CS370: Operating Systems

Study Guide for Midterm

Description: Following are some questions to help you study for Midterm. These questions are meant as a starting point for studying, not as a complete description of the material to be covered on the exam. The instructor reserves the right to ask questions of sorts not in this list or covering material from lecture, or the text that is not specifically addressed by this list. Material covered in the assignments and quizzes is included.

You should be prepared is to provide evidence that you have understood the important concepts, terms, algorithms and approaches discussed in the lectures or the text by **S,G & G**.

While a questions may be true/false, multiple choice, some questions will require you to solve problems including those that test the understanding of the approaches we have examined. Contents of quizzes, programming assignments, and self-exercises are included.

Chapter 1: Introduction

- Define operating systems in terms of what they do.
- What is dual mode operation? How does the system switch between modes?
- What is the difference between I/O-bound jobs and CPU-bound jobs?
- What is multiprogramming?
- What is time-sharing? What are the main advantages of time-sharing?
- How is time-sharing usually implemented?
- What is an interrupt vector?
- List instructions that might be usable only from supervisor mode in a dual-mode processor.
- What resources must be protected by the operating system?

Chapter 2: OS structures

- Describe the layered approach for the structure of operating systems. What are some advantages/disadvantages?
- What is a kernel?
- What is a virtual machine?
- What is a command interpreter? How can one be implemented?

Chapter 3: Processes

- What is a process? What is the difference between a process and a program?
- What is a PCB?
- List some of the queues on a typical operating system. Describe the function of each.
- What does the long-term scheduler do?
- What does the medium-term scheduler do?
- What does the short-term scheduler do?

- What is the degree of multiprogramming?
- When is the long-term scheduler invoked?
- What is time sharing? What kind of scheduling does it involve?
- What is a context switch? What does the kernel do during a context switch? (What is the difference between this and the previous question?)
- Describe the producer/consumer problem.
- What happens at process creation and termination?
- What is IPC?
- Describe the difference between direct and indirect communication.
- What is the difference between asymmetric and symmetric direct communication?
- What capacity may a IPC buffer have? How does the capacity affect the communication?

Chapter 4: Threads

- What is a thread?
- How does a thread differ from a process?
- How can threads be useful in Java?
- Describe how user threads differ from kernel threads.
- Describe the actions taken by a kernel to context switch between kernel-level threads.
- Describe the actions taken by a thread library to context switch between user-level threads.
- Describe the 3 multithreading models: many-to-one, one-to-one, many-to-many.
- Which multithreading model is common now?

Chapter 5: Scheduling

- What is a CPU burst? What is an I/O burst?
- What is FIFO?
- What does preemptive mean? Non-preemptive?
- What is the dispatcher? What does it do?
- What performance criteria could be selected for optimizing an operating system?
- What is throughput?
- What is FCFS? What are its advantages/disadvantages?
- What is a Gantt chart? How is it used?
- What is SJF? What are its advantages/disadvantages?
- How are SJF and priority schedulers similar? Different?
- List externally-derived priorities. List internally-derived priorities.
- What is starvation?
- What is aging?
- What is round-robin scheduling?
- What is the time quantum used for?
- How should the time quantum be related to the context switch time?
- How should the time quantum be related to the CPU burst times?
- How can multilevel queues be scheduled?
- What are multilevel feedback queues? What are the defining characteristics?

Chapter 6, 7: Synchronization tools and Examples

- What is a critical section?
- What is the critical-section problem?
- Why have critical sections in programs, as defined by authors of the textbook?
- What three requirements must a solution to the critical-section problem satisfy?
- What is meant by synchronization in this chapter?
- What is mutual exclusion?
- What variables are used in Peterson's Algorithm? Which are common to both processes?
- Describe hardware support used to assist the implementation of mutual exclusion?
- What is a semaphore?
- Define the wait operation: wait(S).
- Define the signal operation: signal(S).
- What is a monitor?
- What is meant by starvation in this chapter?
- Describe the Producer/Consumer problem.
- Describe the Bounded Buffer problem.
- Describe the Readers/Writer problem.
- Describe the Dining Philosophers problem.
- Describe the effect of synchronized in Java.
- Describe the Java functions yield, wait, notify, notifyAll. In what situations are they used? What do they do?
- Compare the concepts of monitors with using Java's synchronized, wait, and notify.

Chap 8: Deadlocks

- True or False? The system model for deadlocks first requires a process request a resource, then use the resource, and finally release the resource.
- What are the four necessary conditions for characterizing deadlock?
- How is a resource allocation graph used to identify a deadlock?
- Describe one strategy for dealing with deadlocks.

Be sure to read over the summary section. Be sure to review the figures in the text and slides.

Acknowledgement: Many of these review questions have been taken from the textbook Silbershatz & Galvin, Operating System Concepts.