# CS370: Systems Architecture & Software
# Study Guide for Midterm Exam 1 (Test 1)

Description: Following are some questions to help you study for Midterm Exam 1. 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.

One important type of question not in this list for which you should be prepared is to provide definitions for important terms and concepts given in lecture or the text.

This exam will cover Chapters 1 through 5 from the Silbershatz, Galvin & Gagne text. It will not cover any material from Chapter 6 or later in the text.

## S,G & G: Chapter 1

1. Text Questions: 1.13, 1.17
2. Define operating systems in terms of what they do.
3. What is dual mode operation? How does the system switch between modes?
4. What is the difference between *I/O-bound* jobs and *CPU-bound* jobs?
5. What is *multiprogramming*?
6. What is *time-sharing*? What are the main advantages of time-sharing?
7. How is time-sharing usually implemented?
8. What is a *real-time system*? List examples.
9. What is an *interrupt vector*?
10. List instructions that might be usable only from supervisor mode in a dual-mode processor.
11. What resources must be protected by the operating system?

## S,G & G: Chapter 2

1. Text Questions: 2.13, 2.18, 2.24
2. Describe the *layered approach* for the structure of operating systems. What are some advantages/disadvantages?
3. What is a *kernel*?
4. What is a *virtual machine*?
5. What is a *command interpreter*? How can one be implemented?

## S,G & G: Chapter 3

1. What is a *process*? What is the difference between a *process* and a *program*?
2. What is a *PCB*?
3. List some of the queues on a typical operating system. Describe the function of each.
4. What does the *long-term* scheduler do?
5. What does the *medium-term* scheduler do?
6. What does the *short-term* scheduler do?
7. What is the *degree of multiprogramming*?
8. When is the long-term scheduler invoked?
9. What is *time sharing*? What kind of scheduling does it involve?
10. What is *swapping*?
11. What is a *context switch*? What does the kernel do during a context switch?
    *(What is the difference between this and the previous question?)*
12. Describe the **producer/consumer** problem.
13. What happens at process *creation* and *termination*?
14. What is *IPC*?
15. Describe the difference between *direct* and *indirect* communication.
16. What is the difference between *asymmetric* and *symmetric* direct communication?
17. What capacity may a IPC buffer have? How does the capacity affect the communication?

## S,G & G: Chapter 4

1. Text Questions: 4.10
2. What is a *thread*?
3. How does a thread differ from a process?
4. How can threads be useful in Java?
5. Describe how user threads differ from kernel threads.
6. Describe the actions taken by a kernel to context switch between kernel-level threads.
7. Describe the actions taken by a thread library to context switch between user-level threads.

8. Describe the 3 multithreading models: many-to-one, one-to-one, many-to-many.

## S,G & G: Chapter 5

1. Text Questions: 5.9, 5.13
2. What is a *CPU burst*? What is an *I/O burst*?
3. What is *FIFO*?
4. What does *preemptive* mean? *Non-preemptive*?
5. What is the *dispatcher*? What does it do?
6. What performance criteria could be selected for optimizing an operating system?
7. What is *throughput*?
8. What is *FCFS*? What are its advantages/disadvantages?
9. What is a *Gantt chart*? How is it used?
10. What is *SJF*? What are its advantages/disadvantages?
11. How are SJF and *priority* schedulers similar? Different?
12. List externally-derived priorities. List internally-derived priorities.
13. What is *starvation*?
14. What is *aging*?
15. What is *round-robin* scheduling?
16. What is the *time quantum* used for?
17. How should the time quantum be related to the context switch time?
18. How should the time quantum be related to the CPU burst times?
19. How can *multilevel queues* be scheduled?
20. What are *multilevel feedback queues*? What are the defining characteristics? How are these queues used in Linux?

Be sure to read over the summary section and exercises for each chapter. Be sure to review the figures in the text and slides.

Acknowledgement:
Many of these review questions have been taken from the Instructor's Manual for the class textbook: Silbershatz & Galvin, *Operating System Concepts*, Addison-Wesley.

The original version of this review was created by Dr. C. Schauble and modified by Dr. Grit.