Fall 2008

CEG 433/633: Operating Systems

Prabhaker Mateti
Wright State University - Main Campus, prabhaker.mateti@wright.edu

CEG 433/633:Operating Systems

Prabhaker Mateti

Catalog Description: Overview of operating systems internals. File-system usage and design, process usage and control, virtual memory, multi user systems, access control. Course projects use C++ language. 4 Credit Hours. Three hours lecture, two hours lab.

Prerequisites: CEG 320 and CS 400

Source Material

SG

Silberschatz & Galvin, Operating Systems, 7th Edition with Windows XP, 2004: John Wiley & Sons, Inc. (Some previous editions are by Addison-Wesley.) http://www.cs.yale.edu/homes/avi/os-book/os7/index.html

RS

W. Richard Stevens, Advanced Programming in the Unix Environment, Addison-Wesley, ISBN: 0-201-56317-7. http://www.kohala.com/start/apue.html

Home Page

www.cs.wright.edu/people/faculty/pmateti/Courses/433/

News Group

wright.ceg.433 Post all your questions, helpful comments, criticisms, and suggestions regarding this course (lectures, projects, home work, exams) to our news group. I am hoping for a lively discussion leading to good answers and clarifications. Keep an eye on this newsgroup.

1. Course Content

The numbers in parentheses are a rough estimate of the number of (75-minute) lectures on each topic. Project work is a significant part of this course. The ordering of lectures, in contrast to the course content topics listed below, is largely due to this influence.

Systems Software (3)

Components of a Computer System: Hardware, Systems Software and Applications.; Systems Software: Operating Systems, Monitors, bootstrap and regular loaders. Compilers, assemblers, macro-processors, and linkers.; Operating Systems: Interrupt handlers, device

drivers, file systems, networking, memory management, multi-tasking, and resource control.; Hardware Characteristics Relevant to Operating Systems: Instruction set. IO hardware. Interrupts. Kernel and User modes.; The sequence of events from initial power-on cold booting to shut down of a computer system. The `ps` and related commands. The standard Unix processes: `init`, `getty`, etc.; Chapters 1, 2, and 3 of SG. Chapter 2 of RS.

Programming Support (2)

Project work is heavily dependent on the material of this section. Lectures on this topic are spread over the 10 weeks. *Using Unix:* Man pages. Makefiles. Debugging. Executable binaries v. `bash` shell programming. IO redirection. Filters and pipes. Signal handling. Exception handling via `setjmp` and `longjmp`.; Systems implementation languages v. problem oriented languages, and command/script languages.; *C and C++ Languages:* Expressions and statements. Assignment statement as an expression. Block structure. Type compatibility. Pointers: `void *, T *`, pointers to functions. Prototypes. The Preprocessor directives. Portability.; *Libraries:* Standard libraries v. operating system calls. Unix system calls, and the standard. Shared libraries. Routines v. packages. Memory allocation, file io. Chapters 1 - 4, 7, 10 of RS.

Memory Management (3)

Dynamic storage allocation and liberation. The `malloc+free, new+delete` of C/C++ languages, and the `sbrk` of Unix. Garbage collection.; Virtual memory. Address spaces. Swapping. Secondary storage. Page faults, Memory management units, and other architectural support for paging, and segmentation. Demand paging. Page replacement algorithms.; Chapters 8 and 9 of SG. Chapter 7 of RS.

File Systems (4)

The semantics of file `open, read/write, close` and `unlink`. Naming and allocation of resources. Hierarchical directories. Security and protection mechanisms. Keyboard, display, and mouse viewed as files.; Unix file system. The structure of i-nodes. Buffer cache. The mechanism of mount.; Chapters 10, 11, and 21 of SG; Chapters 3 and 4 of RS.

Processes and Threads (3)

Programs v. processes. Loading of programs into run-time images of processes. Creation and control of processes. Chapters 3 and 4 of SG; Chapters 7 - 10 of RS.

Device Drivers (2)

IO subsystem and Device Drivers: DMA; Interrupt handlers; driver interfaces; overview of drivers for: disk, clock, terminal. Disk scheduling, and management. Using disks and slow RAM for swap space. Chapters 12 and 13 of SG.

Case Studies (2)

Unix and Windows. Discussion of Unix is spread across the entire course.; Chapters 21, 22, 23 of SG. Chapter 2 of RS.

Grading

Exams

There will be two exams contributing 30 points and 35 points to the final grade. The mid term is scheduled around the sixth week, and the final during the exam week as set by the Registrar.

Projects

The projects contribute 35 points to the final grade. I *expect* to give the project in five parts worth 5+5+10+10+5 points respectively. The due dates for these will be announced in class.; Subsequent parts build on the code you write. So, make sure your source code *quality* is high. Some of the above have a cascading effect. The projects will be evaluated based on three criteria: (1) approach, clarity, and elegance, (2) correctness, and (3) efficiency. These projects must be work done *solely by you*, except for the parts I provided you with. The implementation must be in `C or C++` demonstrable on our Linux systems. Projects must be submitted on-line using the `turnin` program.; All work for this course is expected to be done in the OSIS Lab (Operating Systems and Internet Security Lab, 429 RC) using Linux. Our grading of your submitted project work includes the execution by us on *these* machines. *I may ask you for a demo of your projects. During or after the demo, I may also ask you questions pertaining to your projects.* Bonus points are awarded in recognition of good work, in addition to the max possible points. Quality is subjectively judged. Merely turning in a file will not receive full score. Bonus points are used not in "curving" but in possibly pushing an individual student's grade up.

Homework Assignments

I will recommend that you work on various problems from the book and other places. However, as this course has currently no TA support, I will neither grade the solutions nor provide solutions.

Discussion Group Activity

Post all your questions, helpful comments, criticisms, and suggestions regarding this course (lectures, projects, home work, exams) to our **discussion**group. I am hoping for a lively discussion leading to good answers and clarifications. Keep an eye on this group. I am not looking for mere volume of submitted articles, or just questions, but quality of answers and discussion you provide in the group.

CEG 633

Students enrolled in CEG 633 are required to do additional reading(s) on operating systems and write a brief (about 5 to 10 pages) report. This quarter the task is to search the Web and write a *technical* summary that demonstrates your understanding on the topic outlined below.

> How should a file be deleted ("shredded") so that it is as difficult as possible to recover it? Survey to what extent this can be done in NTFS5 (Windows) and Ext3 (Linux).