

Algoritmos y Programación Paralela

ALGORITMOS NO EFICIENTES

Dra. Ing. Ana Cori Morón

Los esquimales - descripción

- Imaginemos 2 esquimales y un agujero para pescar como recurso compartido. Existe un iglú con un pizarrón. Solo uno de los esquimales puede acceder a la vez al pizarrón. Cuando uno de los esquimales quiere acceder al agujero para pescar debe consultar si tiene permiso para hacerlo en el pizarrón, si en el pizarrón se indica que el permiso lo tiene el otro esquimal espera un tiempo y lo vuelve a probar de nuevo más tarde. Si en el pizarrón se indica que tiene permiso, irá a pescar. Una vez que termine de pescar en el agujero, irá al pizarrón y cederá el permiso al otro esquimal.



ANÁLISIS

Definición de variables, (VARIABLE COMPARTIDA):

estado → libre (el recurso compartido esta libre o disponible para ser usado).

 ↓
ocupado (el recurso compartido esta siendo ocupado por otro proceso)

Definición de procesos

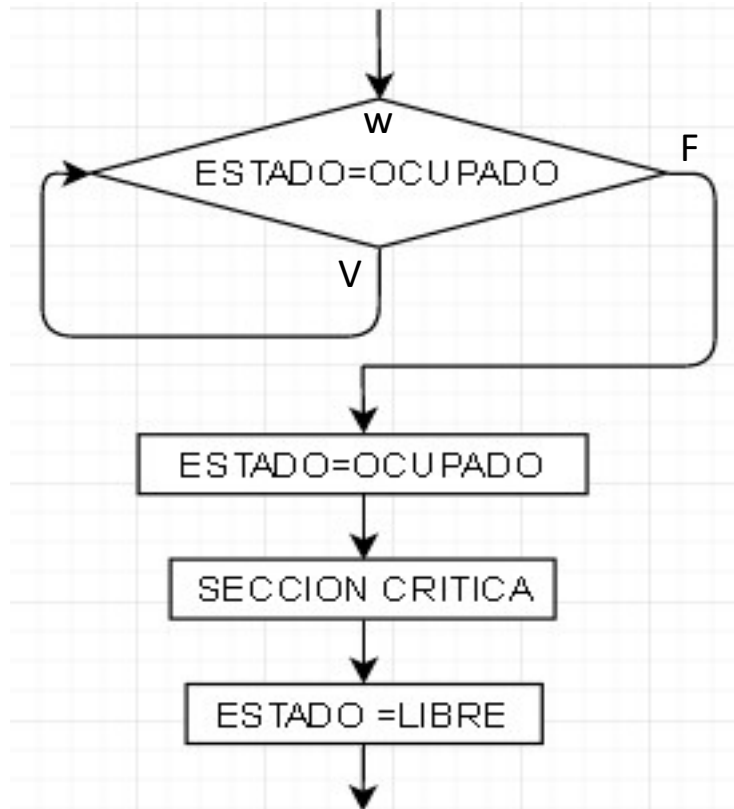
PROCESO P1

PROCESO P2

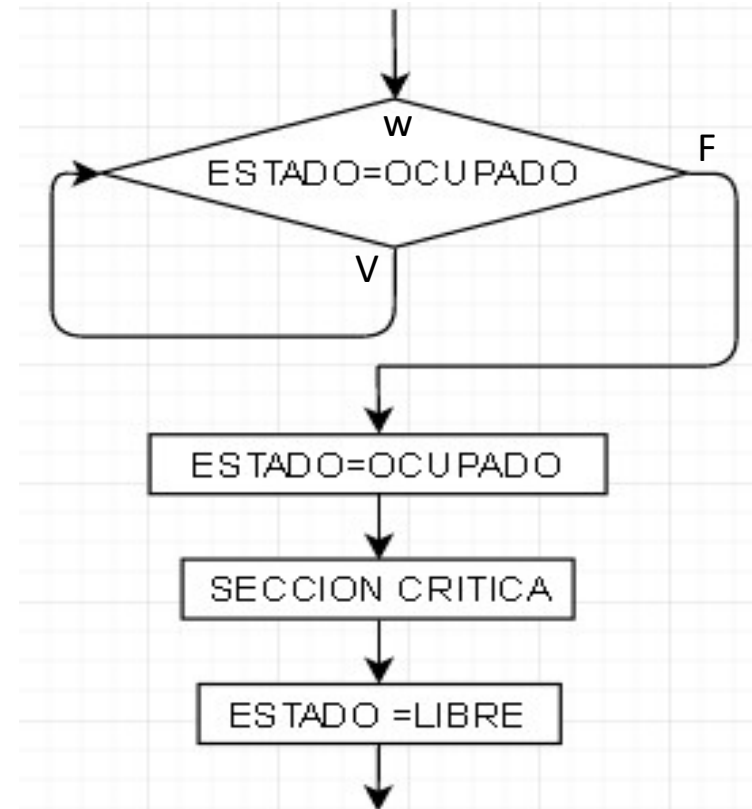
Primer intento

Inicializar: estado = libre

PROCESO P1



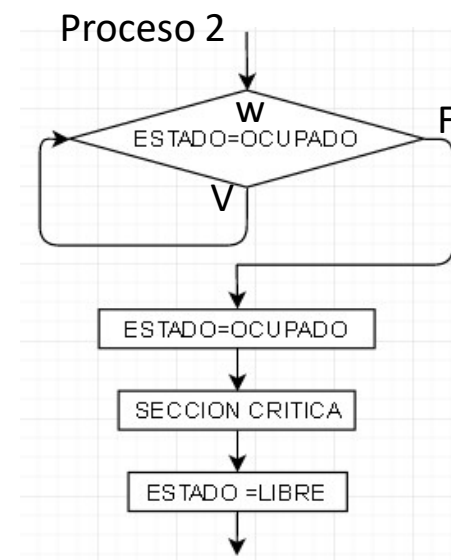
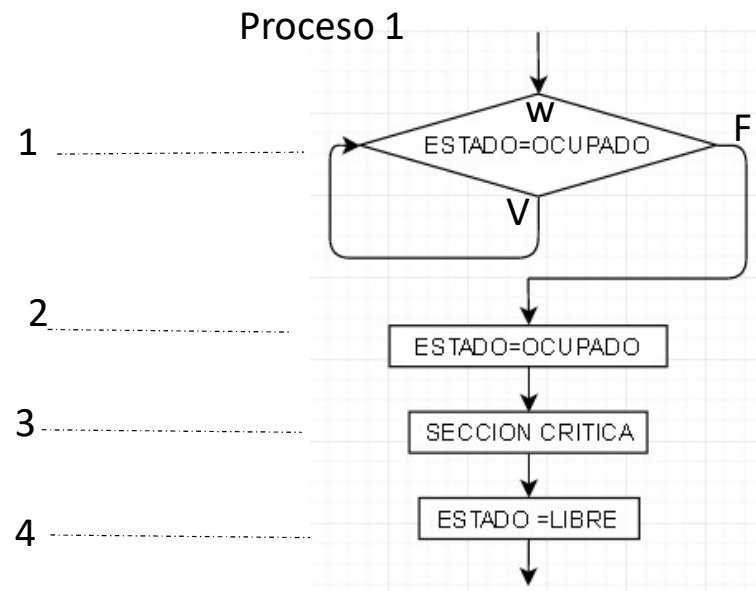
PROCESO P2



Vulnera la condición de exclusión mutua
¿Que sucede si ambos procesos ejecutan el while?

Suponiendo que P1 inicia la ejecución

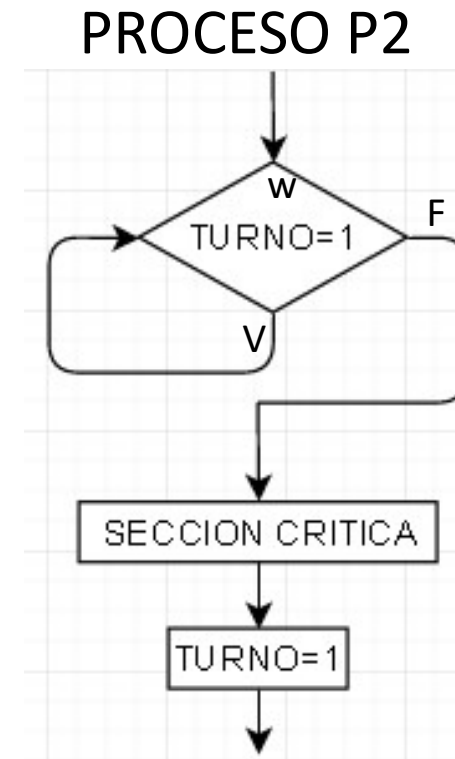
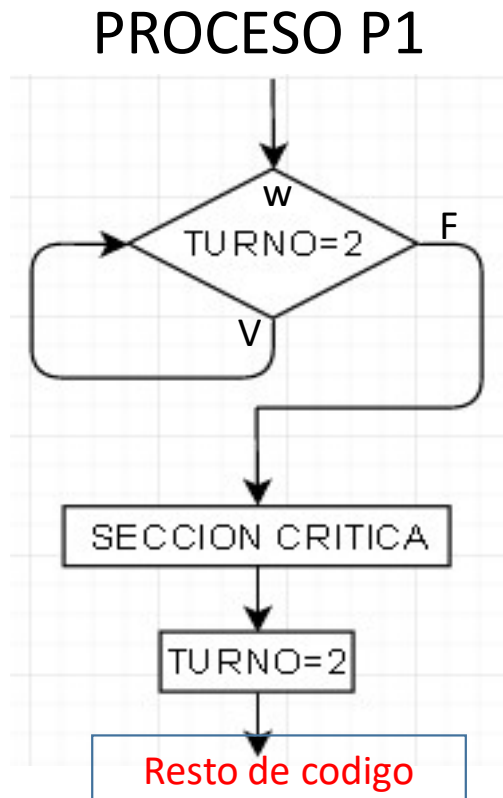
estado	libre		ocupado				Libre		ocu				libre		ocu
P1		1	2		3-SC		4			1		1		1	2
P2				1		1		1	2		3-SC		4		
Tiempo	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14



Suponiendo que P2 inicia la ejecución 

Segundo intento

Turno puede inicializar en 1 ó 2 es indistinto

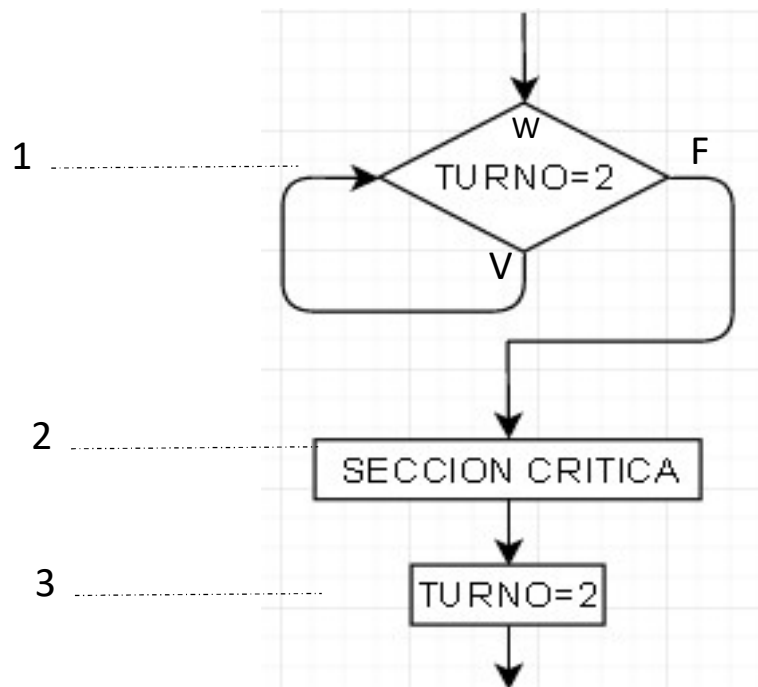


Vulnera la condición de progreso en la ejecución.
El algoritmo garantiza la exclusión mutua, sin embargo, el algoritmo obliga a que la ejecución de ambos procesos sea estricta alternancia: P1-P2-P1,etc... lo cual vulnera la condición de progreso en la ejecución. Además, en el caso que un proceso falle el otro quedará detenido.

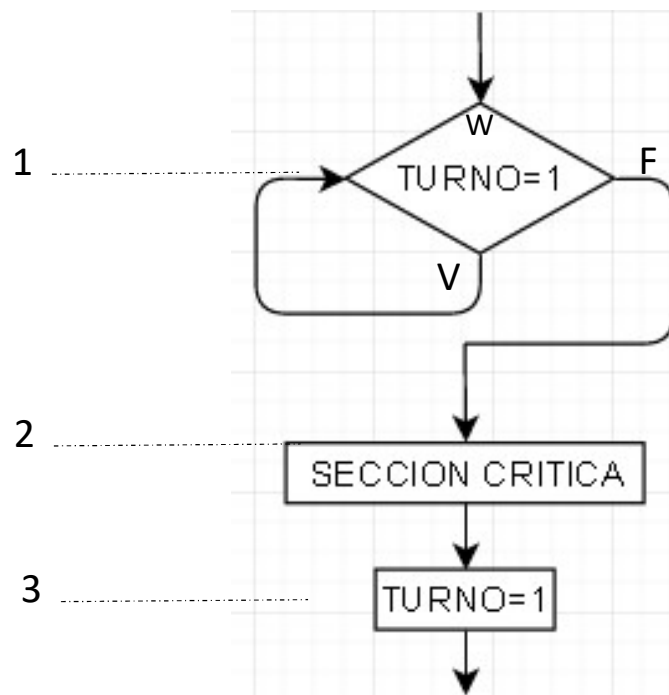
Segundo intento

Turno puede inicializar en 1 ó 2 es indistinto

PROCESO P1



PROCESO P2



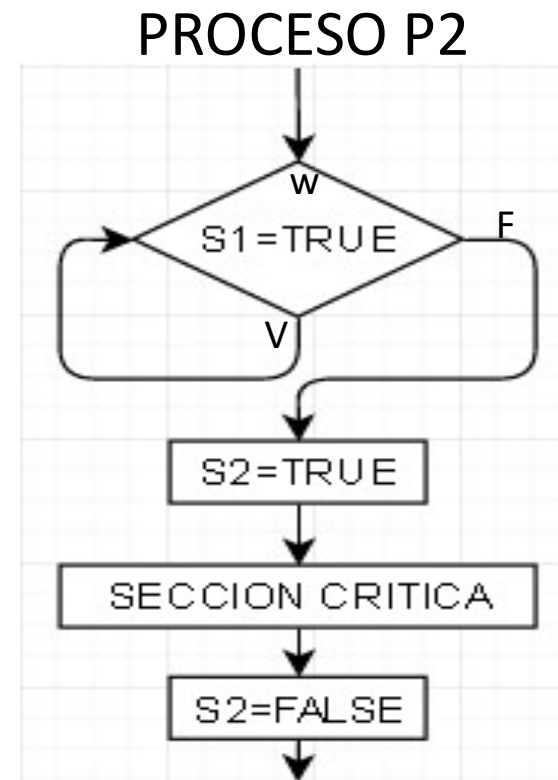
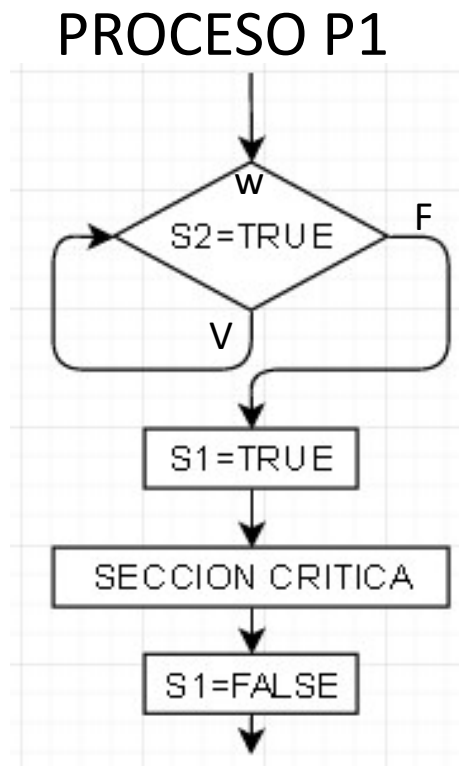
Turno=1

Turno	1				2			
P1		1	2(SC)		3			1
P2				1		1	2(SC)	
Tiempo	T0	T1	T2	T3	T4	T5	T6	T7

¿Turno=2?

Tercer intento

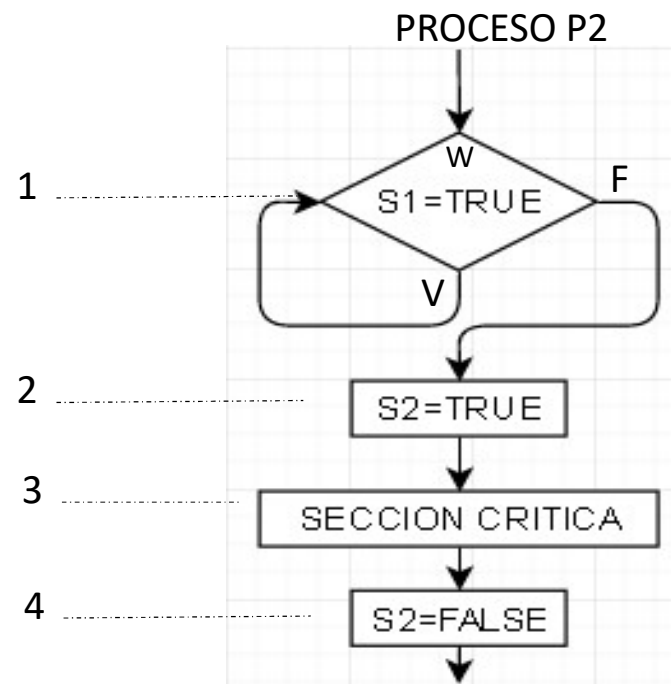
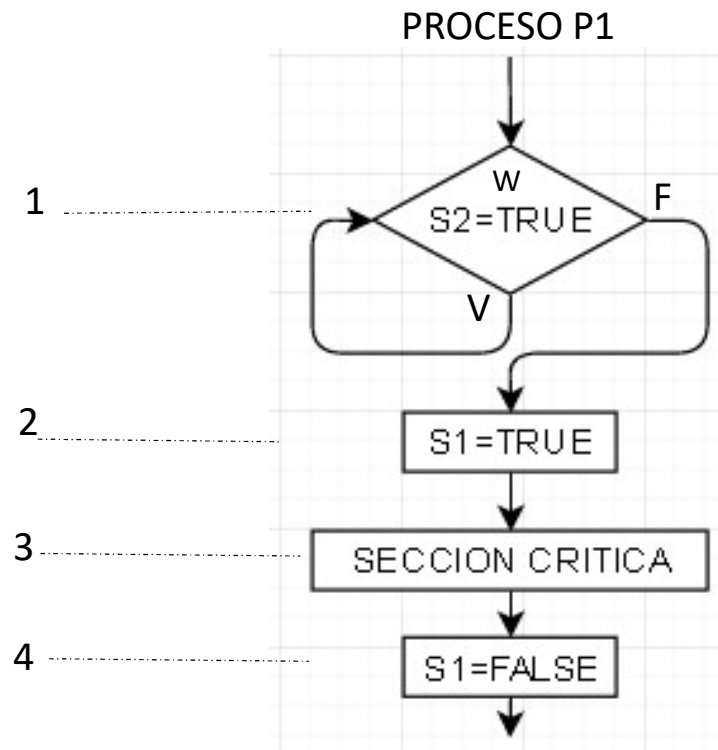
Inicializar $S1=FALSE$ y $S2=FALSE$.



El algoritmo falla en un caso específico cuando los dos procesos evalúan la condición ($S1=TRUE$ y $S2=TRUE$) luego se activan sus respectivas señales ($S1=TRUE$ y $S2=TRUE$), por lo que ambos procesos entrarán a sus secciones críticas. No cumple la condición de exclusión mutua.

Tercer intento

Inicializar
S1=FALSE y
S2=FALSE.



¿Si inicia P2?

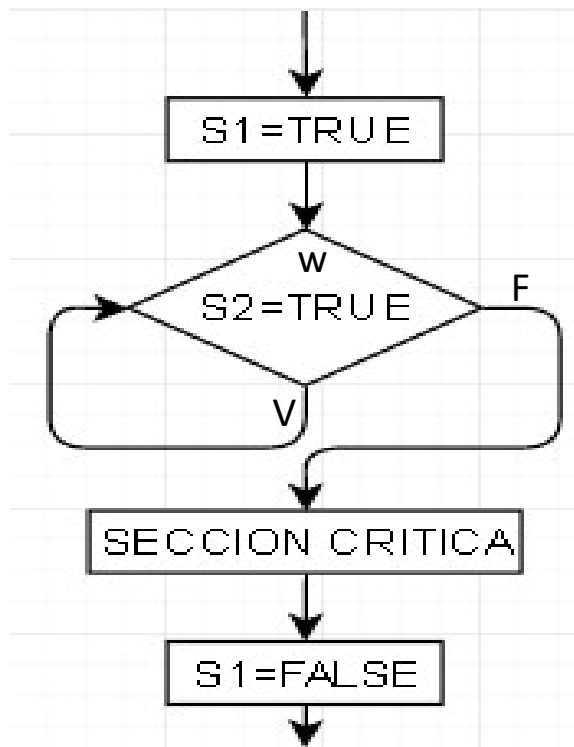
S1	F		T				F		F	F	F		F
S2	F		F				F		T	T	T		F
P1		1	2		3-SC		4			1		1	
P2				1		1		1	2		3-SC		4
Tiempo	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12

Si inicia P1

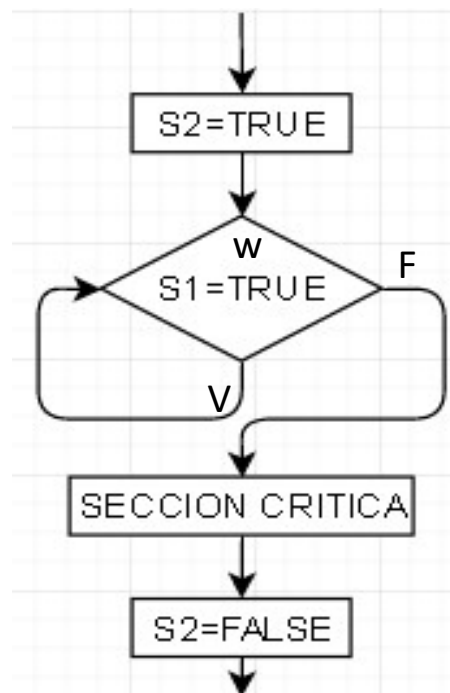
Cuarto intento

Inicializar S1=FALSE y S2=FALSE.

PROCESO P1



PROCESO P2



El algoritmo asegura la exclusión mutua.

Sin embargo, el algoritmo falla si se da la siguiente secuencia de ejecución.

P1 activa su señal (S1=TRUE)

P2 activa su señal (S2=TRUE)

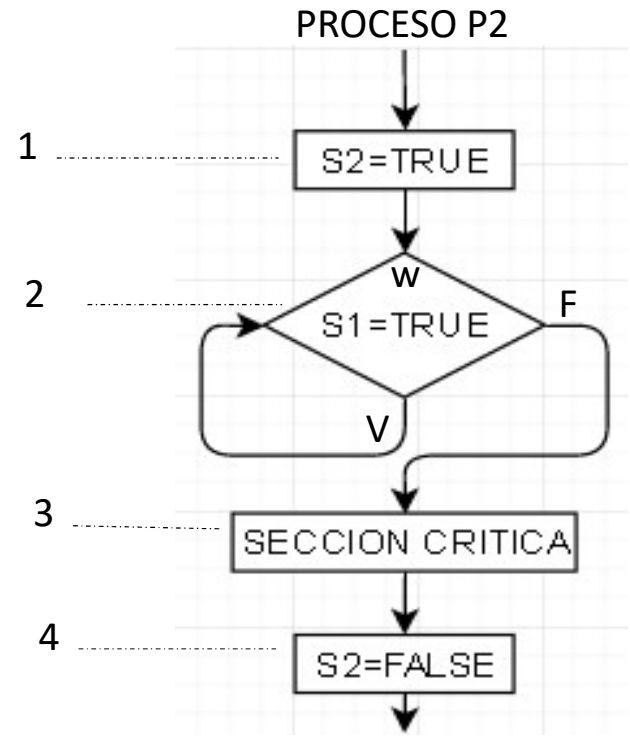
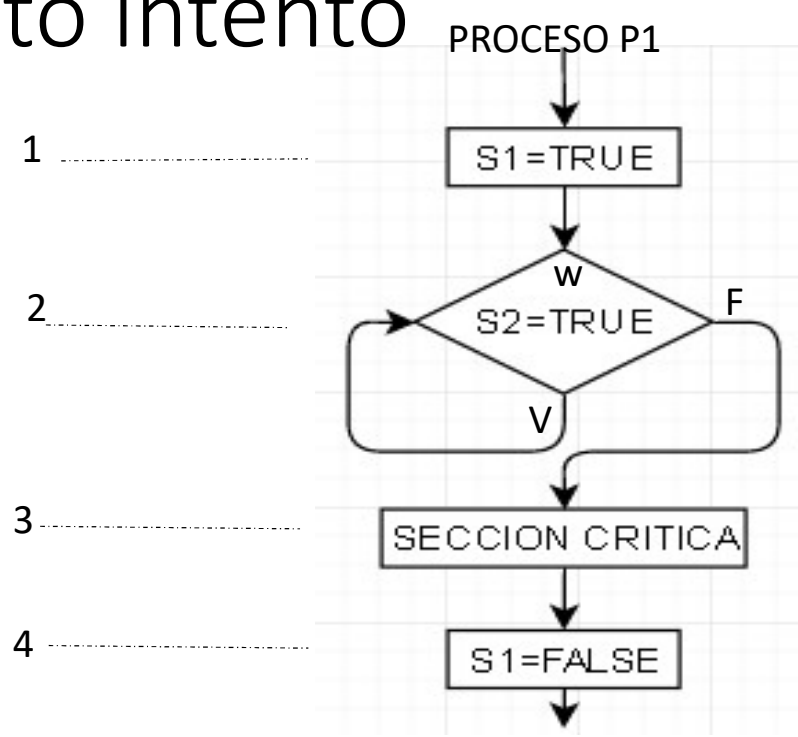
P1 entra en el bucle (S2=TRUE)

P2 entra en el bucle (S1=TRUE)

Error en el progreso de la ejecución.

Livelock (interbloqueo activo)

Cuarto intento



Inicializar
S1=FALSE y
S2=FALSE.

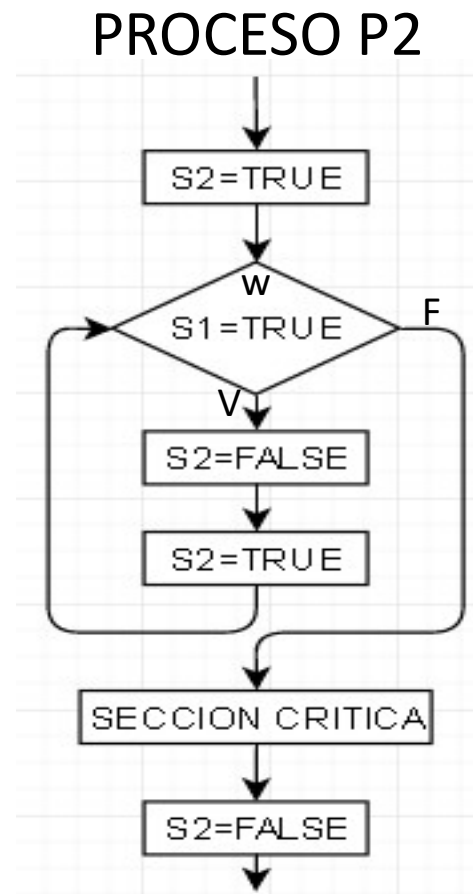
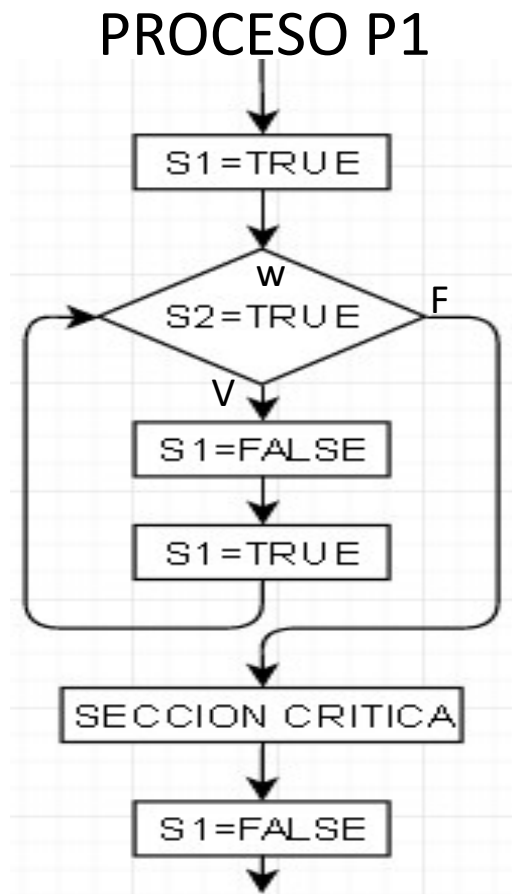
S1								
S2								
P1								
P2								
Tiempo	T0	T1	T2	T3	T4	T5	T6	T7

¿Si inicia P1?

¿Si inicia P2?

Quinto intento

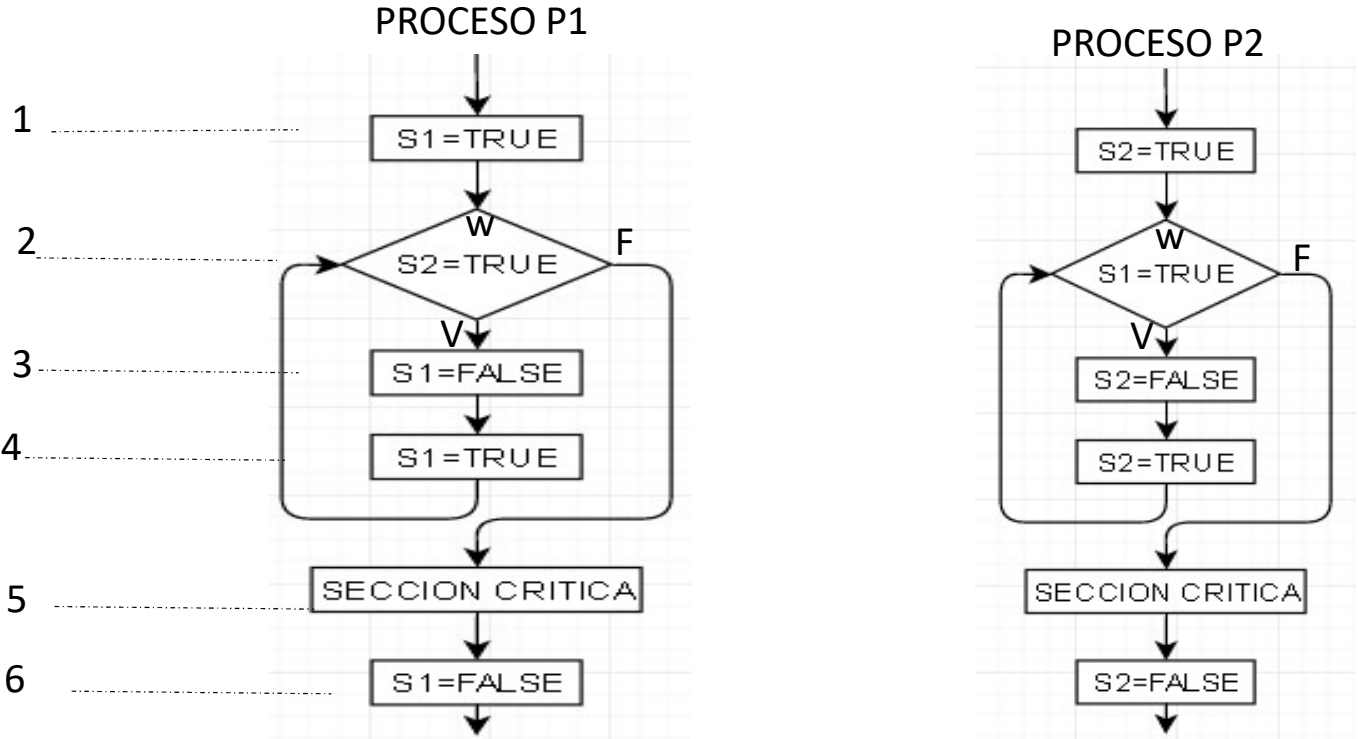
Inicializar S1=FALSE y S2=FALSE.



El algoritmo asegura la exclusión mutua, el tratamiento de cortesía puede llevar a que los procesos se queden de manera infinita cediéndose el paso

Quinto intento

Inicializar
S1=FALSE y
S2=FALSE.



S1								
S2								
P1								
P2								
Tiempo	T0	T1	T2	T3	T4	T5	T6	T7

¿Si inicia P1?

¿Si inicia P2?