

Installing and Using the Windows Subsystem for Linux

Installing the Windows Subsystem for Linux (WSL) on your computer allows you to run Linux command-line utilities directly on Windows. WSL only works on later versions of Windows 10 64-bit (Version 1709 or later). This guide shows how to install and use the Ubuntu WSL, but you can use any Linux distribution in the Microsoft Store. The steps for another distribution are likely different. The Ubuntu terminal will allow you to use command-line tools such as apt, bash, git, and ssh.

Installing WSL:

1. Search “Turn Windows features on or off” in the Windows search bar and open it.
2. Look for “Windows Subsystem for Linux”, make sure the box is checked, and hit OK (You will have to restart your computer).
3. Open Microsoft Store application and search Ubuntu.
4. Click on Ubuntu, hit get, and wait for it to install (You can use any version, the one without a version number will give you the latest version of Ubuntu).
5. Launch the Ubuntu application and allow first time setup.
6. Create a username and password (When entering a password, no characters will appear, but you are typing your password).
7. Update Ubuntu with the command:
`sudo apt update && sudo apt upgrade`
(You will have to enter the password you set)
You should run this command periodically as Ubuntu will not update on its own.

You have successfully installed and setup WSL and can now use the terminal as you would on Ubuntu.

Getting g++:

g++ is the standard C++ compiler. You can install it in Ubuntu by doing the following:

1. Open Ubuntu terminal
2. Install g++ with the command:
`sudo apt install g++`

You have successfully installed g++ and can now use it as you normally would.

Using Makefiles:

In order to use Makefiles, you must install make. If you don’t know what a Makefile is, you can skip this step. You can install make in Ubuntu by doing the following:

1. Open Ubuntu terminal
2. Install make with the command:
`sudo apt install make`

You have successfully installed make and can now use Makefiles.

Accessing Your Files:

Accessing where your files are stored may be a little tricky. To access your files in Windows, you must first change into the directory /mnt, which you can do with the command:

```
cd /mnt
```

You can now use cd to access your files in Windows. If any file or folder contains a space in it, you must put it in quotation marks e.g. "CS Projects".

For example, if your files are stored in your documents folder you can access them by typing the command:

```
cd /mnt/c/Users/*enter your username here*/Documents
```

You want to enter the username set for your account after Users. This is likely different than the username you created for Ubuntu.

If you don't know your username, search cmd in the Windows search bar and open it. Then type the command:

```
echo %username%
```

This should be the username you want to put for *enter your username here*. If your username contains a space, you must put it in quotation marks e.g. "Alan Turing".

Writing Scripts:

Writing a script can make doing a repetitive task quicker to do. For example, a script can make it easy to go to where your files are instead of typing a long change directory command. You can write a simple script with the following steps:

1. Open Ubuntu terminal
2. Open a new script with the command:

```
vi *name of script*
```

Where *name of script* is what you want your new script to be named or the name of an old script you want to edit.
3. Press "i" for insert to write to the file.
4. Type:

```
#!/bin/bash
```

This is how bash scripts must start, to signify that we are using bash.
5. Type what commands you want the script to run.
For example, if you wanted it to go to where your files are stored, you could type:

```
cd /mnt/c/Users/*enter your username here*/Documents
```
6. When finished, press ESC to leave insert mode.
7. Save the file and quit vi by typing:

```
:wq
```
8. Give permission to run the script with the command:

```
chmod +x *name of script*
```

9. Execute your script by typing:

```
./*name of script*
```

If your script contains the command `cd`, you must run it with the following:

```
. /*name of script*
```

Without the extra dot, changing directories will not take effect in your terminal.

Troubleshooting:

If you are having issues with WSL, please consult the document linked below.

Link: <https://docs.microsoft.com/en-us/windows/wsl/troubleshooting>

Questions and Feedback:

If you have questions about or feedback for this guide, please email them to projects AT owenkunhardt DOT com.