

## Spring 2010 CPS312 Exam 1 Review

Expect about 7 - 10 questions ranging over the 8 sections that were covered in class and your Linux textbook. In order to prepare for this exam – review class notes for all subjects covered, read your Linux textbook (concentrating on the command syntax for the most essential commands covered), and review your understanding of thread creation and synchronization by looking at our lecture code examples and the online material resources given on our class website.

### Example questions

#### Basic short answer question.

In your own words, explain the purpose of having an operating system.

Possible Answer: (based on the slide#2 in the lecture entitled “OS Organization” and throughout “OS Overview”)

An operating system provides an abstraction between user/application and the hardware system and also provides an organized use of this abstraction. An OS also simplifies the management and sharing of system resources (eg. Disk management, memory management, etc.)

#### Basic coding question.

Give the correct code to create 2 threads that compete to use a critical section – setup some basic semaphore code that protects the use of the critical section.

#### Basic Linux OS question.

Give the minimum number of Linux commands that will create a subdirectory called “exam” and then copy the contents of a folder located at “/exam/contents” to this new directory.

What is the best way to prepare for this closed book exam?

1. Review lecture notes.
2. Prepare flashcards for terms, definitions, and lists.
3. Use the OS textbook on reserve to solidify your understanding of particular topics covered in lecture
4. Look over the code covered in class – making sure you understand the proper way to use these OS facilities.
5. Read and study the online resources (see our website)
6. Study your Linux textbook (pay particular attention to the most essential Linux commands)

## OS Overview and Usage

Why study OS design?  
What is an Operating System? goals?  
OS abstraction and resource sharing  
Multiprogramming concepts  
Concept of batch programming and time-sharing  
Difference between a classic process and a modern process with threads  
Concept of context switching  
Code to create processes – fork, join, etc.  
Bootstrapping an OS

## Computer Organization (see notes – CPS311 review)

Computer Startup  
Functions of Interrupts, Interrupt Handling  
I/O, Storage Structure  
Storage Hierarchy (Caching)

## OS Organization

Overview of Process, Memory, and Storage Management  
OS design constraints  
Various other design decisions

## Processes and Threads

Thread state transition  
Model of process execution  
Understand pthreads, thread pools, fork and exec, signal handling, cancellation issues  
Programming with POSIX Threads (know the specifics)

## CPU Scheduling

All topics except the specific OS examples  
Model of process execution  
Details of context switching  
What makes a good scheduling policy  
Different scheduling algorithms (understand Gantt charts and criteria for judging the algorithms)

## Process Synchronization

Should understand bounded-buffer, producer/consumer, and readers/writers problem  
Aspects of synchronization – spin locks, mutex, semaphore and semaphore implementation  
Programming with POSIX semaphores (know the specifics)

## Communication

Signal handling, message passing (both asynch and synch), and pipes  
Dining philosopher's problem (deadlock problem), simultaneous semaphores

## Linux OS System (from your Linux textbook Chapters 1 - 9)

Concentrate on the core Linux commands