Quizlet

Operating Systems Chapter 5

Study online at quizlet.com/_1ptx2q

1.	**Abstract Data Type (ADT)**	a mathematical model for a certain class of data structures that have similar behavior; or for certain data types of one or more programming languages that have similar semantics
2.	**Adaptive Mutex**	.
3.	**Assignment Edge**	.
4.	**Atomically**	propelled or driven by
5.	**atomic instructions**	.
6.	**Binary Semaphore**	is a semaphore with the integer value ranges over 0 and 1, allows only one thread to access the resource at a time. but counting semaphore allows N accesses at a time
7.	**bounded buffer**	.
8.	**Bounded-Buffer Problem**	.
9.	**Busy Waiting**	to wait for an event by spinning through a tight look or timed-delay loop that polls for the event on each pass, as opposed to setting up an interrupt handler and continuing execution on another part of the task
10.	**classic problems**	.
11.	**Conditional-Wait**	.
12.	**Cooperating Process**	a process is said to be a cooperating process if it can affect or be affected by other processes in the system
13.	**Coordination**	.
14.	**Counting Semaphore**	.
15.	**Critical Section**	a counting semaphore is a synchronization object that can have an arbitrarily large number of states. the internal state is defined by a signed integer variable, the counter
16.	**Critical Section Object**	provides synchronization similar to that provided by a mutex object, except that a critical section can be used only by the threads of a single process
17.	**Deadlock**	is a situation in which two or more competing actions are each waiting for the other to finish, and thus neither ever does
18.	**Deadlock Avoidance**	.
19.	**Deadlock Prevention**	.
20.	**Dining-Philosophers Problem**	is an example problem often used in concurrent algorithm design to illustrate synchronization issues and techniques for revolving them
21.	**Dispatcher Objects**	.
22.	**Entry Section**	.
23.	**Events**	.
24.	**Exit Section**	.
25.	**Funtional**	.
26.	**Hardware Transactional Memory (HTM)**	.
27.	**Imperative, Procedural**	.
28.	**Indefinite Blocking, Starvation**	when a process doesn't get to execute because the scheduler never lets it
29.	**Locking**	.
30.	**Memory Transaction**	.
31.	**Monitor**	a screen that displays output from a computer
32.	**Mutex Locks**	.
33.	**Named, Unnamed**	.
34.	**Non-preemptive Kernels**	.
35.	**Non-signaled State**	.
36.	**Peterson's Solution**	a concurrent programming algorithm for mutual exclusion that allows two processes to share a single-use resource without conflict using only shared memory for communication

37. **Preemptive Kernels** a method used mainly in monolithic and hybrid kernels where all or most device drivers are run in kernel space, whereby the scheduler is permitted to forcibly perform a context switch on a driver or other part of the kernel during its execution, rather than co-operatively wait for the driver or kernel function to complete its execution and return control of the processor to the scheduler

38. **Priority-Inheritance Protocol** a method for eliminating priority inversion. using this programming method, a process scheduling algorithm increases the priority of a process to the maximum priority of any other process waiting for any resource one which A has a resource lock

39. **Priority Inversion** .

40. **Priority Number** .

41. **Process Synchronization** .

42. **Race Condition** several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place

43. **Readers-Writers Problem** examples of a common computing problem in concurrency. the two problems deal with situations in which many threads must access the same shared memory at one time, some reading and some writing, with the natural constraint that no process may access the share for reading or writing while another process is in the act of writing to it

44. **Reader-Writer** .

45. **Remainder Section** .

46. **Request Edge** .

47. **Semaphore** a variable or abstract data type that is used for controlling access, by multiple processes, to a common resource in a parallel programming or a multi user environment

48. **Signal and Continue** .

49. **Signal and Wait** .

50. **Signaled State** .

51. **Software Transactional Memory (STM)** a concurrency control mechanism analogous to database transactions for controlling access to shared memory in concurrent

52. **Spinlock** a lock which causes a thread trying to acquire it to simply wait in a loop (spin) while repeatedly checking if the lock is available

53. **swap and blah** .

54. **System Resource-Allocation Graph** .

55. **test and set** .

56. **threads** why?
what are they?
multithreading models
threading issues
linux

57. **Transactional Memory** attempts to simplify concurrent programming by allowing a group of loaded and store instructions to execute in an atomic way. it is a concurrency control mechanism analogous to database transactions for controlling access to shared memory in concurrent computing

58. **Turnstile** .