Resources for reading pi-OS

Pi-OS contains many configuration processes and low-level interaction with CPU. It's very useful to understand the existing codes in pi-OS. Here are some resources for you to read codes in pi-OS.

(Download pi-OS from: https://github.com/fakaiwang/Pi-OS-Design)

1.Basic C++ concepts

Currently, pi-OS is implemented in C++, you need to know some basic concepts about C++.

C++ is object-orientated language, while C is procedure-oriented language. Object-oriented programming is an essential skill for software engineer.

Only a small portion of C++ features are used in pi-OS. So you don't need to know all the stuff of C++(This may need several years' experience). In future experiments, you can implement your design either in C++ or C.

Some C++ concepts needed for reading pi-OS:

You need to understand what is a class, object, class members

You need to understand difference between private members and public members.

You need to understand class constructor and class destructor

You need to understand inheritance and Polymorphism

Some tutorials to these concepts:

http://www.cplusplus.com/doc/tutorial/classes/

http://www.cplusplus.com/doc/tutorial/templates/

http://www.cplusplus.com/doc/tutorial/inheritance/

http://www.cplusplus.com/doc/tutorial/polymorphism/

http://www.cplusplus.com/files/tutorial.pdf

2. Basic Operating System concepts

Since this is the beginning of semester, some concepts are still not covered in lecture. But you can always find introduction to many OS concepts in Google. You need to get some basic impression for OS concepts, like memory system, ARM MMU, paging, interruption, timer. You need to know how interruption works.

Resources:

you can find these concepts in Google, Wikipedia or your textbook

All these concepts need to be built on ARM, so you will also find implementations in ARM documents, such as cortex_a7_mpcore_r0p5_trm.pdf. (ARM architecture reference manual is more detailed, and for reference when you want to check a particular topic).

3. ARMv7 instructions and BCM chip features

For our projects, we also need to know ARMv7 instructions and BCM283* features, since we're on the BCM2836(or BCM2837) platform. Instructions include general operation instructions and system setting instructions. Features include GPU, mailbox, interruption system, timer.

Resources:

cortex_a7_mpcore_r0p5_trm.pdf and ARM architecture reference manual.

Existing codes in pi-OS.

BCM2835-ARM-Peripherals.pdf

4. IDE (Integrated Developing Environment)

You'd better read and write codes in an IDE, which is more efficient than Vim or text editor.

Get Eclipse from: (for C and C++ projects)

http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/keplersr2

A source for arm-none-eabi- tool chain

I find pre-built arm-none-eabi- tool chains:

https://launchpad.net/gcc-arm-embedded/+download

arm-none-eabi- can generate ARM32 objects.

2.

https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads