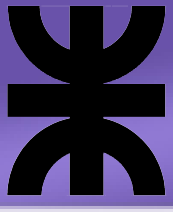
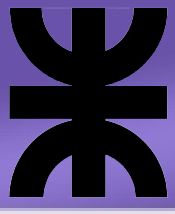


SEMÁFOROS

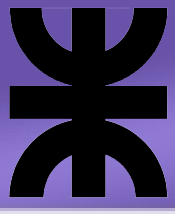


Semáforos

Semáforos

Un semáforo es un número entero positivo que el kernel mantiene, cuyo valor se limita a ser mayor o igual a 0.

- Se utilizan cuando dos o más procesos o hilos quieren acceder a un mismo recurso compartido.
- Es una forma de sincronización para acceder a memoria compartida.
- Un semáforo da acceso al recurso compartido a un solo proceso o hilo por vez.



Semáforos

Semáforos

Operaciones:

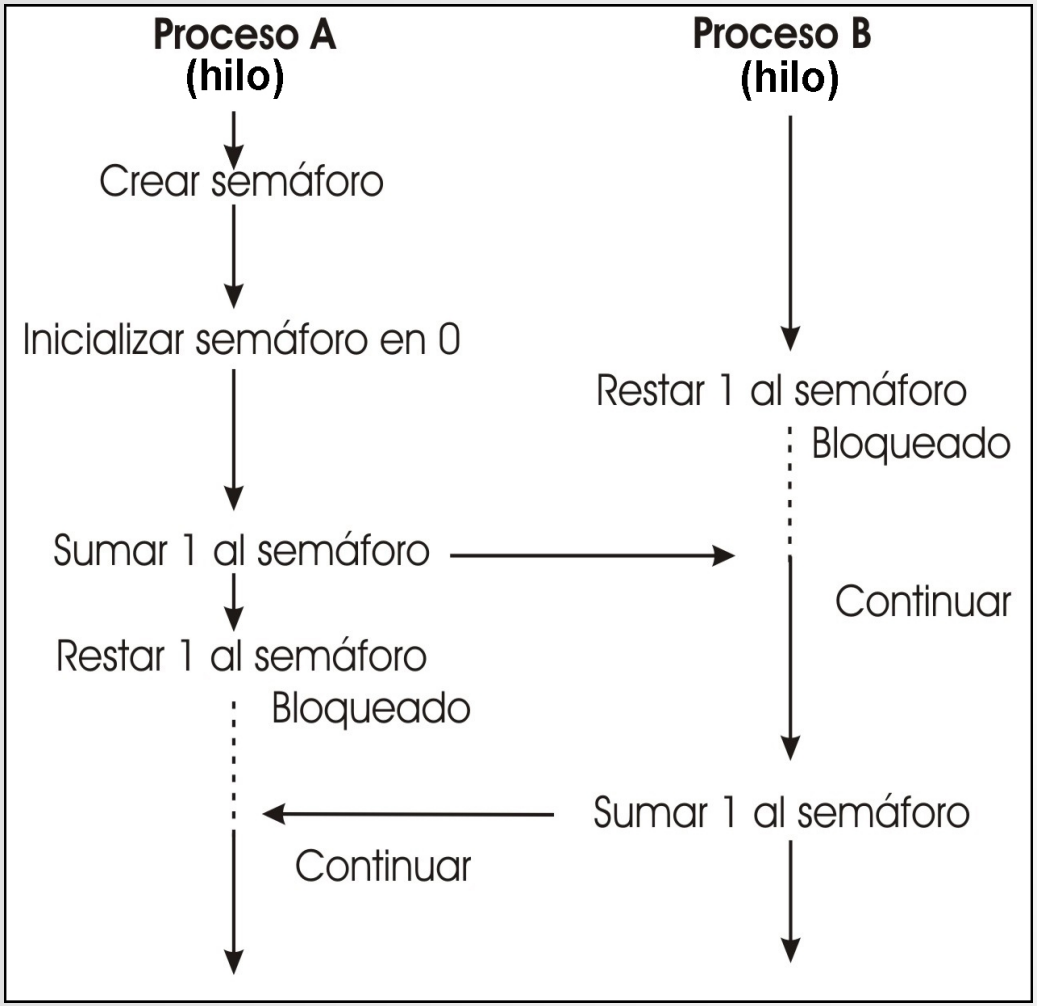
- init: fijar el valor del semáforo en un número entero positivo o cero.
- wait: esperar si el valor del semáforo es igual a 0, si no restar 1 al valor actual del semáforo y continuar.
- post: sumar 1 al valor actual del semáforo.

La operación wait disminuye en 1 el valor del semáforo. El kernel bloquea al proceso (hilo) que la ejecuta si el valor del semáforo es igual 0.

El kernel impide que el valor del semaforo sea menor a 0.

La operación post le suma 1 al semáforo. Puede producir el desbloqueo de otros procesos (hilos) a la espera de que el valor del semáforo sea mayor que 0.

Semáforos



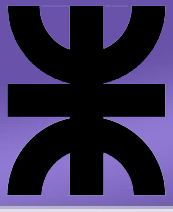
Se crea un semáforo y se inicializa en 0.
Cuando se intenta restar 1 al valor del semáforo, ese proceso (hilo) se bloquea, hasta que otro proceso (hilo) le sume uno y se pueda realizar la resta.

Semáforos

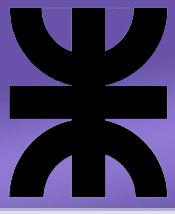
Clasificación:

Semáforos con nombre: Este tipo de semáforos tiene un nombre. Al llamar a sem_open() con el mismo nombre, los **procesos no relacionados** pueden acceder al mismo semáforo.

Semáforos sin nombre: Este tipo de semáforo no tiene un nombre, sino que reside en un lugar acordado en la memoria. Semáforos sin nombre se pueden compartir entre procesos o entre un grupo de hilos. Cuando se comparten entre los procesos, el semáforo debe residir en una región de la memoria compartida. Cuando se comparten entre los hilos, el semáforo puede residir en un área de memoria compartida por los hilos.



SEMÁFOROS CON NOMBRE



Semáforos con nombre

Apertura de un semáforo con nombre

El sem_open() crea y abre un semáforo nuevo con nombre o abre un semáforo existente.

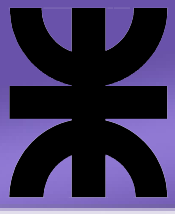
```
#include <semaphore.h>
sem_t *sem;
sem = sem_open(const char *name, int oflag, mode_t mode, unsigned int value);
```

Si el semáforo está creado y solo lo abre se puede simplificar

```
sem = sem_open(const char *name, int oflag);
```

Devuelve el puntero al semáforo si tuvo éxito o SEM_FAILED (-1) en caso de error

Si el semáforo con el nombre name ya existe el campo mode y value se ignoran



Semáforos con nombre

Apertura de un semáforo con nombre

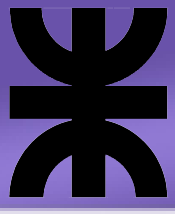
El argumento name: identifica el semáforo.

El argumento mode especifica los permisos con los que se crea el semáforo

El argumento value es un entero sin signo que especifica el valor inicial que será asignado al nuevo semáforo.

El argumento oflag es una máscara de bits que determina si estamos abriendo un semáforo existente, o creando y abriendo un semáforo nuevo.

O_CREAT: `sem_open()` crea un semáforo si no existe. Si existe, accede al semáforo e ignora los parámetros restantes.
O_CREAT | O_EXECL: `sem_open()` crea un semáforo si no existe. Si existe se devuelve -1.
Si oflag es 0 se indica que se quiere acceder a un semáforo creado (si no existe, `sem_open()` devuelve -1).



Semáforos con nombre

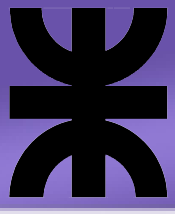
Esperando un semáforo

La función sem_wait() decrementa en 1 el valor del semáforo referido por sem.

```
#include <semaphore.h>
sem_t *sem;
int sem_wait(sem);
```

Devuelve 0 si tiene éxito, o –1 en caso de error

Si el semáforo actualmente tiene un valor mayor que 0, `sem_wait()` vuelve inmediatamente. Si el valor actual del semáforo es 0, `sem_wait()` se bloquea hasta que el valor del semáforo se eleva por encima de 0, en ese momento, el semáforo se decrementa y luego `sem_wait()` vuelve.



Semáforos con nombre

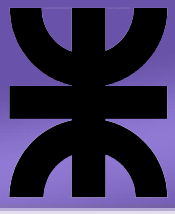
Esperando un semáforo

La función `sem_trywait()` es una versión sin bloqueo de `sem_wait()`.

```
#include <semaphore.h>
sem_t *sem;
int sem_trywait(sem);
```

Devuelve 0 si tiene éxito, o –1 en caso de error

Si la operación de decremento no se puede realizar de inmediato, `sem_trywait()` falla con el error EAGAIN.



Semáforos con nombre

Esperando un semáforo

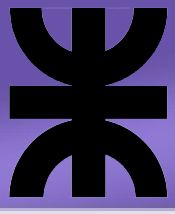
La función `sem_timedwait()` es otra variante de `sem_wait()`. Permite al que llama a la función especificar un limite de tiempo para el cual se bloqueara la llamada.

```
#include <semaphore.h>
sem_t *sem;
int sem_timedwait(sem, const struct timespec *abs_timeout);
```

Devuelve 0 si tiene éxito, o –1 en caso de error

Si la llamada a `sem_timedwait()` termina sin poder decrementar el semáforo, la llamada falla con el error de ETIMEDOUT.

El argumento abs_timeout es una estructura timespec que especifica el tiempo de espera como un valor absoluto en cuestión de segundos y nanosegundos



Semáforos con nombre

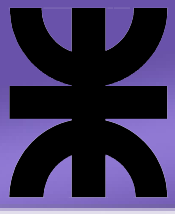
Incrementar un semáforo

La función `sem_post()` incrementa en 1 el valor del semáforo referido por sem.

```
#include <semaphore.h>
sem_t *sem;
int sem_post(sem);
```

Devuelve 0 si tiene éxito, o –1 en caso de error

Si el valor del semáforo era 0 antes de la llamada `sem_post()`, y algún otro proceso (o hilo) está bloqueado en espera para decrementar el semáforo, entonces ese proceso se despierta, y su llamada `sem_wait()` procederá a decrementar el semáforo. Si hay varias procesos (o hilos) bloqueados en `sem_wait()`, entonces, el planificacor determina qué proceso despertará y permitirá incrementar el semáforo.



Semáforos con nombre

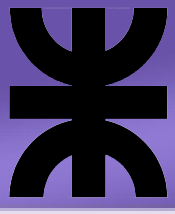
Recuperación del valor actual de un semáforo

La función `sem_getvalue()` devuelve el valor actual del semáforo sem en sval.

```
#include <semaphore.h>
sem_t *sem;
int sem_getvalue(sem, int *sval);
```

Devuelve 0 si tiene éxito, o –1 en caso de error

Si uno o más procesos (o hilos) están bloqueados a la espera de disminuir el valor del semáforo, entonces el valor devuelto en sval depende de la implementación. SUSv3 permite dos posibilidades: 0 o un número negativo cuyo valor absoluto es el número de esperadores bloqueados en `sem_wait()`. Linux y otras implementaciones adoptan el comportamiento anterior.



Semáforos con nombre

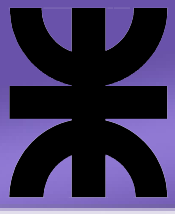
Cierre de un semáforo

Cuando un proceso abre un semáforo con nombre, el sistema registra la asociación entre el proceso y el semáforo.
La función `sem_close()` termina esta asociación (cierra el semáforo), libera todos los recursos que el sistema tiene asociado con el semáforo para este proceso.

```
#include <semaphore.h>
sem_t *sem;
int sem_close(sem);
```

Devuelve 0 si tiene éxito, o –1 en caso de error

Un semáforo previamente creado se cierra al terminar un proceso o si el proceso ejecuta un `close()`. Cerrar un semáforo no implica eliminarlo.



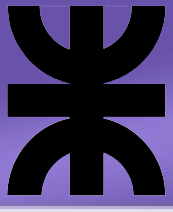
Semáforos con nombre

Eliminación de un semáforo con nombre

La función `sem_unlink()` elimina el semáforo identificado por su nombre y marca el semáforo para ser destruido una vez que todos los procesos dejan de usarlo (esto puede ser inmediatamente, si todos los procesos que tenían el semáforo abierto lo han cerrado).

```
#include <semaphore.h>
int sem_unlink(const char *name);
```

Devuelve 0 si tiene éxito, o –1 en caso de error



Semáforos sin nombre

Bibliografía

Kerrisk, Michael. *The linux programming Interface.* 2011. **Capítulo 53.**