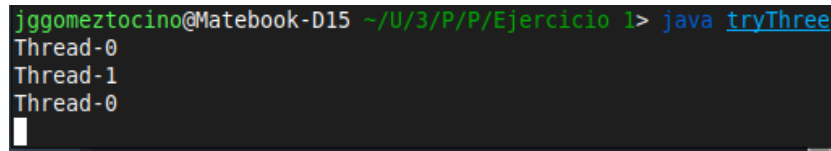


El tercer intento del algoritmo de Dekker introduce una mejora respecto al segundo intento con la reubicación de la condición de espera, incluyéndola en la sección crítica, después, pero igualmente presenta escenarios de interbloqueo, como se demuestra en la siguiente traza:

Proceso P	Proceso Q	wantp	wantq
p1: Sección no crítica	q1: Sección no crítica	false	false
p2: wantp \leftarrow true	q1: Sección no crítica	false	false
p2: wantp \leftarrow true	q2: wantq \leftarrow true	false	false
p2: await wantq=false	q2: wantq \leftarrow true	true	false
p2: await wantq=false	q2: await wantp=false	true	true

En este caso, ambos procesos están intentando entrar en la sección crítica, pero ninguno lo conseguirá.

Este comportamiento es el que precisamente podemos observar en el ciclo de vida de ejecución del *tryThree*, que eventualmente encuentra un interbloqueo y su ejecución se congela.



```
jggomeztocino@Matebook-D15 ~/U/3/P/P/Ejercicio 1> java tryThree
Thread-0
Thread-1
Thread-0
█
```

Figura 1: Interbloqueo encontrado en el tryThree.