Step 1 - Work Log
M2M Lectures
Grenoble University

Your Names Here

January 19, 2016

1 Preface by Pr. Olivier Gruber

This document is your work log for the first step in the M2M course, master-level, at the University of Grenoble, France. You will have such a document for each step of our course together.

This document has two parts. One part is about diverse sections, each with a bunch of questions that you have to answers. The other part is really a laboratory log, keeping track of what you do, as you do.

The questions provide a guideline for your learning. They are not about getting a good grade if you answer them correctly, they are about giving your pointers on what to learn about.

The goal of the questions is therefore not to be answered in three lines of text and be forgotten about. The questions must be researched and thoroughly understood. Ask questions around you if things are unclear, to your fellow students and to me, your professor.

Writing down the answers to the questions is a tool for helping your learn and remember. Also, it keeps track of what you know, the URLs you visited, the open questions that you are trouble with, etc. The tools you used. It is intended to be a living document, written as you go.

Ultimately, the goal of the document is to be kept for your personal records. If ever you will work on embedded systems, trust me, you will be glad to have a written trace about all this.

REMEMBER: plaggia is a crime that can get you evicted forever from french universities... The solution is simple, write using your own words or quote, giving the source of the quoted text. Also, remember that you do not learn through cut&paste. You also do not learn much by watching somebody else doing.

2 Qemu

1. What is Qemu for?

2. Why cannot you run a linux kernel in a regular linux process?
3. Comment the different options you used to start qemu.

3 Boot Process

1. How is an x86 machine booting up?
2. Hints: Bios, MBR (Master Boot Record), Kernel.
3. What is the role of each involved parts?
4. How is built the disk image that you use to boot with qemu? Describe its layout in terms of sectors and the contents of those sectors.

4 Using Eclipse to browse the sources

Explain how you configured Eclipse to be able to browse the given sources.

5 Master Boot Record

1. From what sources (.c and .S files) is the MBR built?
2. What is the purpose of those different files?
3. What is an ELF? (Hint: man elf, Google is your friend)
4. Why is the objcopy program used? (Hint: look in the Makefile)
5. What kind of information is available in an ELF file?
6. Give the ELF layout of the MBR files (hint: readelf and objdump)
7. Look at the code in loader.c and understand it.
8. What are the function waitdisk, readsect, and readseg doing?
9. Explain the dialog with the disk controller. (Hint: in/out functions).
10. What can you say about the concepts at the software-hardware frontier?

6 Master Boot Record Debugging

Use gdb to step through our bootloader.

Hints:

1. Look at the dbg target in the Makefile.
2. Look at the .gdbinit file.

List and explain the various gdb commands you use.

7 Our mini Kernel

1. What is the code in crt0.S doing?
2. What are the function in/out for at this level?
3. What are the inline attributes for?
4. Explain why is your fan ramping up when you launch qemu with:
   $ make clean ; make run
5. Explain what is the relationship between the qemu option (-serial stdio) and the COM1 concept in the program.
6. Explain what is COM1 versus the console?

8 Debugging with Eclipse

How did you setup your Eclipse to debug your mini kernel?

9 Kernel Extensions

IT IS MANDATORY TO USE THE DEBUGGER TO DEBUG YOUR CODE.

9.1 Echo on the screen

This extension is to have the input from the UART be echoed on the console screen (the greenish output). Do not forget that you have only 25 lines and you will need to implement scrolling.

9.2 History and line editing

This extension is to have a history of typed lines. A line is added to the history when the return key is pressed. The arrow up and down allow you to scroll up and down in the history. The arrow left and write allow you to move left and right in the current line. The backspace and delete allow you delete characters.

9.3 Echo on COM2

This extension is to have the ability to have a printf-like capability on COM2. The code is in the kprintf.c file.
Hints:

1. Look at the target run2 in the Makefile to know how to setup COM2.
2. Add the kprintf.c file to your kernel
3. Launch with "make run2" and use a telnet connection for COM2.

10 Laboratory Log