Semaforos

Definición de semáforo

- Es un TAD
- Definido por
  - Conjunto de valores que se le pueden asignar
  - Conjunto de operaciones que se le pueden aplicar
- Un semáforo tiene asociada una lista de procesos, en la que se incluyen los procesos suspendidos a la espera de su cambio de estado

Semaforos

Valores de un semáforo

- Semáforos binarios: Pueden tomar solo los valores 0 y 1.
- Semáforos general: Puede tomar cualquier valor Natural (entero no negativo)
- Un semáforo que ha tomado el valor 0 representa un semáforo cerrado, y si toma un valor no nulo representa un semáforo abierto

Semaforos

Operaciones

- wait(p): Si el semáforo no es nulo (abierto) decrementa en uno el valor del semáforo. Si el valor del semáforo es nulo (cerrado), el thread que lo ejecuta se suspende y se encola en la lista de procesos en espera del semáforo.
- signal(p); Si hay algún proceso en la lista de procesos del semáforo, activa uno de ellos para que ejecute la sentencia que sigue al wait que lo suspendió. Si no hay procesos en espera en la lista incrementa en 1 el valor del semáforo

Semaforos

Operacion wait

- Pseudocódigo de la operación:

```
wait(p);
if p>0
then p:= p-1;
else
 “suspende el proceso y lo encola en la lista del
  semáforo”
```

Semaforos

Operacion signal

- Pseudocódigo de la operación:

```
signal(p);
if “hay algún proceso en la lista del semáforo”
then
“activa uno de ellos”
else p:= p+1;
```

Semaforos

- La clave para implementar semáforos es disponer de un mecanismo(lock y unlock) que permita garantizar las secciones críticas de las primitivas wait y signal.

Wait

```
lock
  if s>0
  then s:= s-1;
  else Suspende el proceso;
unlock;
```

Signal

```
lock
  if Hay procesos suspendidos
  then Desbloquea uno de los procesos;
  else  s:= s+1;
unlock;
```

Semaforos

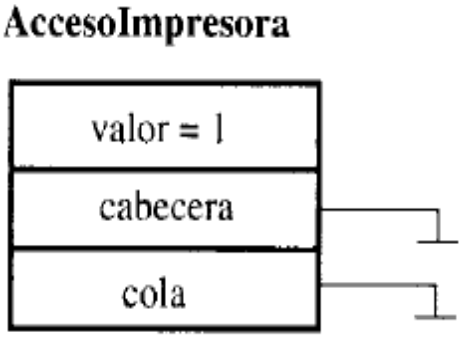
Critica de los semáforos.

- Ventajas de los semáforos:
  - Resuelven todos los problemas que presenta la concurrencia.
  - Estructuras pasivas muy simples.
  - Fáciles de comprender.
  - Tienen implementación muy eficientes.
- Peligros de los semáforos:
  - Son de muy bajo nivel y un simple olvido o cambio de orden conducen a bloqueos.
  - Requieren que la gestión de un semáforo se distribuya por todo el código. La depuración de los errores en su gestión es muy dificil.
- Los semáforos son los componentes básicos que ofrecen todas las plataformas hardware y software como base para construir los mecanismos de sincronización. Las restantes primitivas se basan en su uso implícito.

Semaforos

Ejemplo

1) **Inicializa (AccesoImpresora, 1)**

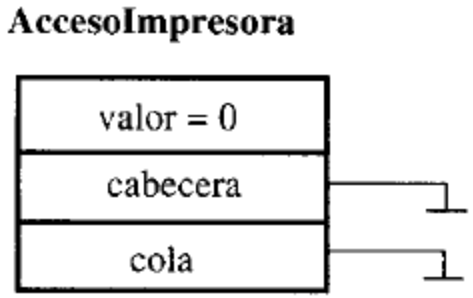


Se inicializa el semáforo AccesoImpresora con el valor 1 de modo que deja libre el acceso a la impresora.

Semaforos

Ejemplo

2) **Proceso P_1 ejecuta: espera (AccesoImpresora)**

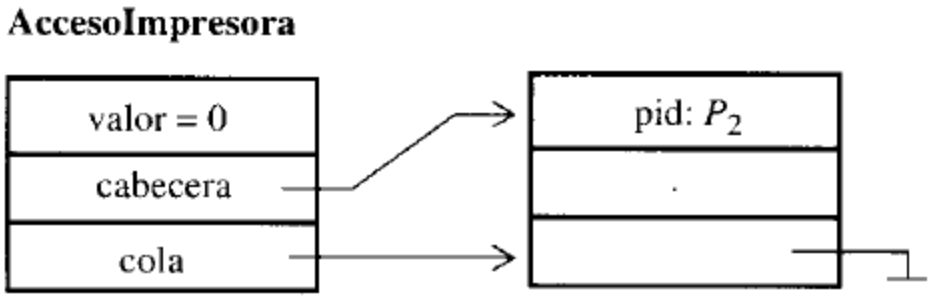


El proceso P_1 toma la impresora e impide el acceso de otro proceso al poner a 0 el valor del semáforo.

Semaforos

Ejemplo

3) **Proceso P_1 bloqueado, Proceso P_2 ejecuta: espera (AccesoImpresora)**

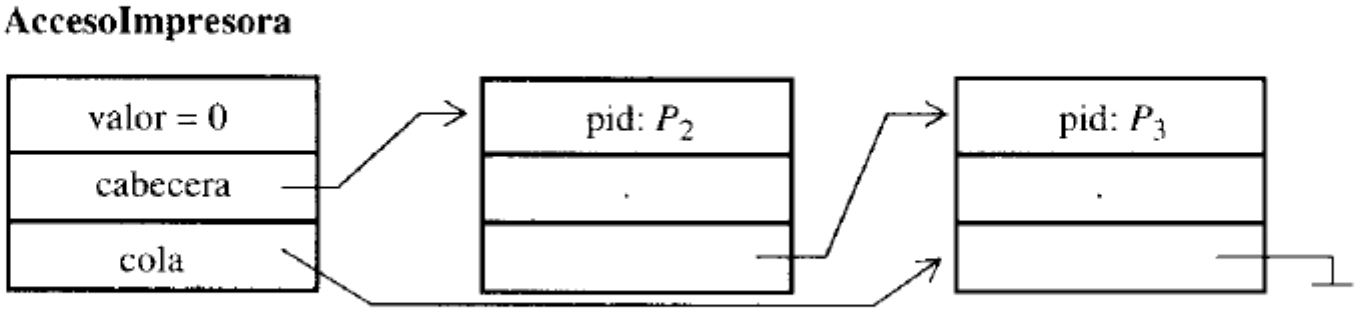


El proceso P_2 intenta acceder al recurso y se suspende, en espera de que éste quede libre, colocándose en la cola del semáforo.

Semaforos

Ejemplo

4) Proceso P_3 ejecuta: espera (AccesoImpresora)

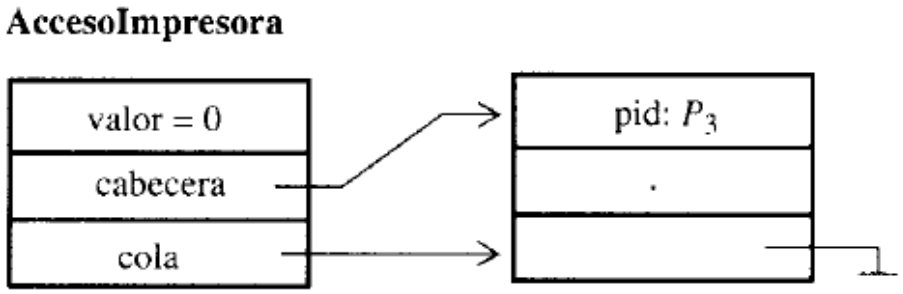


El proceso P_3 intenta acceder al recurso, se suspende y se añade a la cola de espera.

Semaforos

Ejemplo

5) **Proceso P_1 ejecuta: señal (AccesoImpresora)**

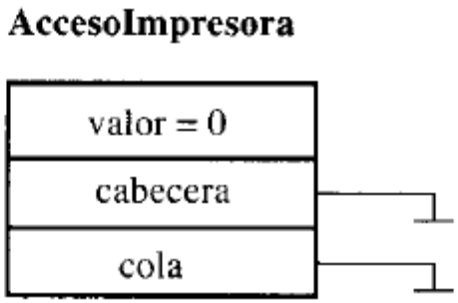


El proceso P_1 deja el recurso; el proceso P_2 pasa al estado preparado y tomará el recurso cuando pase a ejecución.

Semaforos

Ejemplo

6) **Proceso P_2 ejecuta: señal (AccesoImpresora)**



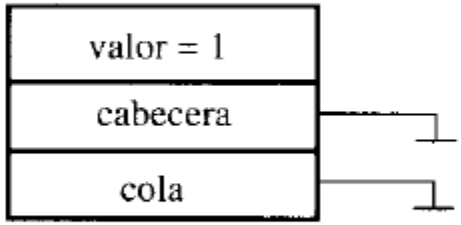
El proceso P_2 deja el recurso; el proceso P_3 pasa al estado preparado.

Semaforos

Ejemplo

7) **Proceso P_3 ejecuta: señal (AccesoImpresora)**

AccesoImpresora



El proceso P_3 deja el recurso. La impresora está disponible para cualquier proceso.

Monitores

Definicion

- Conjunto de procedimientos encapsulados
- Solo un proceso puede estar activo para ejecutar un procedimiento del Monitor
- Para invocar al Monitor, su ejecucion es implicita

Monitores

Proveen...

- Seguridad
- Robustez
- Escalabilidad
- Lenguaje Alto Nivel

Monitores

Señalizacion

- **Signal and Exit**
- **Signal and Continue**

Monitores

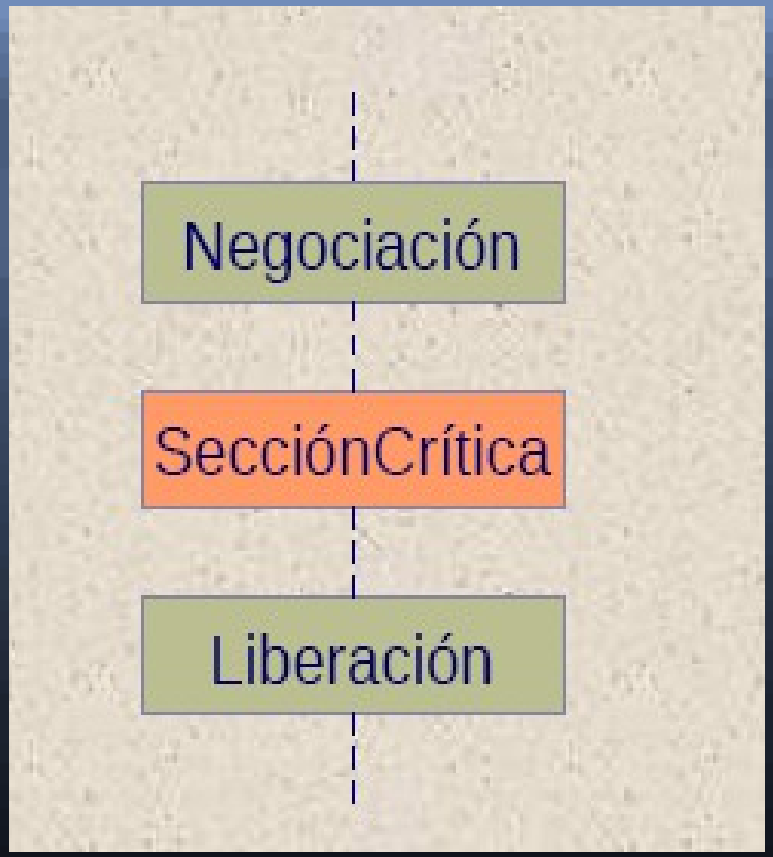
Tipos

- **Monitores de Hoare**
  - **Quien hace Signal cede el cerrojo al del Wait y abandona la CPU. El proceso que había hecho Wait entra en ejecución sobre la marcha. El que hizo Signal tiene que esperar a adquirir de nuevo el cerrojo.**
- **Monitores de Mesa**
  - **Quien hace Signal continúa reteniendo el cerrojo. El proceso que había hecho Wait se pone en la cola de preparados sin ninguna prioridad especial y esperará a adquirir de nuevo el cerrojo (en pugna con los restantes procesos).**

Muchos S.O. lo utilizan por facil implementacion

Semaforos vs Monitores

Semaforos



Monitores

Semaforos vs Monitores

Semaforos

- Recursos globales (SHARED para RCC)
- Código disperso por los procesos
- Las operaciones sobre los recursos no están restringidas

Monitores

- Recursos locales
- Código concentrado
- Los procesos invocan los procedimientos públicos

Memoria

.....proximamente