Quizlet

Operating Systems - Synchronization

Study online at quizlet.com/_2f1wfa

1. **Atomic**	Indivisible
2. **Atomic Execution**	A sequence of linear, indivisible or uninterruptible instructions
3. **Atomic Instruction**	Instructions that cannot be interrupted in their execution
4. **Binary Semaphore**	integer value can range only between 0 and 1
5. **Bounded Waiting**	process p eventually enters its Critical section
6. **Counting Semaphore**	integer value can range over an unrestricted domain
7. **Critical Section**	the portion of code that a process must execute in a exclusive fashion
8. **Critical Section Problem (CSP) Components (4)**	1. entry section, 2. critical section 3. exit section, 4. remainder section
9. **Deadlock**	two or more processes are waiting indefinitely for an event that can be caused by only one of the waiting processes
10. **Deadlock Conditions (4)**	1. Mutual Exclusion 2. Hold and Wait 3. No Preemption 4. Circular Wait (all 4 required for deadlock)
11. **Dining Philosopher Problem**	chopstick[5] semaphore initialized to 1 algorithm can have deadlock and starvation because philosophers pick up chopsticks one at a time, however these chopstick semaphores can be taken by a neighbor, resulting in possible cascading deadlock
12. **Mutex Lock Con**	Requires busy waiting (spinlock)
13. **Mutex Lock Pros (4)**	- Simple solution - Process has no context switching if implemented in user space - Can work for short wait times - Enables for mutual exclusion and progress
14. **Mutual Exclusion Property**	if a process is in its critical section for a resource, then no other process can execute in the critical section for that resource
15. **Mutual Exclusion Scheduling Problems (3)**	1. Deadlock 2. Starvation 3. Priority Inversion
16. **Priority Inversion**	lower-priority process holds or obtains a lock needed by higher-priority process
17. **Process resource utilization system model (3)**	1. Request 2. Use 3. Release
18. **Progress**	some process p enters its Critical section next
19. **Properties of Solutions to Critical Section Problem (3)**	1. Mutual Exclusion 2. Progress 3. Bounded Waiting
20. **Race condition**	a shared resource is used or accessed in a non-deterministic way that may lead to multiple possible outcomes Requires that instructions that ran on it were not atomic
21. **Resource request can be granted to a process only if**	request does not result in the formation of a cycle in the resource allocation graph
22. **Starvation**	lack of progress for at least one process caused by indefinite blocking or lack of scheduling