

Final Review

I. Processes

- A. Common structures
- B. Isolation models
- C. Communication models

II. Threads

- A. Benefits and limits of threads

III. Process & Thread Scheduling

- A. Scheduling algorithms
- B. Metrics for evaluating scheduling

IV. Synchronization

- A. Requirements for the critical section problem
 - 1. Mutual exclusion
 - 2. Progress
 - 3. Bounded wait
- B. Synchronization primitives
 - 1. A need for atomicity
 - 2. Peterson's algorithm
 - 3. Semaphore
 - a) Counting
 - b) Blocking
 - 4. Mutex
 - 5. Hardware support

V. Deadlocks

- A. Requirements and properties for deadlock
 - 1. **Mutual exclusion**: At least one resource can only be held by one process at a time; this can result in other processes waiting for that resources

2. **Hold and wait:** A process must be holding at least one resource while waiting for other resources (held by other processes)
 3. **No preemption:** Resources can only be released voluntarily by a process; Resources cannot be revoked
 4. **Circular wait:** A set of n waiting process $\{P_0, \dots, P_n\}$ such that P_i is waiting for resources held by $P_{(i+1)\%n}$
- B. Solutions to deadlocks
1. Prevent, Avoidance, Detection
- C. Modeling deadlock
1. Resource allocation graph
- D. Common problems: e.g. Dining philosophers

VI. Memory

- A. Allocation strategies
1. Swapping
 2. Paging
 3. Metrics: e.g. fragmentation (external and internal)
- B. Virtual Memory
1. Structure Page Tables
 - a) TLBs; Indexed; Inverted
 2. Page replacement algorithms & metrics for comparison
 - a) Page misses; thrashing

VII. Security

- A. Common roles of OS for security
- B. Protection and enforcement of policies

VIII. File Systems

- A. Common structures and organizations
- B. Allocation strategies

IX. Storage Devices

- A. Disk properties
- B. Disk scheduling algorithms

X. Thank you!