Operating systems B

Finalità

The course objectives are to illustrate in a general way the architectures and features of modern operating systems, to introduce the main concepts of concurrent systems, and to improve the students' skills in programming multiprocess applications in the C++/UNIX environment.

Programma

Operating systems architectures. Virtual machines. Process management. CPU scheduling. Memory management. Virtual memory. I/O management. File systems. Security.
Concurrent programming in the global memory model. Mutual exclusion and synchronization primitives. Resource management with semaphores. Conditional critical regions. Monitors. Concurrent programming in Java.
Concurrent programming in the message passing model. Send and receive primitives. Remote procedure call. Guarded commands.
Deadlock. Static and dynamic deadlock prevention and managment techniques.
Distributed systems. Client-server model. RPC in distributed environments. Distributed mutual exclusion, synchronization, and coordination.
Threads and lightweight processes. APIs for multithreading in Solaris and Linux.

Attività d'esercitazione

Laboratory sessions will be mainly devoted to system programming in C/C++ in UNIX and Linux and to the concurrency features of the Java language.

Modalità d'esame

Exam consists of three parts to be taken in sequence: mid-term programming assignment in C++/UNIX, written test, oral discussion. For those students not attending the scheduled class lectures and laboratories, the programming assignment must be taken in a later exam session before the written test.

Propedeuticità

Sistemi operativi A (Operating systems A).

Testi consigliati

A.S. Tanenbaum, Modern Operating Systems, 2nd edition, Prentice-Hall, 2001.
A. Silbershatz, P.B. Galvin, G. Gagne, Operating Systems Concepts with Java, John Wiley and Sons, 2003.
S.J. Hartley, Concurrent Programming - The Java Programming Language, Oxford University Press, 1998.