to create and edit R Markdown documents you will also need a text editor to go with your LaTeX installation, but we won't go into that here.