ECET 261 Embedded Networking & Operating Systems

Hours: 2/3/0

Prerequisites: ECET 230 Object-Oriented Programming in Hardware, ECET 260 ARM Microcontrollers and the Internet of Things

Short description:
Students will be introduced to the Linux operating system for the Raspberry Pi. Students will compile the kernel and file systems from source and use them to control a robot. They will also cover multi-tasking and real-time operating systems (RTOS), as well as networking and web services.

Learning outcomes:
Upon successful completion of this course a student will be able to:

- differentiate between a real time operating system and a multi-tasking operating system;
- explain methods of inter-process communications;
- define deadlock and mutual exclusion;
- describe the effects of latency and delay;
- describe the difference between interrupts and polling;
- design interface software to computer peripheral hardware;
- modify and compile a current multi-tasking operating system from kernel sources using Linux;
- configure file systems for embedded control systems using Linux;
- design applications using a Raspberry Pi single-board computer;
- explain the layers of the TCP/IP stack and their implementation in devices;
- describe and use IPv4 and IPv6 protocols;
- design web services for embedded applications.

Course outline:

1. Introduction **1 hour**

1.1 Operating systems
- 1.1.1 Embedded operating systems
- 1.1.2 Unix like systems
  - 1.1.2.1 Linux
  - 1.1.2.2 BSD
  - 1.1.2.3 Solaris
  - 1.1.2.4 OSX
- 1.1.3 Windows
- 1.1.4 Other embedded OS es

1.2 Fundamental services

2. Multitasking **4 hours**

- 2.1 Cooperative multitasking
- 2.2 Pre-emptive multitasking
- 2.3 Real-time operating systems
- 2.2 Process management and services
- 2.3 Concurrent processes
- 2.4 Cooperating processes
- 2.5 Inter-process communications
- 2.6 Inter-processor communications
- 2.7 Process synchronization

3. System design considerations **4 hours**

- 3.1 Mutual exclusion
- 3.2 Deadlock
- 3.3 Starvation
- 3.4 Latency and delay
- 3.5 User interface design

4. Resource Sharing **3 hours**

- 4.1 Interrupt handling
- 4.2 Polling
- 4.3 Memory and device sharing
- 4.4 Buffering techniques

5. Embedded Linux **3 hours**

- 5.1 Linux development environment
- 5.2 Compiling the kernel
- 5.3 Building a root file system
- 5.4 Linux boot disk

6. Raspberry Pi and Linux on ARM **4 hours**

- 6.1 Building an image for an embedded ARM device
- 6.2 Configuring Raspberry Pi WiFi
- 6.3 Remote login secure shell (SSH)
  - 6.3.1 Command line
  - 6.3.2 Graphical
- 6.4 Robots with ARM
  - 6.4.1 Review of USB[1]
  - 6.4.2 Interfacing with FTDI USB module
  - 6.4.2 Interfacing with STM32F4Discovery board

7. Networking **5 hours**

- 7.1 Full TCP/IP stack[2]
  - 7.1.1 Masking
  - 7.1.2 Devices
    - 7.1.2.1 Hubs
    - 7.1.2.2 Switches
    - 7.1.2.3 Routers

7.1.2.4 Nodes

7.2 IPv4
- 7.2.1 Classes of network
- 7.2.2 Classless addressing

7.3 IPv6

7.4 Web services
- 7.4.1 Dynamic HTML
- 7.4.2 Javascript
- 7.4.3 Cascading style sheets (CSS)

Tests and review	4 hours
Total	**28 hours**

Notes:

[1] USB architecture, protocols and classes could be covered in ECET 260 ARM Microcontrollers and the Internet of Things.

[2] An introduction to the TCP/IP stack was provided in ECET 260 ARM Microcontrollers and the Internet of Things.

Labs:

1. Raspberry Pi
2. Boot loaders
3. Operating system introduction (Linux)
4. Processes /Threads
5. Signals / Alarms
6. Controlling Hardware (root)
7. Controlling Hardware (modules)
8. Linux / Embedded Linux
9. Linux on Arm
10. USB Interfacing
11. Robot
12. Cloud Services / Servers

Textbooks: None
Reference: Handouts as required. Online references as required.

Evaluation:

Lecture:

Term Tests (2-3) 60
Total Lecture Marks 60

Lab:

Total Lab Marks 40
Total 100

A minimum grade of "C" is required in order to continue on to courses for which this is a pre-requisite. To obtain at least a C grade:

• The student must obtain a composite mark of at least 60% with a minimum of 60% in theory and 60% in labs.

• ALL lab exercises must be completed to a "satisfactory" level. Failure to do so in a timely* manner will result in a grade of "F" for the course. (*Absolute deadline for completion of all lab exercises will be announced.)