ECE472/572 Embedded Microcontroller Linux – 3CR

Instructor: Dr. Aleksander Malinowski http://olek.matthewm.com.pl/

Course description

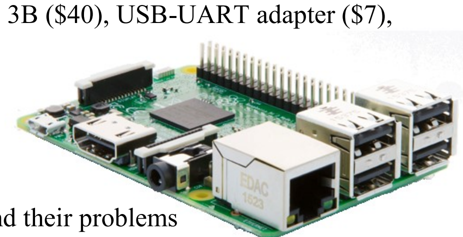
Understanding of Linux and its adoption as an embedded OS platform including process and thread management, communication, synchronization, and deadlocks, also virtual memory and file systems; overview of methods and techniques to design and create embedded systems based on the Linux kernel. The essentials of the Linux operating system are discussed from the embedded system point of view including selecting, configuring, cross-compiling, installing a target-specific kernel, drivers and subsystems; the GNU development toolchain; and tools used to build embedded Linux systems.

Prerequisites by topics

Proficiency in computer programming in C or C++; Concepts of the data structures; some experience in programming a microprocessor system in assembly or C with use of interrupts; ECE103 and ECE205 satisfy these requirements.

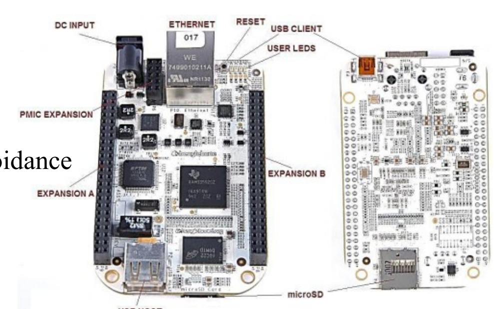
Textbooks and other required material

1. *Operating System Concepts Essentials, 2ed*, by Silberschatz, Galvin, Gagne/978-1118804926/© 2013 Wiley (required)
2. *Linux Pocket Guide, 2ed* by Barrett/978-1449316693/© 2012 Oreilly (required as preapproved notes for the tests)
3. Building Embedded Linux Systems by Yaghmour, Masters, and others/978-0596529680/© 2008 Oreilly (recommended)
4. Students are required to use their own laptops with at least 20GB of dedicated hard drive space and 2-4GB of RAM (Hard drives close to full will make some labs takes a longer time to complete. Lab 6 may even take overnight to do!)
5. Students must purchase two 4GB micro SD cards with regular card adapter, and a USB-based SD card reader ($5)
6. Students are required to purchase their own BeagleBone Black ($65) or RaspberryPi 3B ($40), USB-UART adapter ($7), and Ethernet patch cable ($5), and a



Course Objectives

1. Understanding operating system structures
2. Understanding process and thread scheduling and management
3. Understanding inter-process and inter-thread communication and synchronization and their problems
4. Writing simple applications that utilize multithreading and communication between [redacted]
5. Ability to set up, configure and maintain a Debian type Linux distribution
6. Ability to set up a cross-compiler toolchain for a microcontroller on a Linux platform
7. Ability to set up and customize a microcontroller Linux (ucLinux) development environment, and set up ucLinux on the ARM microcontroller/DSP target board
8. Ability to modify or write simple device drivers that utilize specific hardware, and to write simple programs that interact with such drivers
9. Ability to set up a virtual appliance and understand the differences and limitation between running an OS on a host computer or on a virtual appliance



Topics Covered

1. Operating System structures
2. Processes and Threads: scheduling, control, and data sharing
3. Processes and Threads: synchronization, deadlocks, deadlock detection and avoidance
4. Memory paging and virtual memory
5. File systems, mass storage structures, file access, booting an operating system
6. Distributed systems (if time permits)
7. Linux: set up, configuration, maintenance
8. Makefiles, compilers, and cross-compilers
9. Configuring and building a kernel (use a toolchain to cross-compile ucLinux for a microcontroller)
10. ucLinux device drivers that interact with hardware, communication with such drivers from user processes, ucLinux services, and scheduling with CRON table
11. Virtual machines as tools to run multiple operating systems on one workstation

Note: Linux and microcontroller Linux are introduced in parallel with the theory of operating systems.
Lectures are alternated: OS theory and an introduction to one open lab each week.