

- HW questions
- Course reviews

Final Preparation

CS313

Exam Mechanics

- When: Friday, December 13th, 1:30-3:30
- Wait until you are invited into the room
- Find an unclaimed seat with an exam
- One page of notes is allowed.
 - 8.5x11 paper (double-sided is fine). Typed or handwritten. No Magnification instruments.
- Scratch paper is allowed
- No electronic devices, including: calculators, watches, iwatches, phones, laptops, Tamagotchis, tablets, ...
- Contact drc.ou.edu for appropriate accommodations (drc@ou.edu).

Exam Mechanics

- Multiple choice
- Coverage will be theory to practical programming
- No generation of code
- But: many questions will be about code
 - Here is code, what does it output?
 - Here is what the code is supposed to do + the code; where is the bug?

-> Need to know your API

Midterm Topics

- Byte-level representations and pointers
- Compiler vs linker + Makefiles
- Bit-wise operators in C
- System calls
- Streams
- Files and File Systems
- Processes
- Threads (we were just starting)

Threads

- What is the distinction between a process and a thread?
 - Threads share memory (globals, heap), program spaces
 - Threads have their own stack & registers
- Why should we use threads?
- Parallelization and Amdahl's Law
- User vs Kernel space threads
 - One-to-One, Many-to-One and Many-to-Many

$$speedup \leq \frac{1}{S + \frac{(1-S)}{N}}$$

Implementation of File Systems

- Boot block: code that begins the OS boot process
- Volume Control (or Master) block: disk descriptor (number of blocks, block size, etc.), allocation state
- File meta-data: name, size, access permissions, time stamps, location on disk ...
 - In Unix, we use an INODE for this
- Hard disks (persistent storage) vs memory (ephemeral storage)

Implementation of File Systems

- Implementing directories:
 - Linear lists vs hash tables
- Allocation of disk blocks for file storage:
 - Contiguous
 - Extent-based
 - Indexed (including hierarchies)
 - Linked list

Implementation of File Systems

- Free space management
 - Bit vectors
 - Linked list
- File system recovery
 - Back-ups
 - Log-structured files
 - Copy-on-write: WAFL file system & snapshots

CPU Scheduling

- Performance measures: CPU utilization, throughput, turn-around time, wait time, response time
- Scheduling algorithms
 - First-Come-First-Served
 - Shortest-Job-First
 - Shortest-Remaining-Time-First
 - Priority: with and without preemption
 - Round-Robin
 - Time quanta vs context switch time
 - Multi-level queues
 - Multi-Level Feedback Queue

Synchronization

- Shared data structures
 - Circular buffer
- Producer / Consumer model
- Race conditions and preemption
- Critical sections

Solving the Synchronization Problem

Critical properties that we want to achieve:

- Mutual exclusion
- Progress
- Bounded waiting

Mechanisms

- Hardware solutions: test-and-set, compare-and-swap
 - Mutual exclusion, progress, but no guarantee on bounded wait
- Software solutions:
 - mutex locks: acquire() and release()
 - Semaphores: wait() and signal()
- Busy waiting for many of these methods, but higher level approaches offer process queuing that avoids busy waiting
- Must be careful about starvation

Using Semaphores

- Bounded buffer
- Dining philosophers
- Readers/writers

Deadlock

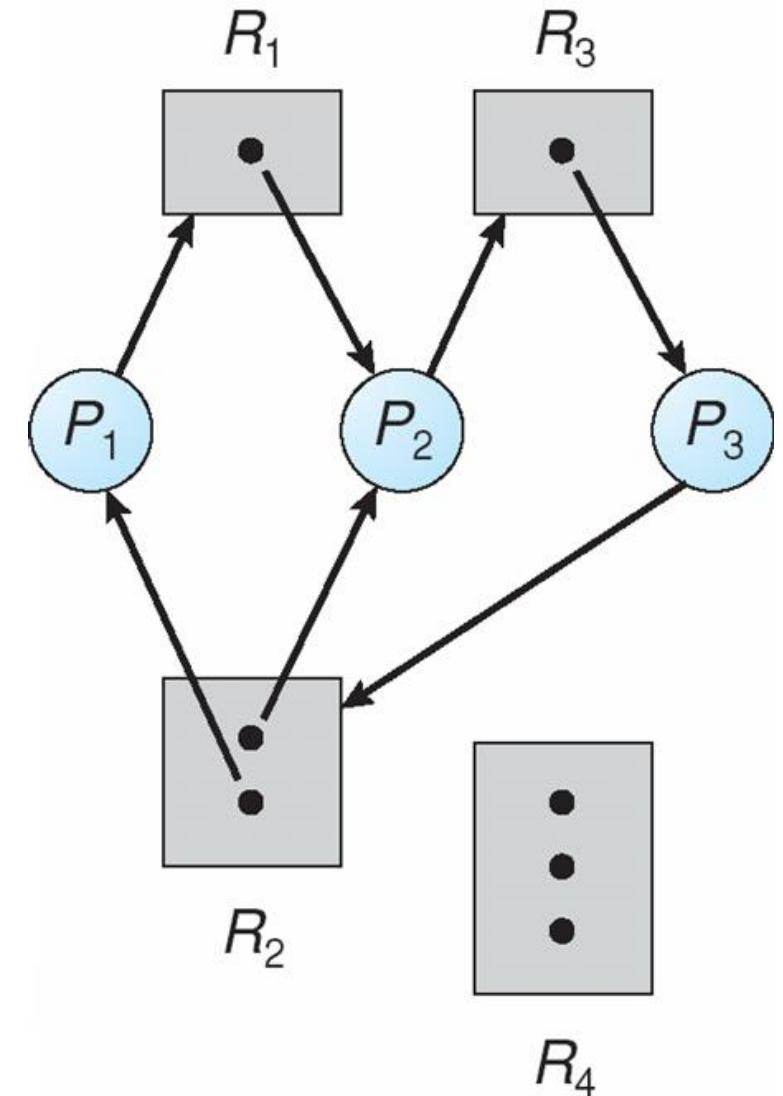
Necessary and sufficient conditions for deadlock (all must be true):

- Mutual Exclusion
- Hold and Wait
- No preemption
- Circular wait

Deadlock

Resource allocation graphs

- Single instance of each resource
- Multiple instances of some resources



Deadlock

- Deadlock Prevention:
 - Fixed set of rules that remove one of the necessary & sufficient conditions for deadlock
- Deadlock Avoidance:
 - Make context-specific decisions on the fly as to whether an allocation request should be granted
 - Single instance per resource type:
 - Use allocation graph
 - If an allocation results in a cycle, then do not grant it
 - Multiple instances per resource type:
 - Banker's Algorithm
 - If an allocation results in an unsafe state, then do not grant it

Office Hours During Finals Week

- Me: T/Th 9-11:30
- Dorian: W 1-3; Th 3-5
- Appointments are possible, too

Preparing

- Lecture notes
- Assigned readings
- Quizzes / homework assignments
- We have also done many coding examples in class
 - Review these: focus on the functionality
- Prior exams: see the ***prior classes*** section of my home page

