Table 6.1 Summary of Deadlock Detection, Prevention, and Avoidance Approaches for Operating Systems [ISLO80]

Principle	Resource Allocation Policy	Different Schemes	Major Advantages	Major Disadvantages
Prevention	Conservative; undercommits resources.	Requesting all resources at once.	•Works well for processes that perform a single burst of activity. •No preemption necessary	•Inefficient •Delays process initiation •Future resource requirements must be known
		Preemption	•Convenient when applied to resources whose state can be saved and restored easily	•Preempts more often than necessary •Subject to cyclic restart
		Resource ordering	•Feasible to enforce via compile-time checks •Needs no run-time computation since problem is solved in system design	•Preempts without much use •Disallows incremental resource requests
Avoidance	Midway between that of detection and prevention	Manipulate to find at least one safe path	•No preemption necessary	•Future resource requirements must be known •Processes can be blocked for long periods
Detection	Very liberal; requested resources are granted where possible.	Invoke periodically to test for deadlock.	•Never delays process initiation •Facilitates on-line handling	•Inherent preemption losses

Table 6.3 Windows 2000 Synchronization Objects

Object Type	Definition	Set to Signaled State When	Effect on Waiting Threads
Process	A program invocation, including the address space and resources required to run the program	Last thread terminates	All released
Thread	An executable entity within a process	Thread terminates	All released
File	An instance of an opened file or I/O device	I/O operation completes	All released
Console Input	A text window screen buffer. (e.g., used to handle screen I/O for an MS-DOS application)	Input is available for processing	One thread released
File Change Notification	A notification of any file system changes.	Change occurs in file system that matches filter criteria of this object	One thread released
Mutex	A mechanism that provides mutual exclusion capabilities for the Win32 and OS/2 environments	Owning thread or other thread releases the mutant	One thread released
Semaphore	A counter that regulates the number of threads that can use a resource	Semaphore count drops to zero	All released
Event	An announcement that a system event has occurred	Thread sets the event	All released
Waitable Timer	A counter that records the passage of time	Set time arrives or time interval expires	All released

Note: Shaded rows correspond to objects that exist for the sole purpose of synchronization.