

Semafori

stato intero non negativo, non può essere letto direttamente;

P aspetta finché il valore non è positivo, lo decrementa atomicamente e restituisce il controllo;

V incrementa atomicamente il valore, e sveglia un thread che sta aspettando con P se necessario.

Mutua esclusione

Semaforo inizializzato a 1, P è acquire e V è release.

Attesa di eventi

Le chiamate a P aspettano una chiamata a V, ma se V è stato chiamato prima di P allora il thread non viene sospeso (come `thread_join`, se il thread è già terminato non va in attesa).

Un'altra differenza rispetto alle CV è che P non rilascia un lock. Perciò, se il semaforo protegge stato condiviso il lock deve essere rilasciato (non atomicamente) prima della chiamata a P, e bisogna assicurarsi che ciò non causi problemi.

I/O

I semafori sono utili per l'attesa del completamento di operazioni di I/O. Kernel e dispositivi operano su un buffer condiviso tramite operazioni atomiche (non si può implementare un lock software), il dispositivo segnala tramite interrupt il completamento delle operazioni, e l'interrupt handler è una funzione molto semplice che si limita a svegliare un thread. Se questo avviene tramite signal, il dispositivo potrebbe generare l'interrupt tra l'ultimo controllo e il wait (visto che non c'è lock), e in questo caso il thread non verrebbe risvegliato.