1. 可以，TEST AND SET LOCK 與 SWAP 指令兩者行為是相似的。使用 TSL 時，進入 enter_region()所做的行為是將原先的值儲存到暫存器，並且將 lock 設為 1，如果 lock 不等於 0，則代表有其他 process 正在臨界區間，此 process 就必須 busy-waiting。 而 SWAP 指令處理的原理也與 TSL 相同，將傳入的兩值互換，若 lock=True 則代表該 process 進入臨界區間；另一個 process 則必須 busy-waiting。 此外，為了以防暫存器原先儲存的值被其他 process 修改共有變數，必須強調 SWAP 的副程式執行時是不可被中斷且不可分割的。

   When the register's contents get swapped with the memory word, even if it is called by multiple processes nearly simultaneously, whatever value used to be in the register will be swapped into the memory word. The value that was originally in the register can't be lost, since the swapping is done in a "single indivisible action" and can't be interrupted.

2. 因為 CPU efficiency = $\frac{\text{useful CPU time}}{\text{total CPU time}}$

   average process time T(before blocking on I/O)
   process switch time S (effectively wasted)
   round-robin scheduling with quantum Q

   times switched = $\frac{T}{Q}$

   Time wasted switching = $\frac{T}{Q} \times S$

   efficiency = $\frac{T}{T + \text{Time wasted switching}} = \frac{T}{T + \frac{ST}{Q}}$

   a) Q = ∞

      CPU Efficiency = $\frac{T}{T+S}$

   b) Q > T

      CPU Efficiency = $\frac{T}{T+S}$

   c) S < Q < T

      CPU Efficiency = $\frac{T}{T+\frac{ST}{Q}} = \frac{Q}{Q+S}$

   d) Q = S

CPU Efficiency = $\frac{Q}{Q+S}$ = $\frac{Q}{Q+Q}$ = 50%

e) Q ~= 0

CPU Efficiency = $\frac{Q}{Q+S}$ ≒ $\frac{0}{0+S}$ ≒ 0%

3. 與 send 無關，用 process P 建立兩條 threads，
   執行 receive(A,message A); receive(B,message B);
   或 receive(B,message B); receive(A,message A);

4. A=1
   B=1
   C=2
   D=2
   因為 fork()之後，不知道 A、B、C、D 誰先結束，所以順序不一定，有 4!的排列組合。