CS452 F23 Lecture Notes

Lecture 01 - 07 Sep 2023

1. Course Overview

- **real time** systems
  - emphasis is on experience with software that interacts with the real world
  - data acquisition, model/control physical world, communication
- real time **systems**
  - small operating system, plus train control application
  - start from scratch, get a better understanding of things you may take for granted
    - booting
    - device interaction
    - memory
    - timing
    - threading and concurrency
- few guard rails
- room to roam
  - design flexibility in assignments
  - you define and implement a project
- this term
  - new serial hats
  - new boot proceedure

2. Course Workload

- Assignments
  - A0: getting started
  - K1-K4: build your kernel
  - TC1-TC2: train control application
- Demos, and kernel design presentation
- Final Project
- no midterm, 24 hour take-home final exam
- marking
  - assignments: 60%
  - project: 10%
  - final: 30%
- Piazza

3. Infrastructure

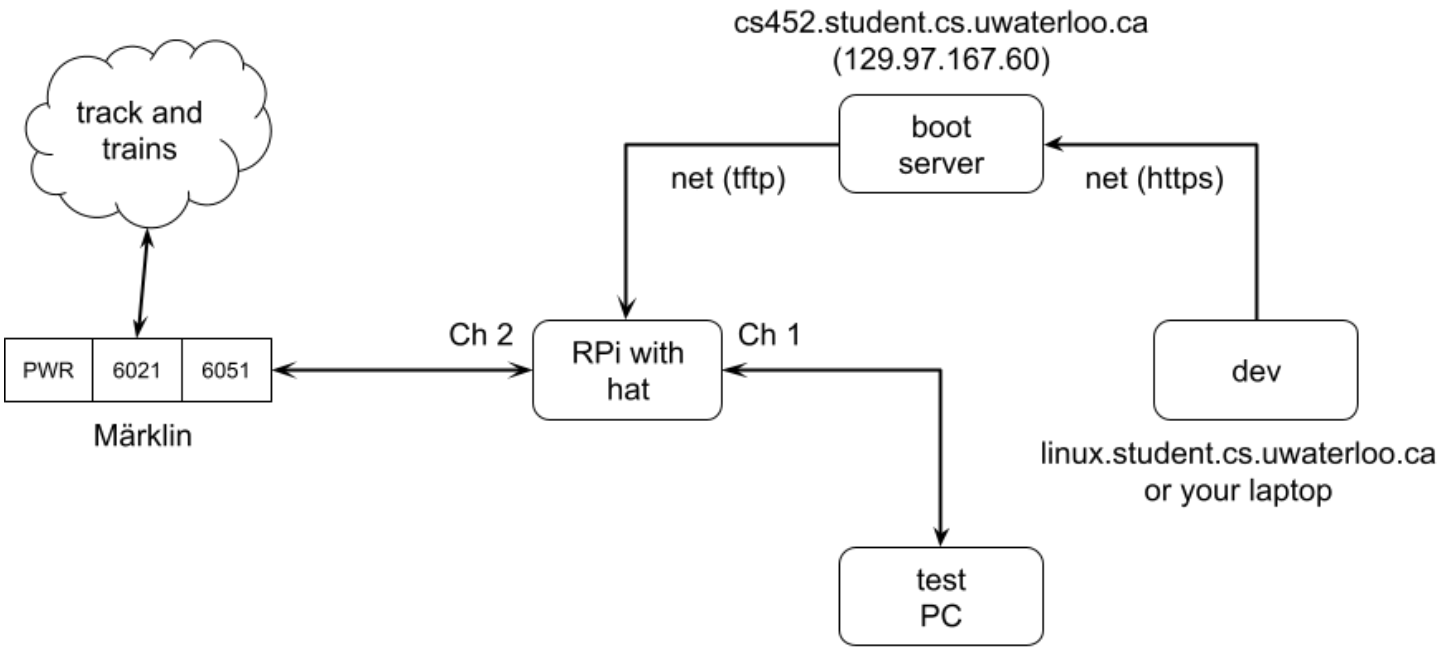

Figure 1: Overview of CS452 Dev/Test Infrastructure

4. Development Overview

- basic procedure 1 build and create .img file on dev box 2 upload .img to boot server (cs452.student.cs.uwaterloo.ca) 3 power on RPi, wait for u-boot prompt (output to gtkterm on test PC) 4 use u-boot to load .img into RPi memory and run it

5. Sample Code

- iotest
  - very simple, does console input/output using channel 1
  - use it to try the dev/test process
- development tools
  - cross-compiler and other usual tools (linker, readelf, etc)
  - if dev on linux.student.cs.uwaterloo.ca, tools are pre-installed in `/u/cs452/public/xdev/bin`
  - if dev on your laptop, course webpage has instructions for installing a cross-compiler and tools there

6. More On Development

- bare-bones, freestanding code
  - no operating system, no libraries (including libc)
- ELF/image files
  - at runtime, program will have code, data, stack (no heap)
  - build process creates ELF file
    - text, data, bss sections
    - start of the text section is the entry point
  - img file is produced from ELF - includes text and data (not bss)
    - bss variables need to be initialized by the program
- libc issues
  - can't rely on underlying OS to provide services, like I/O, memory allocation
  - SIMD instructions rely on MMU, which is disabled

    - options:
        - don't use libc (like iotest)
        - use only non-problematic bits of libc (see check in iotest Makefile)
- keep things simple by sticking with C

7. Assignment 0

- due at start of class on Tuesday 19th (8:30am)
    - bring printed hard copy to class
- start now

- polling loop, checking track state, clock, user input

```
for (;;) {
  if (c1) f1();
  if (c2) f2();
  ...
}
```

Author: Ken Salem
Created: 2023-09-07 Thu 11:00