Brooklyn College
Department of Computer & Information Sciences

CISC 7100 [701] Foundations of System Programming
37½ hours plus conference and independent work; 3 credits

Programming with the basic resources of the operating system. Process, threads, and the inter-process/thread communication facilities, signals, pipes, sockets, semaphores, and shared memory. Allocation and protection of resources. Process and thread scheduling. Network programming.

Objectives

Parallel and distributed computing is now ubiquitous in both business and scientific computing. The recent introduction of Multiple Core technology, the cost effective construction of Cluster and Grid supercomputers, and the increasing importance of client/server distributed enterprise level applications all point to a new focus on parallel computing. In particular, multiple cores brings the problem of effective use of parallelism into the computing mainstream as never before. Before our students can address the changes necessary to program in multiprocessor and network environments, they need a firm background in the issues of system programming on single processors. In addition, they need a thorough introduction to thread programming which is not often covered in standard courses on operating systems.

- To provide students with an understanding of the life cycle of program processes – creation, communication and termination.
- To provide students with a knowledge of the organization of the file system and its interaction with processes .
- To provide students with an understanding of interprocess communication and communication between the operating system and a process.
- To provide students with knowledge of thread programming – synchronization and communication.
- To provide students with an introduction to network programming via sockets.

Syllabus

Week 1. Overview of operating systems with an emphasis on the philosophy of the Linux and Unix operating systems. Introduction to the shell scripting language.

Week 2. Introduce the concept of a process, discuss with respect to Linux/Unix operating system.

Week 3. Discuss the creation of processes. Introduces the fork() system call.

Week 4. The process environment which includes a discussion of the file system and resource allocation.

Week 5. Using processes with more on fork(). Introduce the exec(), exit() and wait() system calls.

Week 6. How processes communicate with the operating system – signals.

Week 7. How processes communicate with each other on a single processor – pipes.

Week 8. How processes communicate with each other on a single processor – semaphores and shared memory.

Week 9. Exam

Week 10. Introduction to thread programming on a single processor.

Week 11. Thread programming and synchronization – mutex locks, semaphores and condition variables.

Week 12. Thread programming on a multiprocessor parallel computer.

Week 13. Introduction to sockets and remote communication.

Weeks 14. Client/server programming and synchronization with sockets.

Bibliography

"Design of the UNIX Operating System," Maurice J. Bach, Pearson Education. 1987.

"Interprocess Communication in Linux," Alfred Gray, Pearson Education, 2002.

"Interprocess Communications in UNIX: The Nooks and Crannies, 2nd Edition" John Shapley Gray, Pearson Education, 1998.

"UNIX System Programming, 2nd edition" Keith Haviland, Ben Salama, and Dina Gray, Addison Wesley, 1999.

"Understanding UNIX/LINUX Programming: A Guide to Theory and Practice," Bruce Molay, Prentice Hall, 2002.

"Advanced Programming UNIX Environment, 2nd edition" W. Richard Stevens, Richard Stevens, and Stephen A. Rago, Addison-Wesley, 2005.

"UNIX Network Programming: The Sockets Networking API, Vol. 1, 3rd edition,"
W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Bill Fenner, and Andrew M. Rudoff, Pearson Education, 2003.

"UNIX Network Programming, Volume 2: Interprocess Communications,"
W. Richard Stevens and Richard W. Stevens, Pearson Education, 1998.

"Intel Threading Building Blocks: Outfitting C++ for Multi-Core Processor Parallelism," James Reinders, O'Reilly Media, Incorporated, 2007.

"Unix Systems Programming: Communication, Concurrency and Threads," Steve Robbins, Prentice Hall Professional Technical Reference, 2003.