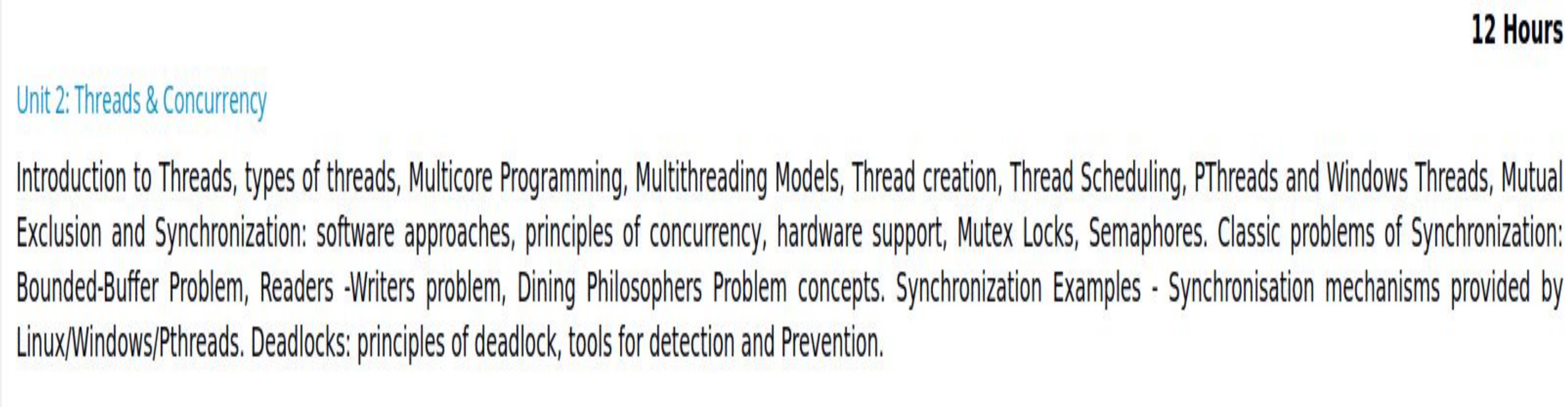
OPERATING SYSTEMS

Synchronization Examples

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 2



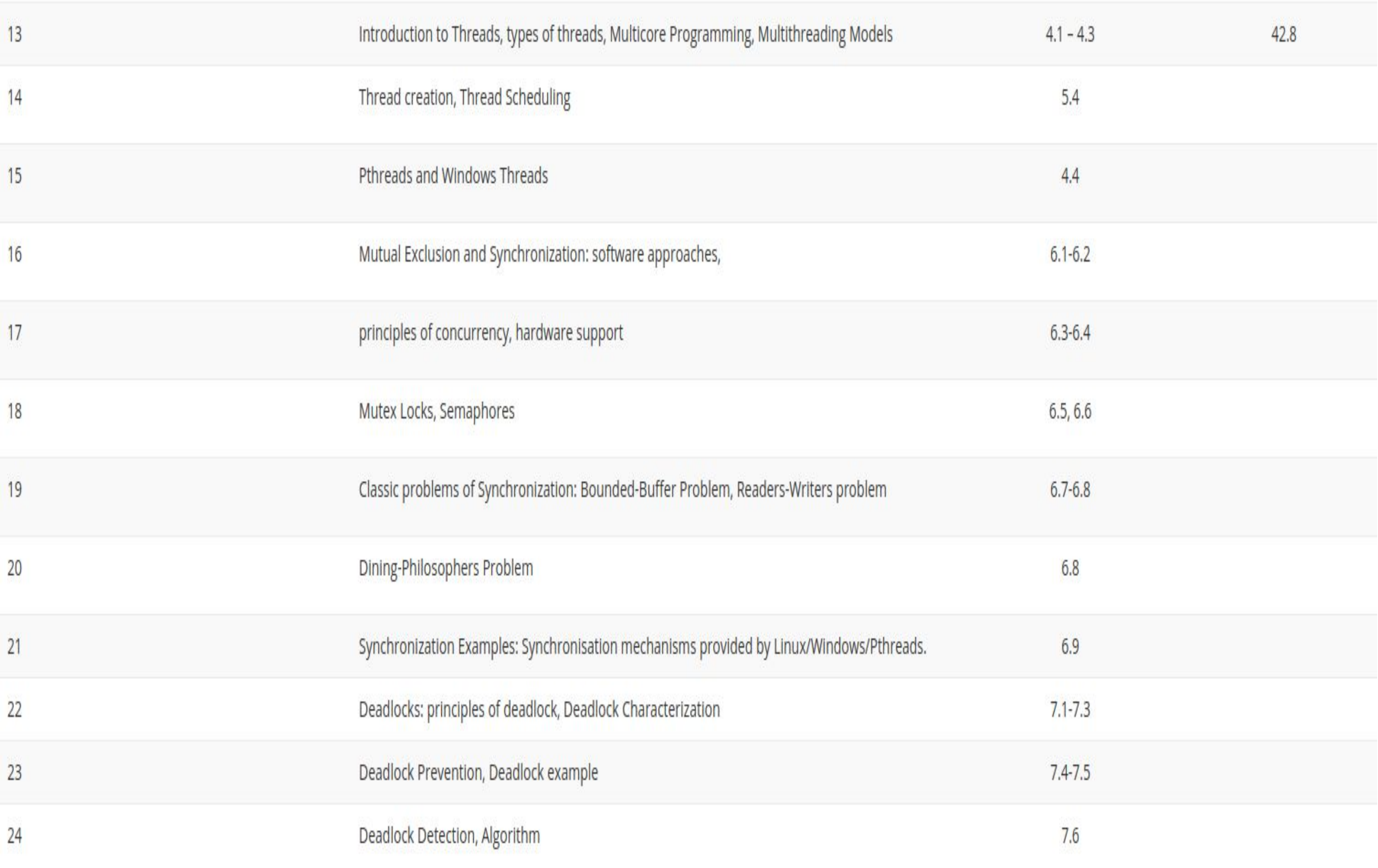
12 Hours

Unit 2: Threads & Concurrency

Introduction to Threads, types of threads, Multicore Programming, Multithreading Models, Thread creation, Thread Scheduling, PThreads and Windows Threads, Mutual Exclusion and Synchronization: software approaches, principles of concurrency, hardware support, Mutex Locks, Semaphores. Classic problems of Synchronization: Bounded-Buffer Problem, Readers -Writers problem, Dining Philosophers Problem concepts. Synchronization Examples - Synchronisation mechanisms provided by Linux/Windows/Pthreads. Deadlocks: principles of deadlock, tools for detection and Prevention.

OPERATING SYSTEMS

Course Outline



13	Introduction to Threads, types of threads, Multicore Programming, Multithreading Models	4.1 - 4.3	42.8
14	Thread creation, Thread Scheduling	5.4	
15	Pthreads and Windows Threads	4.4	
16	Mutual Exclusion and Synchronization: software approaches,	6.1-6.2	
17	principles of concurrency, hardware support	6.3-6.4	
18	Mutex Locks, Semaphores	6.5, 6.6	
19	Classic problems of Synchronization: Bounded-Buffer Problem, Readers-Writers problem	6.7-6.8	
20	Dining-Philosophers Problem	6.8	
21	Synchronization Examples: Synchronisation mechanisms provided by Linux/Windows/Pthreads.	6.9	
22	Deadlocks: principles of deadlock, Deadlock Characterization	7.1-7.3	
23	Deadlock Prevention, Deadlock example	7.4-7.5	
24	Deadlock Detection, Algorithm	7.6	

Topic Outline

- Synchronization Examples
  - Synchronization in Solaris
  - Synchronization in Windows
  - Synchronization in Linux
  - Synchronization using pthread

Synchronization Examples : Synchronization using spinlocks

Spinlocks

- A spinlock is a special type of mutex that does not use OS synchronization functions when a lock operation has to wait. Instead, it just keeps trying to update the mutex data structure to take the lock in a loop.

- If the lock is not held very often, and/or is only held for very short periods, then this can be more efficient than calling heavyweight thread synchronization functions.

- However, if the processor has to loop too many times then it is just wasting time doing nothing, and the system would do better if the OS scheduled another thread with active work to do instead of the thread failing to acquire the spinlock.

Slides Adapted from Operating System Concepts 9/e © Authors

Synchronization Examples : Synchronization in Solaris

- Traditional operating system license (1982 to 2004)

- Open source (2005 until March 2010)

- Post-Oracle closed source (March 2010 to present)

Synchronization Examples : Synchronization in Solaris

- Implements a variety of locks to support multitasking, multithreading (including real-time threads), and multiprocessing

- Uses adaptive mutexes for efficiency when protecting data from short code segments
  - Starts as a standard semaphore spin-lock
  - If lock held, and by a thread running on another CPU, spins
  - If lock held by non-run-state thread, block and sleep waiting for signal of lock being released

Synchronization Examples : Synchronization in Solaris

- The Solaris kernel implements a facility known as turnstiles for managing kernel threads that are waiting for synchronization primitives, which include specifically,

    - mutex (mutual exclusion) locks and reader/writer locks.

    - These are some of the synchronization primitives used to implement support for shared memory processor (SMP - also referred to as symmetric multiprocessor) parallel processing architectures.

Synchronization Examples : Synchronization in Solaris

- Uses condition variables
- Uses readers-writers locks when longer sections of code need access to data
- Uses turnstiles to order the list of threads waiting to acquire either an adaptive mutex or reader-writer lock
- Turnstiles are per-lock-holding-thread, not per-object
- Priority-inheritance per-turnstile gives the running thread the highest of the priorities of the threads in

Synchronization Examples : Synchronization in Windows

- Uses interrupt masks to protect access to global resources on uniprocessor systems
- Uses spinlocks on multiprocessor systems
- Thread using spinlocks will never be preempted
- Also provides dispatcher objects user-land which may act mutexes, semaphores, events, and timers
- Events : An event acts much like a condition variable
- Timers notify one or more thread when time expired
- Dispatcher objects either signaled-state (object available) or on-signaled state (thread will block)

Synchronization Examples : Synchronization in Linux

- Linux: Prior to kernel Version 2.6, disables interrupts to implement short critical sections
- Version 2.6 and later, fully preemptive
- Linux provides:
  - Semaphores
  - atomic integers
  - spinlocks
  - reader-writer versions of both
- On single-cpu system, spinlocks replaced by enabling and disabling kernel preemption

Synchronization Examples : Synchronization using pthread

- Pthreads API is OS-independent

- It provides:
  - mutex locks
  - condition variable

- Non-portable extensions include:
  - read-write locks
  - spinlocks

- https://man7.org/linux/man-pages/man7/pthreads.7.html

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com