國立清華大學 107 學年度碩士班考試入學試題

系所班組別：資訊工程學系

考試科目（代碼）：計算機系統(2302)

1. (3%) Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1 millisecond and that all processes are long-running tasks. What is the CPU utilization for a RR (Round-Robin) scheduler when the time quantum is 5 milliseconds?

2. (12%) Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process	Burst Time	Priority
P_1	12	2
P_2	8	1
P_3	5	4
P_4	10	5
P_5	7	3

The processes are assumed to have arrived in the order of P_1, P_2, P_3, P_4, P_5, all at time 0.

a. (6%) Draw three Gantt charts illustrating the execution of these processes using the following scheduling algorithms: SJF (Shortest Job First), nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 5 milliseconds).

b. (6%) What is the waiting time of each process for each of the scheduling algorithms in Part a ?

3. (10%) Consider the Banker's algorithm for deadlock avoidance. The following is a snapshot of a system with four types of resources A, B, C, and D.

	Allocation	*Max*
	A B C D	*A B C D*
P_0	0 6 3 2	0 6 5 2
P_1	0 0 1 2	2 0 3 2
P_2	0 0 1 4	2 6 5 6
P_3	1 0 0 0	1 7 5 0
P_4	1 3 5 4	2 3 5 7

Available
A B C D
1 5 2 0

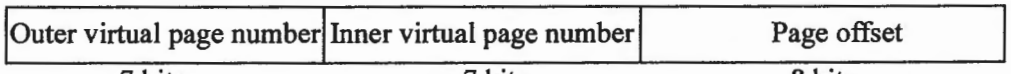
Allocation: An $n \times m$ matrix defines the number of resources of each type currently allocated to each process. If ***Allocation***$[i][j]$ equals k, then process P_i is currently allocated k instances of resource type Rj.

Max: An $n \times m$ matrix defines the maximum demand of each process. If ***Max***$[i][j]$ equals k, then process P_i may request at most k instances of resource type Rj.

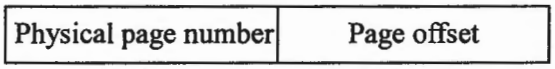
Available: A vector of length m indicates the number of available resources of each type. If ***Available***$[j]$ equals k, then k instances of resource type Rj are available.

Suppose that the algorithm checks the processes in the order of P_0, P_1, P_2, P_3, P_4. Give the number of available resources of each type after each iteration of the banker's algorithm.

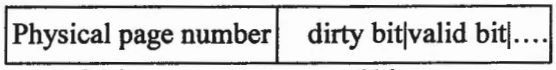
4. (11%) Consider a two-level page table memory management scheme that translates 22-bit virtual addresses to 16-bit physical addresses using page tables with 16-bit table entries. All the address formats are shown below:



22-bit virtual address



16-bit physical address



16-bit page table entry

a. (2%) Explain what the purpose of dirty bit is.

b. (2%) Give a logical reason why the designer might have made the virtual page number field 7 bits each for the inner and outer page tables.

c. (3%) Consider a computer system that uses a TLB (Translation Lookaside Buffer) with only 2 entries with memory access time 200 nanoseconds and TLB search time 10 nanoseconds. What is the average effective memory access time for a process that is just context switched to access the sequence of hexadecimal virtual memory addresses: 22A956, 22AD76, 22ADFF, 22AF56, 22A958, 22AF58 ?

d. (4%) Consider the case of an address translation from a virtual address 22AF58 to a physical address 7B58. Assume that we know the base address for the inner page table used in this translation is at the physical address location: A700. Which entry of the outer table is used for this address translation? What is the physical page number stored in that outer table entry? Which entry of the inner table is used for this address translation? What is the physical page number stored in that inner table entry?

5. (8%) Answer the following questions related to I/O methods.

a. (3%) Compare I/O based on polling with interrupt-driven I/O. In what situation would you favor one technique over the other?

b. (5%) Use an example to illustrate the steps of DMA (Direct Memory Access) I/O. Explain why DMA can improve overall system performance.

6. (6%) Which of contiguous, linked, indexed file allocation method would you use in the following scenario and why?

a. (2%) A swap space manager that stores pages in a limited amount of disk space.

b. (2%) A database that serves record insertion, deletion, and search operations.

c. (2%) A logging system that periodically appends new messages to log files.

7. (12%) In computer architectures, an operation can have several instructions for different addressing modes. For instance, Figure 1 shows four Add instructions of different addressing modes.

Addressing mode	Example instruction	Meaning
Immediate	`Add R4,#3`	`Regs[R4] ← Regs[R4] + 3`
Displacement	`Add R4,100(R1)`	`Regs[R4] ← Regs[R4] + Mem[100+Regs[R1]]`
Register indirect	`Add R4,(R1)`	`Regs[R4] ← Regs[R4] + Mem[Regs[R1]]`
Scaled	`Add R1,100(R2)[R3]`	`Regs[R1] ← Regs[R1] + Mem[100+Regs[R2] + Regs[R3]*d]`

Figure 1: Four addressing modes for Add operation. In the Scaled addressing mode, *d* is the size of an element.

a. (4%) Explain when to use each instruction from the high level programming language aspect.

b. (4%) What are the advantages or disadvantages of supporting different addressing modes for the same operation in the instruction set design?

c. (4%) Why do all the instructions in Figure 1 have at least one addend stored in registers? What would be the problems if an architecture has an instruction like

Add (R1), (R2)

which performs Mem[Regs[R1]] ← Mem[Regs[R1]] + Mem[Regs[R2]]?

8. (13%) Pipeline is often used in computer architecture to accelerate the system performance. However, its effectiveness is influenced by many factors.

a. (4%) What are pipeline hazards? Explain what three major types of pipeline hazards are.

b. (5%) The frequency of branches of a program and their penalties for predicted taken/untaken are summarized in the following table. Suppose that a branch predictor can successfully predict 80% of a conditional branch taken or untaken (which means if a conditional branch is taken, the predictor has 80% to predict taken, and 20% to predict untaken; and if a branch is untaken, the predictor has 80% to predict untaken, and 20% to predict taken). Suppose that the ideal CPI (Cycles Per Instruction) of the pipelined processor without any branches is 1. What is the effective CPI of the program with this branch predictor?

Branch type	*Frequency*	*Penalty of predicted taken*	*Penalty of predicted untaken*
Unconditional branch	5%	2 cycles	2 cycles
Conditional branch, taken	5%	2 cycles	4 cycles
Conditional branch, untaken	10%	3 cycles	0 cycle

c. (4%) The early architecture design usually has a very long pipeline. For instance, Intel Prescott (2004) has 31 pipeline stages. But recent architectures have relatively short pipelines. For example, Intel Kaby Lake (2016) has only 14 stages. Explain the possible reasons of this trend of architecture design.

9. (6%) In general, a fully associative cache is better than a direct-mapped cache in terms of miss rate. However, it is not always the case for a cache, especially for a small cache. Give an example with 10 times memory references and use it to demonstrate that a direct-mapped cache outperforms a fully associative cache in terms of miss rate in detail under the replacement policy, LRU (Least Recently Used).

10. (14%) Given a unified cache consisting of instruction cache and data cache with NO write-buffer in a memory system. It uses the write through mechanism when a write-miss occurs. Given the following measurements:
    - The base CPI with a perfect memory system = 1.5
    - The miss rate = 5%
    - The memory latency = 100 cycles
    - The transfer rate = 4 bytes/clock cycle
    - 30% of the instructions are data transfer instructions
    - The size of a block = 32 bytes

a. (2%) Explain "write-buffer".
b. (2%) Explain "write through".
c. (10%) Determine the effective CPI for this memory system without considering the TLB.

11. (5%) For a typical disk without considering queuing delay, the average seek time is 5 milliseconds, and the transfer rate is 2M-bytes per second. The disk rotates at 600 RPM (Revolution Per Minute), and the controller overhead is 0.5 millisecond. Determine the average time to read a 1792-bytes sector.