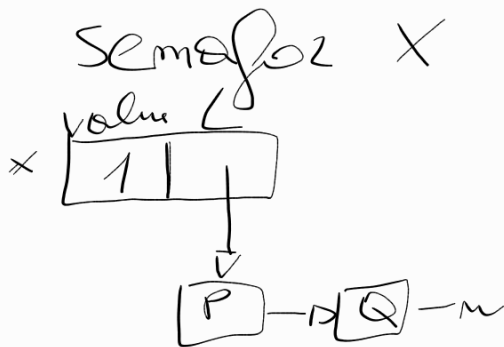


Problema BFW (busy from waiting)

↓ si risolve con

Semaforo (è un tipo di dato, in particolare un record)



valore, intero, inizializzato a 1 per garantire la mutua esclusione
L'è una coda di tipo FIFO

Le operazioni fatte sui semafori sono:

- ① Inizializzazione
 - ② Wait
 - ③ Signal
- } sono tutte sys. call

Var condivise

var mutex: semaphore

Processo P_i :

repeat

wait(mutex)

sezione critica

signal(mutex)

sezione "normale"

until false

wait(S):

S.value := S.value - 1

if S.value < 0

then begin

aggiungi questo processo a S.L

block

end

signal(S):

S.value := S.value + 1

if S.value ≤ 0

then begin

rimuovi un processo P da S.L;

wakeup(P);

end

Wait e signal sono atomiche, e due processi non possono eseguire wait contemporaneamente (non verrebbe garantita la mutua esclusione).

Il semaforo (inizializzato ad altri v) può avere diversi scopi.

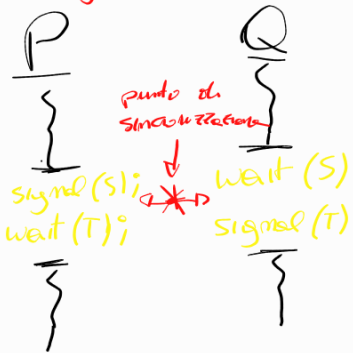
Semaforo per la precedenza

$\beta < 2$ (sezioni di codice)



vogliamo far sì che P venga eseguito prima di Q
 P , quindi, aspetta^(wait), subito prima di Q , che P di Q
 venga eseguito (signal)

Semafor per ottenere un rendez-vous



Voglio che i due processi eseguano insieme quella
 sezione di codice.

Ho bisogno di una coppia di semafori S e T

	value 2
S	0
	value 1
T	0

Il primo processo che arriva deve dare due che è arrivato e aspetta che arrivi l'altro