國立臺灣大學99學年度碩士班招生考試試題

1. (18 points) Consider the first reader-writer problem, where no readers will be kept waiting unless a writer has already obtained permission to use some shared data. Please answer the following questions:

1a. (4 points) Suppose that there is no reader or writer initially. When the first reader and the first writer come at the same time. Please tell us which one is favored to enter the critical section first.

1b. (14 points) Let us revise the first reader-write problem such that no writer needs to wait for more than 3 readers. Please use semaphores to write programs for the entry and exit sections of readers and writers.

2. (6 points) Process control provides the ways to create, execute, and terminate processes. Please answer the following questions.

2a. (3 points) Explain how virtual memory improves the performance of process creation such as fork() in UNIX.

2b. (3 points) Define a zombie process and give two reasons to explain why such process is required by operation systems.

3. (10 points) To maximize throughput, a system should simultaneously maximize CPU and I/O device utilization. Consider four common CPU scheduling algorithms, including first-come, first-served (FCFS) scheduling, shortest-job-first (SJF) scheduling, round-robin (RR) scheduling, and priority scheduling algorithms. Please answer the following questions.

3a. (3 points) Consider a system running a few CPU-bound processes and many I/O-bound processes. Which of the four algorithms is the best for maximizing throughput? Please explain you answer.

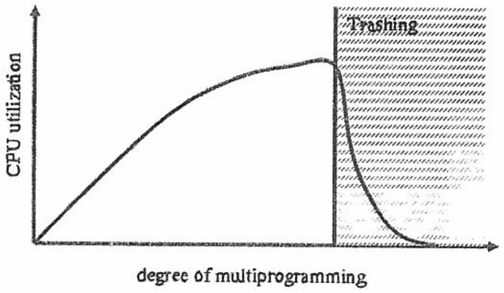
3b. (3 points) Consider a system running only CPU-bound processes. Which of the four algorithms is the best for maximizing throughput? Please explain you answer.

3c. (4 points) One key issue of RR is how to pick an appropriate time quantum. Consider a system with processes, which run and then issue an I/O operation every 20 milliseconds. Assume each I/O operation takes 100 milliseconds and a context switch requires 2.5 milliseconds. Please compare the CPU utilization for RR scheduling with different time quantum durations of 10 milliseconds and 30 milliseconds, where the CPU utilization is defined as the percentage of the CPU time used by the processes. Show your calculation.

見背面

4. (16 points) Thrashing

Most operating system books introduce thrashing in Virtual Memory Management. Thrashing occurs when there isn't enough physical memory to fit working sets from all processes. This produces a large number of page faults, and processes spend more time paging than executing. This thrashing phenomenon is shown in the figure below, in which both CPU utilization and system throughput drop with increasing number of processes.



4a. (6 points) In general, the cause of thrashing is not limited to overuse of physical memory (and the virtual memory management). Overuse in other types of system and networking resources can also produce thrashing in which the overall system performance and throughput decrease with increasing workload. Use simple language to describe a new "thrashing scenario" in which the cause of thrashing is **unrelated** to physical memory resource (e.g., database, networking protocol, etc.).

4b. (5 points) Based on the thrashing scenario you described above, **use simple language** to describe how to detect thrashing.

4c. (5 points) Based on the thrashing scenario you described above, **use simple language** to describe how to prevent or resolve thrashing.

接次頁

5. (25 points) Consider a computer architecture with an instruction set which is composed of *N* different instruction classes. The execution time of class-*i* instruction is 4+*i* ns, where 1≦*i*≦*N*.

5a. (5 points) If the computer is implemented with a single-cycle design, what is the minimum clock cycle time? Why?

5b. (5 points) Consider a multicycle control implementation with a clock cycle of 1 ns. Ignore the overhead associated with multicycle control. What is the performance advantage (speedup) of multicycle control relative to single-cycle control, assuming that the various instruction classes are used with the same frequency?

5c. (5 points) With the performance benefit function derived in part 5b, would a speedup factor of 5 be possible? What would be the requirement to achieve that speedup?

5d. (5 points) Repeat part 5b for the case when the relative frequency of class-i instruction is proportion to *i*.

5e. (5 points) Repeat part 5b for the case when the clock cycle time is 2 ns and the class-i instruction requires (4+*i*)/2 clock cycles. Assume *N*=5 in this case.

6. (15 points) The CPU in a computer system uses two levels of caches L1 and L2. Level L1 is accessed in one clock cycle and supplies the data in case of an L1 hit. For an L1 miss, occurring 4% of the time, L2 is consulted. An L2 hit incurs a penalty of 10 clock cycles while an L2 miss implies a 100-cycle penalty.

6a. (5 points) Assuming pipelined implementation with a CPI (cycle per instruction) of 1 when there is no cache miss, what is the effective CPI if L2's local miss rate is 25%?

6b (5 points) If we were to model the two-level cache system as a single cache, what miss rate and miss penalty should be used?

6c. (5 points) Changing the mapping scheme of L2 from direct to two-way associative can improve its local miss rate to 20% while its hit penalty is increased to 12 clock cycles due to the more complex access scheme. What is the effective CPI after this change? Is this change a good idea?

見背面

7. (10 points) Consider a bus-based shared memory system consisting of three processors. The shared memory is divided into four blocks, *a*, *b*, *c*, *d*. Each processor has a cache that can fit only one block at any given time. Each block can be in one of two states: valid (*V*) or invalid (*I*). Assume that caches are initially flushed (*empty*) and that the contents of the memory are as follows:

Memory block	Contents
a	10
b	20
c	40
d	80

Consider the following sequence of events given in order:

(1) P1: read(a)
(2) P2: read(a)
(3) P3: read(a)
(4) P1: a=a+20
(5) P1: read(c)
(6) P2: read(a)
(7) P3: a=15
(8) P1: c=c+10

7a. (5 points) There are two fundamental cache coherence policies: write-invalidate and write-update. Write-invalidate maintains consistency by reading from local caches until a write occurs. When any processor updates the value of X through a write, it invalidates all other copies by setting the state of the block to invalid (*I*). Suppose write-back and write-invalidate protocols are used in the system, what would be the contents of the caches and memory and the state of cache blocks after the above sequence?

7b. (5 points) On the other hand, write-update maintains consistency by immediately updating all copies in all caches. When a block is updated, the state of the block is set to valid (*V*). Suppose write-through and write-update protocols are used in the system, what would be the contents of the caches and memory and the state of cache blocks after the above sequence?

試題隨卷繳回