國立清華大學 102 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(2002)

1. (10%) Suppose that a disk drive has 1,000 cylinders, numbered 0 to 999. The drive is currently serving a request at cylinder 200. The queue of pending requests, in FIFO order, is

   80, 910, 190, 120, 10, 280, 900

   Starting from the current head position, what is the reading order from each of the following disk-scheduling algorithm? (no need to compute the total distance)
   (a) SSTF
   (b) SCAN (move to higher number first)
   (c) C-SCAN (move to higher number first)

   Note that SSTF stands for shortest-seek-time-first and C-SCAN stands for circular SCAN.

2. (4%) Figures (a) and (b) below depict two different RAID designs where ***Xp*** is the parity bit of data ***X***. Explain what is the main advantage of the RAID in Fig.(b) over the RAID in Fig.(a).

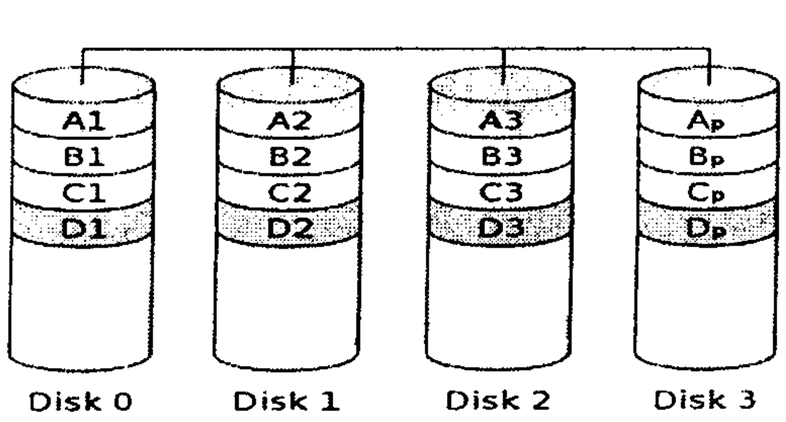

Fig. (a)

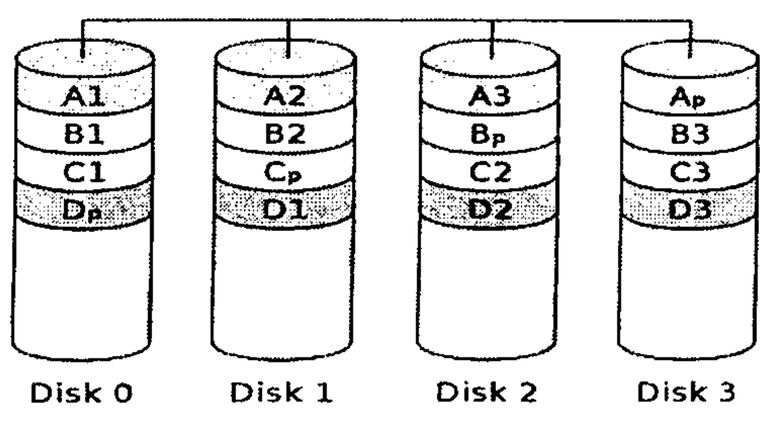

Fig. (b)

3. (10%) Suppose we have a demand paged memory. In the system, we try to reduce the page faults with LRU and FIFO replacement schemes, respectively. Consider the following two C programs:

program A and program B. Assume that the number N in the program is sufficiently large.

Program A:

```
#include <stdio.h>
double A[N][N];
double Anew[N][N];
Compute() {     int i,j;
                for (i=1; i<(N-1); i++)
                        for (j=1; j<(N-1); j++)
                                Anew[j][i]=
        0.25*(A[j][i+1]+A[j][i+2]+A[j][i-1]);
}
```

Program B:

```
#include <stdio.h>
double A[N][N];
double Anew[N][N];
Compute() {     int i,j;
                for (j=1; j<(N-1); j++)
                        for (i=1; i<(N-1); i++)
                                Anew[j][i]=
        0.25*(A[j][i+1]+A[j][i+2]+A[j][i-1]);
}
```

(a) Assume LRU replacement algorithm is used. Which program will be likely with less page faults? Please explain the reason.
(b) Assume FIFO replacement scheme is used. Which program will be likely with less page faults? Please explain the reason.

4. (4%) Which of the following programming techniques and structures are good for a demand-paged environment? Which are "not good"? Explain your answers.
   a. Stack
   b. Sequential Search
   c. Vector operations
   d. Indirection

5. (9%) Consider a system running **two** I/O-bound processes, P1 and P2, and **one** CPU-bound process, P3. Assume that **each** I/O-bound process issues an I/O operation after **0.9** millisecond of CPU computing and that each I/O operation takes **8.9** milliseconds to complete. Assume multiple I/O operations can be executed simultaneously without causing any delay. Also assume that the context switching takes **0.1** millisecond and that all processes are long running tasks. Suppose the Round-Robin scheduler is used and the time quantum is **3.9** millisecond.
   (a) Suppose initially the processes in the waiting queue are P1, P2, and P3. Draw the Gantt chart for the first 20 milliseconds.
   (b) What is the CPU utilization in the long run?
   (c) What is the throughput of I/O operations (per second) in the long run?

6. (8%) Synchronization barrier is a common paradigm in many parallel applications. A barrier is supposed to block the calling thread until all *N* threads have reached the barrier. The following code uses semaphores to implement the barrier function. The initial values of variables are

```
N = 2   (which means there are two processes, P1 and P2.)
count = 0
R1 = Semaphore(1)
R2 = Semaphore(0)
```

```
1. R1.wait()
2. count = count + 1
3. if (count == N) R2.signal()
4. R2.wait()
5. R2.signal()
6. R1.signal()
```

   (a) Show that the execution sequence `P1(1)`, `P1(2)`, `P1(3)`, `P1(4)`, `P2(1)` causes a deadlock, where `Pi(j)` means process *i* executes line *j*.
   (b) Draw the Resource-Allocation Graph for the system after the execution sequence.
   (c) Does your Resource-Allocation Graph have a cycle? If yes,

label each edge with an execution code `Pi(j)` that creates the edge. If no, explain why your Resource-Allocation Graph cannot detect the deadlock.

(d) Rearrange the above code to produce a correct implementation of synchronization barrier without any deadlock for arbitrary execution order of `P1` and `P2`. Give your answer in a permutation of number 1 to 6.

7. (8%) Given the table of latencies of individual stages of a MIPS instruction, please answer the following questions:

Instruction fetch	Register read	ALU operation	Data Access	Register write
200ps	100ps	250ps	400ps	100ps

A. What is the clock cycle time in a nonpipelined and pipelined processor?
B. What is the total latency of a lw instruction in a nonpipelined and pipelined processor?
C. If the time for an ALU operation can be shortened/increased by 25%, how much speedup/slowdown will it be from pipelining? Explain the reason.
D. If you can split one stage into two, each with half the original latency, to improve the pipelining performance. Which stage would you split and what is the new clock cycle time in a pipelined processor?

8. (8%) Please answer the following questions about interfacing I/O devices to the processor and memory:
A. (4%) Describe the following terminologies:
   - Memory-mapped I/O.
   - I/O instruction.
   - Device pooling.
   - Interrupt-driven communication.

B. (2%) Which communication pattern is most appropriate for a "Video Game Controller"? Explain.
C. (2%) Prioritize the following interrupts caused by different devices:
   - Mouse Controller.

- Reboot.
- Overheat.

9. (5%) In general, a fully associative cache is better than a direct-mapped cache in term of miss rate. However, it is not always the case for a cache, especially for a small cache. Please design an example to demonstrate that a direct-mapped cache outperforms a fully associative cache in term of miss rate under the replacement policy, Least Recently Used (LRU).

10. (10%) Consider adding a new index addressing mode to the machine A such that the following code sequence

```
ADD   R1,   R1,   R2   // R1 = R1+R2
LW    Rd, 0(R1)        //load
```

can be combined as

```
LW    Rd, 0(R1+R2)
```

(a) Assume that the load and store instructions are 10% and 20% of all the benchmarks considered, respectively, and this new addressing mode can be used for 5% of both of them. Determine the ratio of instruction count on the machine A before adding the new addressing mode and after adding the new addressing mode.
(b) Assume that adding this new addressing mode also increases the clock cycle by 3%, which machine (either machine A *before* adding the new addressing mode or machine A *after* adding the new addressing mode) will be faster and by how much?

11. (8%) The following Boolean equation describes a 3-output logic function.

$$O1 = A'BC'D' + AB'C + FGH + E'$$
$$O2 = BCD + CDE'H + FGH'$$
$$O3 = AB'CD + ABEF + F'G'$$

(a) Draw a circuit diagram for *O1* using only 4-input NAND gates.
(b) If the 3-output function is implemented with a read-only memory, what size of ROM is needed?

12. (8%) Assume the floating point is stored in a 32 bits wide format. The leftmost bit is the sign bit, the exponent of 16 in excess-64 format is 7 bits wide, and the mantissa is 24 bits long and a hidden 1 is assumed.
    (a) (2%) What is the smallest positive number representable in this format (write the answer in hex and decimal)?
    (b) (2%) What is the largest positive number representable (write the answer in hex and decimal) ?
    (c) (2%) Assume A=2.34375x10^{-2} and B=2.9015625x 10^{2}. Write down the bit pattern using this format for A and B.
    (d) (2%) Using the format presented in (c), calculate the $A - B$.

13. (8%) Assume garage collection comprises 30% of the cycles of a program. There are two methods used to speed up the program. The first one would be automatically handle garbage collection in hardware. This causes an increase in cycle time by a factor of 1.4. The second one provides new hardware instructions that could be used during garbage collection. This would use only one-third of instructions needed for garbage collections but increase the cycle time by 1.2. Compute the program execution time for both methods and show which one is better.