0203-635-01 Advanced Systems Programming (4-units) Spring 2003

Synopsis:

Nowadays it is well recognized that the master control program for a computer system should be designed from the outset to provide for 'extensibility': that is, in order to avoid early obsolescence, the system software must be able to 'evolve' during its lifetime, not merely to allow for the repair of 'bugs', but to accommodate new hardware equipment as it is introduced and to support new performance requirements that may not even have been foreseen at the time the system software was being designed.

Two mechanisms for 'extensibility' are provided by the Linux operating system. First, its source code is made freely and publicly available, so that this code can be altered by a knowledgeable systems programmer, then recompiled and reinstalled, and the resulting system rebooted, to provide any desired upgrades or bug fixes that an existing system might need. Second, it is possible for programmers to create entirely new pieces of code (called 'installable kernel modules') that can be independently compiled and then added into a system while it is running without modifying any of the original code or shutting down the machine for a system reboot. Moreover, it is just as easy to remove these new 'modules' and thus have the system revert to its former behavior.

Both of these Linux extensibility mechanisms offer invaluable opportunities for computer science students to study in detail the inner workings of a 'live' computer system. Accordingly this course is all about Linux kernel modules! It:

- focuses on the Linux operating system for Intel Pentium processors

- assumes familiarity with the C/C++ programming language, with 80x86 machine architecture, Unix operating system commands, and standard data-structures and algorithms (e.g., CS 112, 210, 245, 326)

- It is open to USF Graduate Students (and to qualified undergraduates with the instructor's permission)

Topics covered include: Shared runtime libraries; Loadable kernel modules; Asynchronous input/output; The '/proc' and 'ext2' file systems; Character and Block Device-Drivers; System data-structures and algorithms; Kernel threads, timers, and wait-queues; Interrupts, Exceptions, and System-Calls; and IA32 Symmetric Multiprocessing architecture.

The course consists of lectures, readings, discussions, demonstrations, and programming exercises, some conducted as in-class experiments, and others completed as independent projects outside class.

Learning Outcomes:

- You will know how to read and write code for an operating system
- You will know how to build customized extensions to an OS kernel
- You will be able to invent new tools which let users control their PCs
- You will be able to assess system features that impact performance

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/

Textbooks:

A. Rubini and J. Corbet, *Linux Devixe Drivers (Second Edition),*
(O'Reilly & Associates, Incorporated, 2001), ISBN 0-596-00008-1

Michael Beck et al, *Linux Kernel Programming (Third Edition),*
(Addison-Wesley Publishing Company, 2002) ISBN 0-201-71975-4

Classroom Facilitiy:

The course is scheduled to meet Tuesdays and Thursdays, 3:15-5:00pm, in the new Kudlick Computer Classroom (HRN-235). Students will need to have individual computer accounts set up for access during classes.

Exam Dates:

Midterm Exam I will be Thursday, March 6.
Midterm Exam II will be Tuesday, April 15.
Final Exam will be Thursday, May 22 (12:30pm)

Grading scheme:

Programming Projects (5)	50 %
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 (plagarism), will result in the student receiving a failing grade.