

Dear Students,

I have put together this review to help you prepare for the exam. You will find information about what you need to read and what type of questions you can get. I may have missed something, but I believe you will be fine if you follow these guidelines. Don't hesitate to send me or the TAs email if you have additional questions.

The exam date is April 11, 2002. The classroom will be announced on the web, check the webpage regularly. Try to be there 15-20 minutes earlier.

The Exam will be 1.5 hours and closed-book. All materials covered in the classroom, even if not in textbook, is required for the exam. Also make sure that you check the homework problems and the labs, you may get questions that relate to them.

The following sections are required from the textbook:

Chapter 1: Introduction – all sections

Recommended exercises: 1.1, 1.3, 1.6, 1.9

Chapter 2: Computer Structures –all sections

Exercises: 2.1, 2.3, 2.4, 2.5, 2.8, 2.9

Chapter 3: Operating System Structures –all sections

Exercises: 3.1, 3.2, 3.8, 3.9, 3.11, 3.12, 3.13, 3.14, 3.16

Chapter 4: Processes –all sections plus UNIX system calls

Exercises: 4.2, 4.4, 4.5, 4.6

Chapter 5: Threads –all sections plus lecture about Java multithreading

Exercises: 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.8, 5.10

Chapter 6: CPU Scheduling

– 6.1-6.7 and Unix System Calls

Exercises: 6.2, 6.3, 6.4, 6.5, 6.8, 6.10

Chapter 7: Synchronization

Sections: 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.8.8, 7.9, 7.10 (note that we have covered some synchronization mechanisms for Java in the lecture about Java multithreading)

Exercises: 7.1, 7.4, 7.7, 7.8, 7.10, 7.11, 7.12

The exam questions will be of the following nature (note list is not exhaustive):

- (1) Short questions. (20%) Require a sentence or two as answers. For example, what is the minimum architecture support for synchronization? What is the difference between test-and-set and swap instructions? What is the difference between asynchronous and synchronous multiprocessing? Rank scheduling techniques based on waiting time. What is the difference between threads and processes? What is the difference between kernel and user space? What is a context switch? How is the DMA used? Give an example of OS and architecture cooperating to solve a problem more efficiently. What are the typical context-switching times in today's machines/OS?
- (2) Problems that require analytical analysis/reasoning. (10%) For example, express the cost of context-switching analytically if the only state that you need to preserve is the content of 32 CPU registers. Assume the following other variables: cost of taking an interrupt T_i , cost of changing the process state C_s , cache access hit-rate of h \$. Assume PCB is in memory and every register read/write is one memory operation. Add other variables as you need.
- (3) Problems about Scheduling. Problems that require analyzing/comparing scheduling algorithms using Gantt charts (similar to homework). I can describe an arrival rate, preemptive and/or non-preemptive scheduling, various burst sizes and ask you to calculate average waiting times, completion times, etc. I can also formulate a question where you need to modify a particular scheduling algorithm to fit a new application domain.
- (4) Representation and use of Processes: I can ask you to write a simple program using system calls. I will provide the semantics of the calls. I can ask you about representation in the OS. I can provide an example using processes and ask you to show its memory layout. I can ask questions about inter-process communication.
- (5) Threads: I can ask you to write simple Java multithreaded applications. I will provide the thread calls that need to use. I can ask you to show its layout in memory. I can ask you about different multithreading implementations in operating systems.
- (6) Problems that require programming in pseudo-code, C, Java of a solution that needs synchronization. Analysis of a given synchronization algorithm. I can provide one implementation of a problem with one set of synchronization primitives and ask you to rewrite it with a different synchronization technique. I can provide a solution and ask you to determine if it can deadlock or if it can starve processes. I can ask you to fix the problem.

Some of the exam questions may be similar to those from the list of exercises. Questions can be asked related to the examples, e.g., simple modifications, given in class.

Questions that relate to the homework can also be asked. The best way to prepare is to study the notes, read the material from the textbook, and try to solve (at least some of) the exercises listed, try a couple of implementations (in pseudo-code and Java) of problems related especially to multithreading, using processes, and synchronization. Check all the programming assignments you had. Make sure you understand how computer architecture and OS interacts.

This is what I won't ask: I will not require you to recall the syntax of system calls (but you need to know the type of calls available), I will not require you to recall the exact implementation of algorithms from the textbook. I will be very flexible in reading your code, i.e., you don't need to put exact package names and/or remember special Java APIs beyond basic Java. If any of these APIs are needed I will provide them, but the point is to solve the problem correctly with a Java like code. You don't need to worry about exception handling unless I advise you to do so.

In general, I will be testing your ability to use this material rather than your recollection of facts. Focus on understanding rather than memorizing when you study. Good luck!

Professor Moritz