Review questions for exam

Part 1 Virtualization

Week1 Introduction

Keywords: user mode/kernel mode, virtual machine, multiprogramming, spooling, timesharing, system calls, process, signals, file, file descriptor, shell

1. What are the two functions of an operating systems?
2. What is multiprogramming?
3. Why is the process table needed in timesharing?
4. Suppose that a computer can execute 1 billion instructions/sec and that a system call takes 1000 instructions, including the trap and all context switching. How many system calls can the computer execute a second and still have half the CPU capacity for running application code?

Note these are example question – you can replace the keywords and ask many other similar What/Why/How questions

For example

What is a signal?

Why we need system calls?

Week 2 CPU Scheduling

Keywords: process states, ready queue, pre-emptive, non-preemptive, FIFO, priority scheduling, response time, turnaround time, context switch, batch process, interactive process, time-slice, fairness

1. What does pre-emptive mean?
2. How can we measure CPU throughput?
3. How should the time quantum be related to the CPU burst times?
4. What are multilevel feedback queues?
5. What advantage is there in having different time-quantum sizes on different levels of a multilevel queueing system?
6. Explain the differences in the degree to which the following scheduling algorithms discriminate in favour of short processes:

   a. FIFO b. Round-Robin c. Multilevel feedback queues

7. Five jobs are waiting to be run. Their expected times are 9, 6, 3, 5 and *X*. In what order should they run to minimize average response time? (your answer will depend on *X*).
8. A computer system has enough room for 5 processes in its main memory. These processes are waiting for I/O 60% of the time. What is the CPU utilization for this system?

Week2-3 Virtual Memory

Keywords: relocation, allocation, page, page frame, MMU, page fault, page table, locality, multilevel page table, TLB, inverted page table, page replacement, FIFO, LRU, second chance, NFU, again, working set, thrashing, fragmentation, segment, segment table.

Memory management

1. What is swapping?
2. What is compaction? Why use it?
3. What is a frame?
4. What is contained in the page table?
5. How much fragmentation occurs with paging? Which type?
6. Why are segmentation and paging sometimes combined into one scheme?
7. What is a page fault?
8. What is page replacement? Ideally, what criteria we use to replace pages?
9. Does LRU require extra hardware?
10. What is mean by special locality?
11. Describe the concept of pre-paging and explain its advantages relative to pure demand paging.
12. (advance question) Suppose we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty frame is available or the replaced page is not modified, and 20 milliseconds if the replaced paged is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70% of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

Part 2 Concurrency

Threads, Process Synchronization

Keywords: fork, process state, threads, race conditions, critical section, mutual exclusion, busy waiting, TSL, atomic, wait & post, mutex, monitor, condition variables, starvation, send, receive, notify, rendezvous, traps.

1. What is the fundamental difference between a process and a thread?
2. What does execute "atomically" mean?
3. What is a critical section?
4. What three requirements must a solution to the critical-section problem satisfy?
5. Describe the semaphore operations wait (S) and post (S)
6. Describe the Readers Writers problem.
7. Consider an atomic broadcast in which a producer shares a buffer with multiple consumers. Every item deposited by the producer has to be fetched by all k consumers before the producer can deposit another item into the buffer. Develop a solution for the problem using semaphores for synchronisation.

Deadlock

Keywords: deadlock, livelock, resources, deadlock prevention, deadlock avoidance, safe state

1. Give an example of a deadlock that could occur in the physical world.
2. A computer has six tape drives, with *n* processes competing for them. Each process may need two drives. For which values of *n* is the system deadlock free?
3. Can a system be in a state that is neither deadlock nor safe?
4. What are the four necessary conditions before deadlock can occur? List two ways we can break the fourth condition to prevent deadlock, using resource ordering.

Part 3 Persistence Concurrency

I/O Management

Keywords: block device, character device, device-independent, device controller, memory mapped I/O, DMA, device driver, standard driver interface, disk partitions, subpartitions, RAM disk, disk scheduling, elevator algorithm, bad block, caching

I/O

5. What is memory-mapped I/O?
6. Explain what DMA is and why it is used.
7. Although DMA does not use the CPU, the maximum transfer rate is still limited. Consider reading a block from disk. Name three factors that might ultimately limit the rate of transfer.
8. Disk controllers have internal buffers and they are getting larger with each model. Why?
9. What is a spool?

Disk

10. Disk requests come to the driver from cylinders 10,22,20,2,40,6 and 38 in that order. The arm is initially at cylinder 20 and a seek takes 5 msec per cylinder moved. How much seek time is needed for

    (a) FCFS, (b) SSF and (c) elevator algorithm which is moving upwards

11. How do bad blocks impact on disk performance?
12. In MINIX, device drivers are implemented in user space, but some of its work needs to be done in supervisor mode. Explain how this is implemented.
13. List and briefly described the six operations that are supported by a MINIX block device driver.
14. Compare the performance of C-SCAN and SCAN scheduling, assuming a uniform distribution of requests. Consider the average response time (the time between the arrival of a request and the completion of that request's service), the variation in response time, and the effective bandwidth. How does performance depend on the relative sizes of seek time and rotational latency?

File systems

Keywords: file extension, file structure, file types, file attributes, sequential access, random access, file operations, directories, absolute path, relative path, working directory, contiguous allocation, linked list allocation, indexed, i-node, hard link, symbolic link, reliability, FS consistency, data integrity,

File system

1. What is a file?
2. Can a direct access file be read sequentially? Please, explain
3. What does OPEN do?
4. List ways to share files between directories in operating systems.
5. Rank the allocation methods on speed.
6. Why do some systems keep track of the type of file, while others leave it to the user or simply do not implement multiple file types? Which system is better?
7. Consider a system that supports the strategies of contiguous, linked and indexed allocation. What criteria should be used in deciding which strategy is best utilized for a particular file?