Part A is *Operating systems*, and Part B *Computer organization*. Your answers for part A must be placed before your answers for part B. The misplaced answers will not be graded.

Part A: Operating systems

1. (5%) Why is the separation of policy from mechanism an important principle in system implementation?

2. (10%) Researchers have argued that none of the basic approaches for handling deadlocks (prevention, avoidance, and detection) alone is appropriate for the entire spectrum of resource-allocation problems encountered in operating systems. One possibility is to combine the three approaches, allowing the use of the optimal approach for each class of resources in the system.

   (a) How does it handle deadlock problem among the classes?
   (b) How does it handle deadlock problem within each class?

3. (10%) The following is to implement a counting semaphore using binary semaphores. Let S be a counting semaphore. We use the following data structures:

   ```
   var S1: binary- semaphore;
   S2: binary- semaphore;
   S3: binary- semaphore;
   C : integer;
   ```

   Initially $S1 = S3 = 1, S2 = 0$, and the value of integer C is set to the initial value of S. The *wait* operation on S can be implemented as follows:

   ```
   wait(S3);
   wait(S1);
   C := C - 1;
   if C < 0 then begin signal(S1);
   wait(S2)
   end
   else signal(S1);
   signal(S3);
   ```

   Write the complete implementation of the *signal* operation on S, based on the partial answer given as follows:

   ```
   wait(______);
   C := C + 1;
   if C <= 0 then signal(______);
   signal(______);
   ```
4. (10%) i) What is logical address? ii) What is physical address? iii) Draw a figure to show the address translation using a relocation register and a limit register.

5. (15%) Consider a paging system with the page table stored in memory. It uses associative registers to speedup memory accesses. Each register consists of two parts: a key and a value. i) Draw a figure to show the translation of a logical address (p, d) to a physical address (f, d). ii) If a memory reference takes 220 nanoseconds and finding a page-table entry in the associative registers takes 120 nanoseconds, determine the effective access time for a 98-percent hit ratio.
Part B. 計算機組織 (50%)

1. What is called Amdahl's law? Give a simple example to explain it clearly and briefly.
(4%)

2. What is called branch hazard? How to overcome it? Explain your methods clearly and briefly as many as possible you have known.
(6%)

3. It is known that two-level cache design plays important role in high performance system.
(6%) Suppose we have a processor with a base CPI of 1.0, assuming all reference hit in the primary cache, and clock rate of 500MHz. Assume a main memory access time of 100ns, including all the miss handling. Suppose the miss rate per instruction at the primary cache is 5%. How much faster will the machine be if we add a secondary cache that has a 20ns access time for either a hit or a miss and is large enough to reduce the miss rate to main memory to 2%?

4. Give simple examples to explain the following terminologies clearly and briefly.
(9%)
   (1) Data hazard in pipeline design. (3%)
   (2) Round to nearest even in floating point operations. (3%)
   (3) Memory interleaving technique. (3%)

5. Let a be a multiplier and b be the multiplicand, and they are 4-bit two's complement signed integers. Explain that these two numbers can work in the Booth's algorithm.
(4%)

6. Each instruction needs from three to five of these steps: (i) instruction fetch;
   (ii) instruction decode/register fetch; (iii) execution, address computation, branch/jump completion; (iv) memory access or R-type completion; and (v) memory read completion.
   Here we have operations to fulfill the above steps for different instructions:
   (1) ALUOut = A op B; (2) ALUOut = A + sign-extend(IR[15-0]); (3) if (A == B) then PC = ALUOut; (4) PC = PC[31-28] || (IR[25-0] << 2); (5) IR = Memory[PC]; (6) PC = PC + 4;
   (12) MDR = Memory[ALUOut]; and (13) Reg[IR[15-11]] = ALUOut.
   For the load instruction, select appropriate operation(s) to put them into each of the three to five steps. You are required just write down Roman numbers followed by one or several Arabic numbers.
7. There are sixteen terminologies: (i) aliasing; (ii) branch prediction buffer; (iii) conflict miss; (iv) daisy chain arbitration; (v) direct-mapped cache; (vi) DMA; (vii) finite state machine; (viii) instruction set architecture; (ix) non-blocking cache; (x) page fault; (xi) polling; (xii) RAID; (xiii) reservation station; (xiv) speculative execution; (xv) superscalar; and (xvi) translation-lookaside buffer.

Choose the most appropriate explanation from the following sentences for each of the sixteen terminologies. You are required just write down each Roman number followed by one Arabic number.

(1) An event that occurs when an accessed page is not present in main memory.
(2) A sequence of bus operations that includes a request and may include a response, either of which may carry data.
(3) A technique in which data blocks needed in the future are brought into the cache early by the use of special instructions that specify the address of the block.
(4) It adds redundant disks to the arrays, offering the opportunity for the array to discover a failed disk and automatically recover the lost information.
(5) It occurs in a set-associative or direct-mapped cache when multiple blocks compete for the same set.
(6) A bus mechanism in which the bus grant line is run through the devices from highest priority to lowest so that when the bus is requested the highest priority device sees the bus grant signal first.
(7) A cache that keeps track of recently used address mapping to avoid an access to the page table.
(8) A series of steps used to coordinate asynchronous bus transfers in which the sender and receiver proceed to the next step only when both parties agree that the current step has been completed.
(9) A structure in which each memory location is mapped to exactly one location in the cache.
(10) A structure in which a block can be placed in any location in the cache.
(11) A mechanism that provides a device controller the ability to transfer data directly to or from the memory without involving the processor.
(12) A cache miss caused by the first access to a block that has never been in the cache.
(13) A mechanism that allows each device to independently requests the bus and that uses a scheme for retrying the arbitration when multiple simultaneous request occur.
(14) A sequential logic function consisting of a set of inputs and outputs, a next-state function that maps the current state and the inputs to a new state, and an output
function that maps the current state and possibly the inputs to a set of asserted outputs.

(15) A method of resolving a data hazard by retrieving the missing data element from internal buffers rather than waiting for it to arrive from programmer-visible registers or memory.

(16) A small memory that is indexed by the lower portion of the address of the branch instruction and that contains one or more bits indicating whether the branch was recently taken or not.

(17) A pipelined execution, in which all following instructions must wait when an instruction is blocked from executing.

(18) An abstract interface between the hardware and the lowest level software of a machine that encompasses all the information necessary to write a machine language program that will run correctly, including instructions, registers, memory size, and so on.

(19) The fraction of memory accesses not found in a level of the memory hierarchy.

(20) It allows the processor to make references to the cache while the cache is handling an earlier miss.

(21) The results of pipelined execution need not be written to the programmer-visible state in the same order that instructions are fetched.

(22) The process of periodically checking the status of an I/O device to determine the need to serve the device.

(23) A buffer that holds instructions in a dynamic pipelined machine whose results have not yet been committed to programmer-visible registers or memory.

(24) A pipelining technique that combines dynamic scheduling with branch prediction.

(25) A pipelining technique that enables the processor to execute more than one instruction per clock cycle.

(26) A situation in which the same object is accessed by two addresses; can occur in virtual memory when there are two virtual addresses for the same physical page.

(27) A cache miss that occurs because the cache, even with full associativity, cannot contain all the block needed to satisfy the request.

(28) A buffer within a functional unit that holds the operands and the operation.