

How to compile your assignment on Windows 10 machine

Hello all,

With the labs closed, we hope you are able to compile your code successfully on your own computer. To help out with Assignment 4, we devised a setup that probably works on your Windows 10 computers.

You will need:

1. Cygwin64 (emulates Linux distribution): <https://cygwin.com/install.html>
2. Linaro (Package for compiling kernel8.img): <https://releases.linaro.org/components/toolchain/binaries/7.3-2018.05/aarch64-elf/> (Download "gcc-linaro-7.3.1-2018.05-i686-mingw32_aarch64-elf.tar.xz", which is 225.8M).
3. 7-zip (Extracts ".xz" and ".tar" files): <https://www.7-zip.org/>
4. QEmu (Emulates operation of Raspberry Pi 3): <https://www.qemu.org/download/#windows>
5. The provided "Makefile", "gpio.h", "snes_uart.h", and "snes_uart.c" in D2L

Steps:

We highly recommend you to test this out using "09_FrameBuffer" example code, as there is a screen to look at, and keys to interact with.

2. Install Cygwin64 using the installer. Make sure to also download the "make" package, since it is not included by default.
3. Extract the Linaro distribution to some location. (e.g. "C:\cygwin64\home\Packages").
 - You will need to extract a compress ".tar" file from ".xz", and then extract the ".tar" file. We recommend using 7-zip since it's free, open-sourced, and a decent file archiver.
4. Install QEmu. You will only require "aarch64" under "System Emulation".
5. Edit the location of your Linaro distribution in your project's makefile, beside the line "INSTALL_DIRECTORY" (e.g. C:\cygwin64\home\Packages\gcc-linaro-7.3.1-2018.05-i686-mingw32_aarch64-elf\bin)
6. Go to your project directory using "Cygwin64 Terminal",
 - Note: Your terminal will open up the location "C:\cygwin64\home\<Profile Name>" by default.
 - Note: You can use your standard UNIX commands to traverse the terminal
7. Transfer "Makefile", "gpio.h", "snes_uart.h", and "snes_uart.c" into your project.
8. Compile and run by using the commands "make" and "make run" in the Cygwin Terminal.

If you have done this correctly, you should see a checkerboard screen. To display UART messages, click on "Cygwin64 Terminal", and then press keys on your keyboard.

Some notes:

- QEmu might crash if you don't have a dedicated graphics card, or if you have a weak computer. From testing, the emulation had a segmentation fault when executed `"initFrameBuffer()"` on Surface Pro 4. It worked fine on a desktop computer.
 - If this is your case, then you can display the state of the maze as string outputs in the terminal using `"uart_puts(...)"`.
- Description of changes:
 - Makefile was edited to use the `".exe"` extension on Windows distribution of Linaro. In addition, the default location for Windows installation was added for QEmu.
 - Because Raspberry Pi 4 is not yet supported on QEmu, we need to switch the MMIO Base Address to the one for Raspberry Pi 3. This is found in `"gpio.h"`.
 - The `"get_SNES()"` function acts similarly to `"07_SNESController"`, including the delay. To use it, `"uart_init()"` has to be called first. The mapping of buttons is described in `"snes_uart.c"`