***Computer Architecture Course***

**LAB 4 Setup**

**Bash and Make**

**Issue 1.0**

## 

# **Introduction**

In this document, we’ll work through the setup for a couple of helpful command-line tools we’ll need to use in Lab 4: Bash and Make. First released in 1989, Bash stands for **B**ourne **A**gain **SH**ell and is the successor to the popular Unix shell [Bourne](https://en.wikipedia.org/wiki/Bourne_shell), now serving as the default command-line or “shell” environment in most Unix operating systems today (think MacOS and Linux). Both Bourne and Bash support the execution of “shell scripts” which are programs used to automate a mundane or arduous task on the command line. Lab 4 uses a shell script to automate the build process for the lab.

You will need a *make* software to run the *makefiles* for the lab (which are used in the shell script). A *make* software is a build automation tool that reads in a set of directives to generate a target program. These set of directives are declared in a file known as a *Makefile*.

## **Setting up “bash” and “make” for Windows OS**

Lab4\_simple\_pipeline folder includes two makefiles and a bash script (that enables you to automatically compile Arm Education Core, assemble all Assembly test files (testcases), and simulate all of them sequentially. This would help you should you need to do more debugging when implementing the code modifications.

There are several options for you to set up in order to run a bash script or a make command using Windows terminal. In this section, we provide 2 options that you can try. If you do not wish to use the bash script or makefiles, you can still assemble and compile manually, in the same way that was done in the previous labs. If you are using MacOS or Linux, simply use the given terminal in your operating system, which should already be able to execute bash scripts and easily install make.

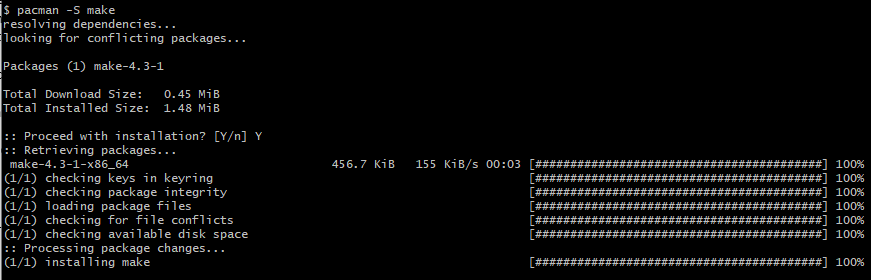
### **Option 1: MSYS2**

MSYS2 is a software based on modern Cygwin and MinGW-64. You can use MSYS2 to install the bash and make software.

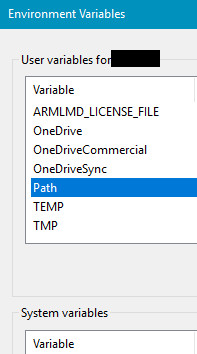
To install MSYS2.exe, follow these steps:

1. Download the MYSY2 installer and follow the “Getting Started” instructions in <https://www.msys2.org/>. We recommend that you install it in C:\, for example, C:\msys2
2. If you haven’t done so, open MSYS2.exe and enter the following command:

pacman -S make



1. Open **Control Panel** in your Windows OS.
2. Select **System and Security** > **System** > **Advanced system settings** > **Environment Variables** > **User Variables** > **Path**

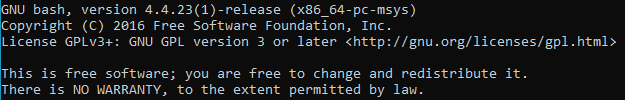


1. Double-click on **Path** in the user variable windows, select **New**, and add the following according to where msys2 is installed:

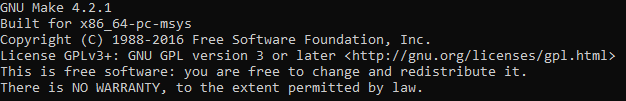
C:\msy2\usr\bin

1. Restart your Windows terminal and enter the following commands:

bash –version



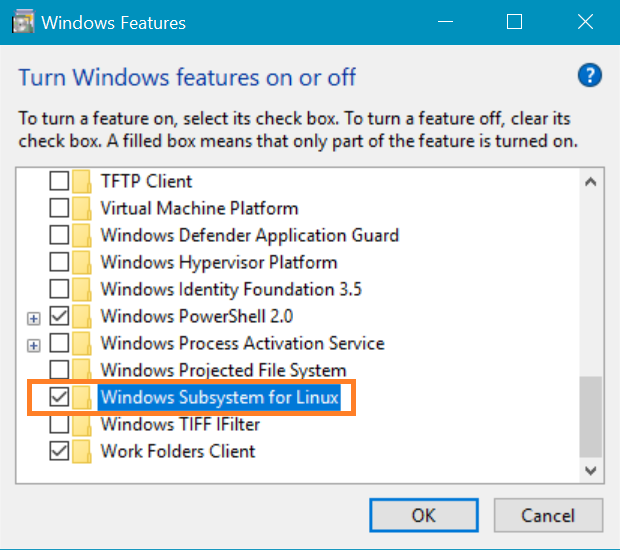
make --version



### **Option 2: Windows Subsystem for Linux + Ubuntu**

To set up Windows Subsystem for Linux, follow these steps:

1. In your Windows OS, open **Settings** > **Update & Security**.
2. Select the **For Developers** tab on the left window pane and select **Developer** mode under developer features.
3. Go to **Control Panel** > **Programs** > **Program and Features**.
4. On the left window pane, click the link **Turn Windows features on or off**.
5. The dialog Windows Features will appear on the screen. Scroll down to the option named **Windows Subsystem for Linux** and enable it as shown below:



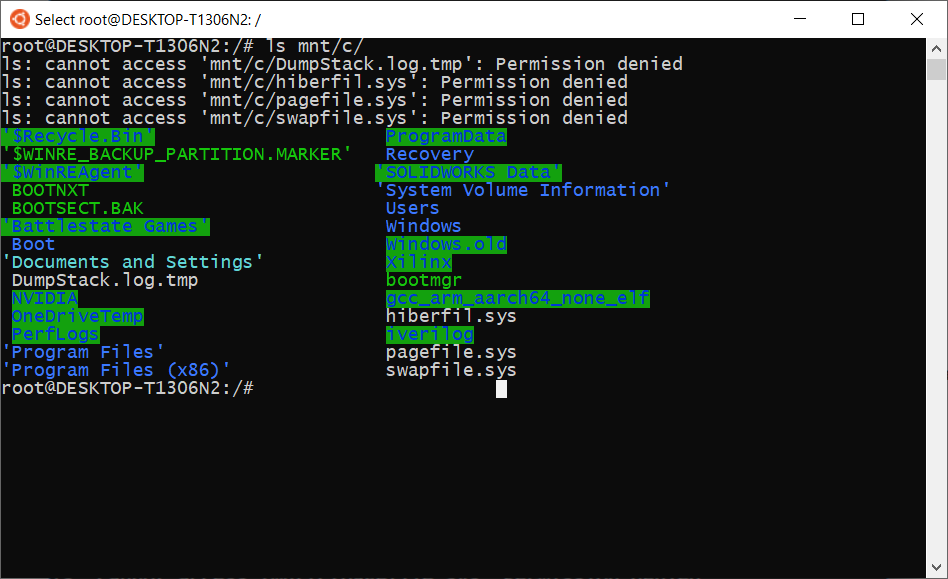
1. Click **OK** to apply the changes you made. Windows will install WSL and reboot the system.
2. After rebooting, open **Microsoft Store** and search for **Ubuntu** and install it.
3. Now open the windows terminal and type bash to go into Linux terminal mode.
4. Type make --version and check if it is installed, and if not, type the following to install it:

sudo apt install make

1. Type exit to end Linux terminal mode. Restart your Windows terminal and proceed to [Task: Running simulation using the provided bash script](https://docs.google.com/document/d/1G6nep7lKxfySOHwNqpaP_urQZxCHqOfZ/edit#heading=h.1ci93xb).

Once you have WSL setup, you can access your Windows files through a WSL terminal by navigating to the /mnt/ directory. For instance, if you want to navigate to the root of the C:\ drive you would execute:

cd /mnt/c/



Say your files for the labs are in your “Documents” folder of your user, you would execute the following command to navigate there, replacing <user> with your Windows username:

cd /mnt/c/Users/<user>/Documents/

## 

## **Setting up “bash” and “make” for Linux**

Since these tools are native to Unix, it’s fairly trivial to install and use them on MacOS and Linux.

**Note**: If you’re on a MacOS version older than Catalina, bash is already installed and is the default shell for MacOS, so going through the install procedure for Bash is not necessary. For Catalina onward, [zsh](https://en.wikipedia.org/wiki/Z_shell) is the default and you can use the instructions below.

### **Linux Steps**

1. Since Linux comes with bash already installed, all you have to do is make sure make is installed:

sudo apt install make