

1. 可以，因為該替代指令可以只用一個行動（不會被干擾、中斷）使暫存器和記憶體中的值交換，所以只要設定條件就能控制程序進入 critical region：

```
enter_region:
    swap(register,memory)
    set register to 1
    if register is not 0: jump to enter_region
    return
leave_region:
    set register to 0
    return
```

當暫存器和記憶體中的值交換時，若暫存器的值為 0，代表該記憶體可以使用；若暫存器的值不為 0，代表該記憶體正在使用中，不能使用。

2. 
$$\text{CPU 效率} = \frac{nT}{nT + S \frac{nT}{Q}} = \frac{T}{T + S \frac{T}{Q}}$$

T 為程序平均花費時間，S 為轉換的時間，Q 為一次可執行的最長時間

(a)  $Q = \infty$  時，幾乎沒有花費轉換的時間，CPU 效率為 100%

(b)  $Q > T$  時，花費轉換的時間較少，CPU 效率大約為 100%

(c)  $S < Q < T$  時，若 Q 趨近於 S，CPU 效率大約為 50%；若 Q 趨近於 T，CPU 效率大約為 100%（ $50\% \leq \text{CPU 效率} \leq 100\%$ ）

(d)  $Q = S$  時，CPU 效率為 50%

(e) Q 大約為 0 時，幾乎所有時間都花費在轉換，CPU 效率大約為 0

3. 建立 receiveA 和 receiveB 兩條 thread，可以同時進行，也沒有順序問題。

4. A=1

B=1

C=2

D=2