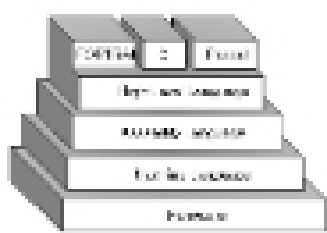


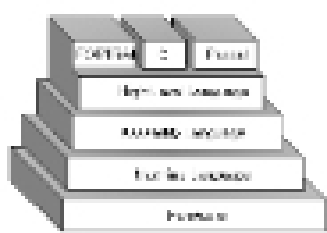
Concurrency

- A PROCESS or THREAD: is a potentially-active execution context.
 - Classic von Neumann (stored program) model of computing has single thread of control.
 - Parallel programs have more than one.
 - A process can be thought of as an abstraction of a physical PROCESSOR.
 - Processes/Threads can come from
 - ◆ multiple CPUs
 - ◆ kernel-level multiplexing of single physical machine
 - ◆ language or library level multiplexing of kernel-level abstraction



Concurrent Processes

- ◆ They can run
 - in true parallel
 - unpredictably interleaved
 - run-until-block
- ◆ Most work focuses on the first two cases, which are equally difficult to deal with.
- ◆ Common scenario:
 - the operating system multiplexes one or more processes on top of one or more physical processors,
 - and a library package or language run-time system multiplexes one or more threads on top of one or more OS processes.



Classes of Programming Notation

- ◆ Two main classes of programming notation
 1. synchronized access to shared memory
 2. message passing between processes that don't share memory

- ◆ Both approaches can be implemented on hardware designed for the other,
- ◆ Although shared memory on message-passing hardware tends to be slow.

- ◆ We'll focus here on shared memory.
 - The book covers both.