ECE391: Computer Systems Engineering   **Fall 2021**
Tux Emulator Setup Guide

This document contains the instructions to setup Tux Emulator for MP2 Checkpoint 2, in which you will be learning to write a device driver for a USB device. Tux Emulator is a software emulator of the physical Tux Controller — a USB game controller with buttons and a LED display used to control the game implemented in MP2.

To start, download the Tux Emulator corresponding to your OS from the course assignment page. Please follow the steps for your home OS carefully and do not skip any steps. If you have any questions or difficulties in setting up the emulator, check with a TA in office hours or make a Piazza post. **We highly recommend that you set up the Tux Emulator as early as possible on both your WFH environment and EWS Lab environment. Like for MP1, MP2 demos will also be conducted on the EWS Lab machines through Remote Desktop.**

Windows

The following steps were tested on Windows 10 machines.

1. Download Tux Emulator for Windows.

2. Run the Tux Emulator executable, Windows might warn you that the app is unrecognized in a pop-up window, click "More Info" and then "Run anyway", you should see the Tux Emulator window.

3. Right-click on the test machine Desktop shortcut, go to "Properties", and add the following line to the end of "Target":   **-serial tcp:127.0.0.1:4444** and click OK to save.

   NOTE: it is possible that your "Target" line has reach its character limits and you are not able to input the full tcp address. To solve this, move the "ece391_share" (not the entire "ece391" folder) to a place with a shorter file location name. Update the "Target" on all qemu shortcuts to reflect the new file name (there's usually one place in devel you need to change and two places in test). Reshare your "ece391_share" folder (IMPORTANT: make sure the share name is set to "ece391_share" and not anything else!). If you get an error code of -20 when mounting your work directory, you reshared your "ece391_share" folder wrong.

4. **IGNORE** the serial port setup (Step 3) in the MP2 document, if you have already setup the serial port according to MP2 document, remove **-serial COM#** if it exists in the "Target" line. You should still follow the other steps in setup instruction in MP2 document.

5. Click "OK" to save and close the properties window.

6. Start the test machine, if the test machine does not start up, make sure you have setup the serial port exactly as described in Step 3, and make sure the Tux Emulator is running.

7. If your setup is correct, you should see "OOPS" displayed on the Tux Emulator's LED display after the test machine has booted.

8. Finally, try running `mazegame-demo` by **sudo ./mazegame-demo** in the MP2 folder. The sudo password is "ece391", you should see a timer display on the Tux LED, and you are able to use the Tux Emulator to play the game.

9. It is possible that the buttons stop responding to user input if you close the demo and restart it the second time without restarting your test machine, this is a known issue of the demo only, so it should not affect your development.

Linux (Ubuntu)

These steps are tested on Ubuntu machines, other Linux distros are not tested, but feel free to try.

1. Download Tux Emulator binary for Linux.

2. In the path that the Tux Emulator is downloaded, open a terminal, and run **chmod +x ./<tux controller file-name>**, this makes the binary an executable file.

3. Execute the binary by either double-clicking it or run it from a terminal.

4. **IGNORE** the serial port setup (Step 3) in the MP2 document, if you have already setup the serial port according to MP2 document, remove **-serial COM#** if you found it exists.

5. Use your favorite text editor to open the test machine shortcut on the Desktop, add **-serial "tcp:127.0.0.1:4444"** (including the quotes) to the end of the second line and save the file.

6. Start the test machine, if the test machine does not start up, make sure you have setup the serial port exactly as described in Step 5, and make sure the Tux Emulator is running.

7. If your setup is correct, you should see "OOPS" displayed on the Tux Emulator LED after the test machine has booted.

8. Finally, try running `mazegame-demo` by **sudo ./mazegame-demo** in the MP2 folder, the sudo password is "ece391", you should see a timer display on the TUX LED, and you are able to use the TUX Emulator to play the game.

9. It is possible that the buttons stop responding to user input if you close the demo and restart it the second time without restarting your test machine, this is a known issue of the demo only, so it should not affect your development.

MacOS

These steps were tested on MacOS Catalina.

1. Download Tux Emulator for MacOS.

2. Double click on the emulator to run.

   a. If the OS says that it cannot verify the developer click "Cancel"

   b. Go to System Preference >Security and Privacy >General.

   c. There should be a prompt that asks if you want to open the app anyway.

   d. Click on "Open Anyway".

   e. On the prompt that comes up click "Open".

   f. You should be able to see the emulator pop up now.

3. **IGNORE** the serial port setup (Step 3) in the MP2 document, if you have already setup the serial port according to MP2 document, remove **-serial COM#**.

4. Drag or open the shortcut for test machine as a text file to your favorite text editor. Add **'-serial tcp:127.0.0.1:4444'** to the end of the second line. Make sure to save the file and close the editor.

5. Start the test machine, if the test machine does not start up, make sure you have setup the serial port exactly as described in Step 4, and make sure the Tux Emulator is running.

6. If your setup is correct, you should see "OOPS" displayed on the Tux Emulator LED after the test machine has booted.

7. Finally, try running `mazegame-demo` by **sudo ./mazegame-demo** in the MP2 folder, the sudo password is "ece391", you should see a timer display on the TUX LED, and you are able to use the TUX Emulator to play the game.

8. Note that it is possible that the buttons stop responding to user input if you close the demo and restart it the second time without restarting your test machine, this is a known issue of the demo only, so it should not affect your development.

Acknowledgements