OS-HW2 B08304004S. Date NO.

1. 可以，

因為二者功能相似。使用TSL時進入enter_region做的事是先將原本memory word LOCK中的值複製到暫存器中，並且將LOCK設置為1，再檢查暫存器中LOCK的舊值，如果是非0，表示LOCK已經設置，有其他process正在使用，那該process必須等待反覆試，直到LOCK為0，才能進入critical region.

而swap在此處可以做到類似的事，將暫存器(存1)和memory word LOCK的值互換達到copy和set 1的動作，再判斷舊LOCK的值決定process是否可以進入critical region。並且因為swap是single indivisible action，所以執行過程不會遭阻礙.

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

average process time T.
process switch time S
round-robin scheduling with quantum Q.

→ time switched = $\frac{T}{Q}$.
time wasted switched = $\frac{T}{Q} \times S$.
$\text{efficiency} = \frac{T}{T + \text{Time wasted switching}} = \frac{T}{T + \frac{T}{Q} \times S}$.

a) $Q = \infty$ → CPU不中斷
$\text{CPU efficiency} = \frac{T}{T+S}$

b) $Q > T$
$\text{CPU efficiency} = \frac{T}{T+S}$.

c) $S < Q < T$ → 跑T/Q次，共switch $(T/Q) \times S$ time.
$\text{CPU efficiency} = \frac{T}{T + \frac{T}{Q} \times S} = \frac{Q}{Q+S}$

d) $Q = S$
$\text{CPU efficiency} = \frac{Q}{Q+S} = \frac{Q}{2Q} = 50\%$.

e) $Q \approx 0$
$\text{CPU efficiency} = \frac{Q}{Q+S} \fallingdotseq \frac{0}{0+S} \fallingdotseq 0\%$.

NANKUAN 7.5mm x 31行

3. 建立 receive A 和 receive B 2個 thread. 即可按任一順序接收.

4. A=1
B=1
C=2
D=2

順序上有4!種，因為fork()一次就會有順序上不同可能，再加上沒有waitpid去等待.