

Final
Jun 10, 2005 10.40–12.30
Good Luck!

Part A. Each question is 5 points

1 What is multiprogramming?

2 What is meant by the term *context switch*? What might cause a context switch to occur?

3 Describe the round-robin scheduling algorithm. Discuss what issues need to be considered to achieve good performance from this algorithm.

4 Explain how, under the basic segmentation scheme, the logical address $\langle \text{segment-number}, \text{offset} \rangle$ is translated into a physical address.

5 What is trashing?

6 A mechanism that can be used for synchronization is the ability to turn on and off interrupts.

a How can you use this to implement a critical section?

b Why does it work?

c Why is this generally a bad idea?

7 Describe four ways to *prevent* deadlock by attacking the conditions required for deadlock.

8 You are to compare reading a file using a single-threaded file server and a multithreaded server. It takes 30 msec to get a request for work, dispatch it, and do the rest of the necessary processing, assuming that the data needed are in the block cache. If a disk operation is needed, as is the case one-third of the time, an additional 90 msec is required, during which time the thread sleeps. How many requests/sec can the server handle if it is single threaded? If it is multithreaded? (Hint: 1000 msec= 1 sec)

10 What is the memory management unit (MMU) and what does it do?

11 One way to use contiguous allocation of the disk and not suffer from holes is to compact the disk every time a file is removed. Since all files are contiguous, copying a requires a seek and rotational delay to read the file, followed by the transfer at full speed. Writing a file back requires the same work. Assuming a seek time of 5 msec, a rotational delay of 4 msec, a transfer rate of 8 MB/sec, and an average file size of 8 KB, how long does it take to read a file into main memory then write it back to the disk at new location? Using these numbers, how long would it take to compact half of a 16-GB disk?

12 Write a pseudocode (or C, Java) for file copying. Don't forget the error checking and reporting but keep as minimal. The buffer size and output mode are 8192 bytes and 764, respectively.

13 You have a 128GB hard disk with 8KB blocks, calculate the necessary list size to keep free disk blocks with

i. *bit tables*

ii. *chained free portion* (a 8-Kb disk block can hold 2048 32-bit disk block numbers)

Hint: assume that all the disk size is free. $2^{24} = 16777216$

14 In a system with pure paging, assume we have a 48-bit address space, and a 64 KB page size.

i. How many bits of an address specify the logical page number (a.k.a. the virtual page number), and how many bits specify the offset? (Hint for part i.: $2^{16} = 65536$)

ii. Let's say we are translating the logical address 0x000000100033; if each logical page is mapped to a physical page that is a single page number higher (i.e., logical page 10 is mapped to physical page 11, logical page 11 is mapped to physical page 12), what is the translated physical address?

(Hint for part ii.: $16^4 = 65536$)