1. **A processor uses 32-bit virtual address and 32-bit physical address. Design the TLB and cache systems for this processor with the following configurations:**
   The instruction cache is two-way set associative design using 32-byte line size. The instruction cache size is 16 KB. The data cache is direct-mapped design using 32-byte line size. The data cache size is 32KB. Both the cache tags use physical address. The page size used is 16KB. The ITLB has 32 entries in total using four-way set associative mapping. The DTLB has 64 entries in total using direct-mapped structure.
   (a). Show the ITLB and instruction cache design. 10%.
   (b). Show the DTLB and data cache design. 10%
   (c). Show the block diagram of a 5-stage pipelined processor integrated with the above designs. 20%.

2. **The operating system treats part of the main memory as a software cache for virtual memory operation. What is the preferred cache organization of this cache? Write policy? Replacement policy? Why? 10%**

3. **What is the supervisor or kernel mode? What is the user mode? What are the differences? Why are they needed? 8%**

4. **Explain in detail what happens when a unix fork() system call is made (at both the kernel and user levels). 8%**

5. **Describe the pros and cons of writing a program with multiple threads (as opposed to a single thread). 8%**

6. **What is the difference between a binary and a counting semaphore? Give a short example of where a counting semaphore is useful. 6%**

(背面仍有題目,請繼續作答)

7. **Consider the following set of processes, with the length of the CPU burst given in milliseconds:**

Process	Burst Time	Priority
P1	10	3
P2	1	1
P3	2	3
P4	1	4
P5	5	2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

(a) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a smaller priority number implies a higher priority), and RR (quantum =1). 5%

(b) What is the turnaround time of each process for each of the scheduling algorithms in part (a)? 5%

(c) What is the waiting time of each process for each of the scheduling algorithms in part (a)? 5%

(d) Which of the algorithms in part (a) results in the minimum average waiting time (over all processes)? 5%