1. (18 pt) Implement the inner product calculation of two 64-entry vectors (i.e. $\sum_{i=0}^{63} x_i \times y_i$).
   (a) Assume x_i and y_i are signed integers (int). Write a C program to calculate their inner product. Use a loop to accumulate the products (i.e. with the loop body of "acc += x[i]*y[i];")

   (b) Assume x_i is {1, -2, 3, -4, 5, -6, ..., -62, 63, -64}, where odd numbers are positive and even numbers are negative. Find the binary representations (2's complement) of x_0, x_1, x_2, and x_3 respectively.

   (c) The leading bits of the binary representations for small numbers (such as those in x_i in (b)) are all sign bits (i.e. 0's for positive and 1's for negative). Therefore, the inner product calculation of (b) will have significant bit transitions (due to the alternated signs) and thus consume large power. Someone unrolled the loop body in (a) as
   acc += x[2*j]*y[2*j];
   acc -= x[2*j+1]*y[2*j+1];
   with all-positive x_i = {1, 2, 3, 4, 5, 6, ..., 62, 63, 64} to reduce bit transitions (i.e. the leading sign bits are all zeroes now), but found it did not actually save power for some processors (also with a single hardware multiplier only). List two possible reasons.

   (d) An alternative to implement the low-power inner product in (c) is to use two separate loops: one for "acc += x[2*j]*y[2*j];" and the other for "acc -= x[2*j+1]*y[2*j+1];". Can this approach help to save power? Briefly explain your answer.

2. (7 pt) Both the following assembly codes implement the C statement "if((R1==R2)&&(R3==R4)) R5++;".

```
        BNE R1, R2, Exit
        BNE R3, R4, Exit
        ADD R5, R5, #1
EXIT:
```
(a)

```
CMP R1, R2
CMPEQ R3, R4
ADDEQ R5, R5, #1
```
(b)

   Calculate the execution times of each assembly code on a classical 5-stage pipelined processor (i.e. IF / ID / EX / MEM / WB), where branch instructions are resolved at EX.

3. (15 pt) 綜合簡答題:
   (a) (3 pt) Explain the process of starvation and how aging can be used to prevent it.

   (b) (3 pt) What is the difference between deadlock prevention and deadlock avoidance?

   (c) (3 pt) Explain how copy-on-write operates.

   (d) (3 pt) Do all file systems suffer from internal fragmentation? Why?

(e) (3 pt) What is the context switch time, associated with swapping, if a disk drive with a transfer rate of 2 MB/s is used to swap out part of a program that is 200 KB in size? Assume that no seeks are necessary and that the average latency is 15 ms. The time should reflect only the amount of time necessary to swap out the process.

4. (4 pt) One technique for implementing lottery scheduling works by assigning processes lottery tickets, which are used for allocating CPU time. Whenever a scheduling decision has to be made, a lottery ticket is chosen at random, and the process holding that ticket gets the CPU. The BTV operating system implements lottery scheduling by holding a lottery 50 times each second, with each lottery winner getting 20 milliseconds of CPU time (20 milliseconds × 50 = 1 second). Describe how the BTV scheduler can ensure that higher-priority threads receive more attention from the CPU than lower-priority threads.

5. (6 pt) Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1.

Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?

- LRU replacement
- FIFO replacement
- Optimal replacement

6. (10 pt) If we want to design a carry-save adder to compute the addition of six 4-bit unsigned numbers: A, B, C, D, E, F with **ONLY** 1-bit full adders, and the delay time of the full adder is D_{FA}. Please determine the minimum delay time for this carry-save adder.

7. (15 pt) The following techniques have been developed to reduce cache miss penalty or miss rate: "critical word first and early restart", "pipelined cache access", and "hardware prefetching of instructions". Please briefly explain these techniques and how they work.

8. (15 pt) The critical-section problem is to design a protocol that the processes can use to cooperate. Please answer following questions.
   (1) (5 pt) What are the requirements that a correct solution to the critical-section problem must satisfy?
   (2) (10 pt) Design a correct software solution to the critical-section problem for two processes. Present a justification of the algorithm's correctness.

9. (10 pt) Give an "efficient" solution to the producer-consumer problem. Present a justification of the algorithm's correctness.