Synopsis:

This course explores fundamental capabilities of Intel's Pentium microprocessor, regarded as a "bare machine" for which the basic software components of an operating system are to be constructed in the context of standard PC-BIOS firmware and peripheral devices.

Topics appropriate to this goal include:

- crafting a bootstrap loader
- identifying the processor model and stepping
- segmented real-mode memory-addressing
- device detection and initialization
- the CMOS non-volatile memory and real-time clock
- the keyboard, display monitor, and interval-timers
- the programmable interrupt controllers
- the floating-point, mmx, and simd instructions
- building protected-mode segment-descriptors
- entering and leaving 16-bit or 32-bit protected-mode
- creating exception-handlers and interrupt-descriptors
- building page-tables for virtual memory support
- processor mechanisms for multitasking and debugging
- emulation of the legacy 8086 execution environment
- initialization and communication among multiple CPUs
- system management mode and performance monitoring counters

Throughout the course we will write small software-component prototypes (to demonstrate various processor capabilities being studied) using the GNU/Linux program development tools (including editors, assemblers and C/C++ compilers).

The course is open to USF computer science graduate students (and to qualified undergraduates with the Instructor's permission). Prior experience with the C/C++ programming language and the UNIX/Linux operating system is assumed, plus an acquaintance with the Intel Pentium processor's registers, instruction-set, and assembly language.

Inside every large program, there is a small program struggling to get out.

-- Sir Tony Hoare, Emeritus Professor of Computing, Oxford University

Learning Outcomes:

- You will broaden your knowledge of standard 32-bit Intel Architectures
- You will become aware of the Intel processors' seldom-used capabilities
- You will increase your proficiency with using C and assembly language
- You will know how to access service-functions provided in BIOS firmware
- You will gain practical experience in programming peripheral I/O devices
- You will acquire background for understanding next-generation CPUs
- You will lay a foundation for pursuing some additional career options

Instructor:

Dr. Allan B. Cruse, Professor of Computer Science and Mathematics
Harney Science Center - Room H-212 Telephone: (415) 422-6562
Office Hours: Mon-Wed 2:30-4:00pm Email: cruse@usfca.edu
Website: http://nexus.cs.usfca.edu/~cruse/

Textbook:

Tom Shanley, ***Protected Mode Software Architecture,***
Addison-Wesley Publishing Company (1996), ISBN 0-201-55447-X

Classroom Facilitiy:

The course meets on Mondays and Wednesdays, 5:15-7:00pm, in the Michael D. Kudlick Interactive Computer Classroom (HRN-235). Students will need to have individual computer accounts set up for access during these class meetings.

Exam Dates:

Midterm Exam I will be Monday, March 1.
Midterm Exam II will be Wednesday, April 7.
Final Exam will be Monday, May 17, 6pm-8pm

Grading scheme:

Class Participation	20%
Programming Projects	30%
Midterm Exams (2)	30%
Final Examination	20%

NOTE: Unprofessional conduct, such as an abuse of USF computer privileges (unauthorized access), or a violation of academic integrity (plagiarism or fraud), will result in the student receiving a failing grade.