**Primitivas Bloqueantes:** bloqueiam o processo chamador até que ele seja sinalizado (tipicamente por outro processo)

-> Algumas verificam uma condição e bloqueiam se ela não for satisfeita;

-> Outras bloqueiam incondicionalmente

Ex.:

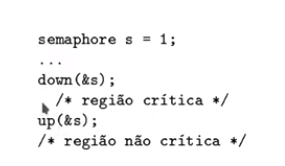
* sleep() e wakeup()
* monitores
* barreiras
* variáveis de condição

**Semáforos:** variável com dois atributos

* um contador
* uma fila de processos bloqueados no semáforo
* quando negativo, o módulo do contador indica quantos processos estão bloqueados em S.

Só pode ser manipulado por:

* **down(S): decrementa S;** se S<0, bloqueia.
* **up(S): incrementa S;** se S<=, acorda um processo que está esperando por S.



Todos semáforos compartilham a mesma variável = **1 semáforo.**

Compartilham diferentes variáveis = **vários semáforos.**

**Semáforos podem ser usados para sincronizar processos (garantir uma sequência desejada de execução.**

* Estrutura geral para o problema:
  + semáforo com valor inicial zero.
  + processo que sinaliza (p1) usa up()
  + processo que espera (p2) usa down()

**Semáforos são implementados pelo núcleo do SO.**

* **não pode inserir se o buffer estiver cheio. (sincronização)**
* **não pode consumir se o buffer estiver vazio. (sincronização)**
* **acessos simultâneos ao buffer estão sujeitos a inconsistências. (exclusão mútua)**

**mutex:** semáforo binário -> garante que apenas um processo acesse o buffer de cada vez.

**full, empty:** semáforos contadores -> empy conta o número de lugares vazios no buffer; full conta o número de lugares preenchidos no buffer.

mutex como primitiva: versão simplificada de semáforos binários: apenas exclusão mútua, sem contagem.

futexes: em espaço de usuário são rápidos quando há pouca disputa de recursos.

* tentam combinar benefícios como:
  + a tentativa de travar um futex ocorre em espaço de usuário.
  + se o futex já estava travado, ocorre uma chamada para o kernel para colocar o processo em uma fila de bloqueados.
  + ao liberar o futex, o processo que o detinha verifica se há processos bloqueados; se houver, avisa ao kernel para desbloquear um deles.

-> kernel só é envolvido quando ocorrer disputa.