Final Exam Guide

This material, if not specified, guides you at the level covered at lectures.

Final Exam Specification

- 2 hours long, the exam date, time and location (see email)

- Close book, Close Note

- Question types:
  - 15 Short answer questions (ex. definitions)
    - 2 points each
  - 10 Yes/no questions
    - 2 points each
  - 10 Multiple choice questions
    - 2 points each
  - 3 Long answer questions
    - 10 points each
    - Each question is made up of several small questions

All slides are important

- Components of a computer system
- Structure of computer system
- Computer system organization
- Computer system operation
- Interrupt, interrupt vector, interrupt handling
- System call, direct memory access
- Storage structure and hierarchy, caching
- Multiprogramming, timesharing
- OS transition from user to kernel mode
- Program counter

All slides are important

- OS context switching
- OS CPU scheduling
- OS interfaces (CLI, GUI, touchscreen)
- System calls, examples, implementation, API-System call-OS relationship, parameter passing, standard c library example
- OS structure, simple structure-MS DOS, traditional UNIX system structure, layered structure, microkernel system structure, loadable kernel modules, hybrid (Mach, iOS, Android)

All slides are important

- Process concept, the multiple parts, process in memory
- Process state, diagram of process state
- Process control block, CPU switch from process to process, process representation in Linux
- Process scheduling, ready queue, queuing diagram, schedulers (short term, long term and medium term)
- Process creation
- Process termination
- Inter-process communication, shared memory, message passing (direct and indirect communication)
- Producer-consumer problem, the buffer full and empty conditions
- Pipes, ordinary pipes and named pipes, parent-child relationship

All slides are important

- Motivation for multithreading, multithreaded server architecture, benefits
- Parallelism and concurrency, concurrency vs. parallelism
- Single and multithreaded processes
- User threads, kernel threads, multithreading models
- Implicit threading (thread pools, openMP, grand central dispatch)
- Signal handler
- setjmp, longjmp, sigsetjmp, siglongjmp
- Address space without threading vs. address space with threads
- Thread control block

All slides are important

- Preemptive and non-preemptive, dispatcher, scheduling criteria, scheduling algorithm optimization criteria
- Scheduling algorithms and examples: FCFS, SJF, shortest-remaining-time-first, priority scheduling, round robin, multilevel queue, multilevel feedback queue

All slides are important

- Race condition
- Critical section
- Solution to critical section problem, the three mutual exclusion characteristics and their relationship
- Preemptive vs. non-preemptive
- Peterson’s solution
- Hardware solutions for locking
- Mutex locks
- conditional variable (difference with semaphore)
- Semaphore, counting semaphore, binary semaphore, semaphore implementation, how to use semaphore as mutex
- Deadlock, starvation, and their relationship
- Bounded buffer problem
- Readers-writers problem
- Dining-philosophers problem

All slides are important

- System model
- Deadlock characterization
- Resource allocation graph, basic facts
- Deadlock prevention
- Deadlock detection, wait-for graph

All slides are important

- Background, base and limit registers, hard-ware address protection, logical vs. physical address space, memory-management unit, dynamic relocation using a relocation register
- Swapping, schematic view of swapping, swapping on mobile systems
- Contiguous allocation, multiple-partition allocation, dynamic storage-allocation problem, fragmentation
- Paging, address translation scheme, paging hardware, paging model of logical and physical memory, paging example, implementation of page table, TLB, paging hardware with TLB, effective access time, memory protection, shared pages
- Structure of page table: hierarchical page table (2-level page table)

All slides are important

- Background, virtual memory, virtual address space, shared library using virtual memory
- Demand paging, basic concepts, valid-invalid bit, page fault, steps in handling a page fault, aspects of demand paging, performance of demand paging, demand paging example,
- Copy on write, before vs. after process modifies a copy-on-write page
- Page Replacement Algorithms

All slides are important

- Disk scheduling algorithms
  - FCFS
  - SSTF
  - SCAN
  - C-SCAN
  - C-LOOK

Example Questions

- **Part 1: Short-answer questions (2 points each)**
  - What’s the major difference between a trap and interrupt?
  - A binary semaphore could be used to implement a mutex lock. What should be the value for it to be initialized to be?

Example Questions

- **Part 2: True/False questions (2 points each). If false, please give the correct answer.**

  – API call is the only way for programs to get services from operating system.

  – Mutual exclusion implies progress, progress implies bounded waiting.

Example Questions

- **Part 3: Multiple-choice questions (2 points each). There is <u>only one correct answer</u>. Please circle your answer.**
  - – Which of the following is true about threads?
    - a. Each thread can have its own stack.
    - b. All threads within the same process share local variables.
    - c. All threads within the same process must run on the same core.
    - d. A thread always competes for the CPU with other threads within the same process and can never compete with threads that belong to other processes.

Example Questions

- **Part 4: Long-answer questions (10 points each)**

Please Schedule each of the following inputs using the algorithm shown below it. First give the output in the form of a **Gantt chart**, and then compute the **average waiting time**. Assume that all processes arrive at time 0 but in the order shown in the table. Show your work.

Input 1:

Process	CPU Burst
P_1	6
P_2	4
P_3	9
P_4	3

Apply Round Robin (with a time quantum of 4) (there are negligibly small differences in arrival times) (5 points).

Input 2:

Process	Arrival Time	CPU Burst
P_1	0	7
P_2	2	3
P_3	4	2
P_4	5	6

Apply Shortest Remaining Time First (5 points).

Good Luck!