

1. Course number and name

CptS 460: Operating Systems and Computer Architecture

2. Credits and contact hours

3 credits, 3 lecture hours

3. Instructor's or course coordinator's name

K.C. Wang

4. Textbook, title, author, and year

K.C. Wang. 2018. *Embedded and real-time operating systems*. Springer International AG.

K.C. Wang. 2015. *Design and implementation of the MTX operating system*, Springer International AG.

Other supplemental materials

Class website: <<http://www.eecs.wsu.edu/~cs460>>.

5. Specific course information

- a. *Catalog description*: Operating systems, computer architectures, and their interrelationships in micro, mini and large computer systems.
- b. *Prerequisites or corequisites*: CptS 360 with C or better. Certified Major in Computer Science, Computer Engineering, or Electrical Engineering.

6. Specific goals for the course

By the end of the course, students will be able to

- Design boot programs (1,2,6)
- Apply concepts and techniques of multitasking and context switch (1,2,6)
- Implement process scheduling (1,2,6)
- Implement operating system kernel, kernel and user mode operations, interrupt processing and device drivers (1,2,6,7)
- Implement a file system and user interface to an operating system (1,2,7)

7. Brief list of topics to be covered

- Introduction to Operating Systems: Unix/Linux, MTX. computer system and operations, system development software, PC emulators, link C and assembly programs.
- Booting: Develop booters for Linux and MTX.
- Processes: Concept and implementation of processes; process states, context switching, process scheduling.
- Process management in Unix/Linux: fork, wait, exit, exec, signals, pipes.
- Process Synchronization: The process model; mutual exclusion and critical regions, Implementation of low-level mutual exclusion primitives, sleep/wakeup, semaphores, deadlock and deadlock handling.

- Process Communication: High-level process synchronization constructs; messages.
- Process Control: Scheduling algorithms.
- Memory Management: Memory management schemes, virtual address and physical address, memory management hardware in ARM, 1-level and 2-level paging, static and dynamic paging, memory exception and handling, demand-paging and virtual memory.
- I/O and device drivers: Display driver, interrupts and interrupt processing, synchronization between interrupt handler and process, interrupt-driven driver design, keyboard driver, timer and timer services, UART drivers, Digital Security Card driver and booting from SDC.
- File Systems: Review of EXT2 file system, file operations in kernel.
- User interface: Design and implementation of system startup programs; init, login and command interpreter sh.