

Semáforos POSIX no nombrados

Los semáforos no nombrados se pueden utilizar por procesos que pertenecen a la misma jerarquía o hilos que pertenecen al mismo proceso. Para utilizarlos es necesario incluir el siguiente archivo de cabecera:

```
#include <semaphore.h>
```

Y declarar los semáforos con el tipo

```
sem_t
```



Inicialización de un semáforos POSIX no nombrado

```
int sem_init(sem_t *sem, int pshared,  
unsigned value);
```

Esta función inicializa el semáforo con el valor `value`. Si el segundo parámetro es:

- `0` el semáforo se puede utilizar solo dentro del proceso
- Algo distinto de `0`, el semáforo se puede utilizar entre procesos que tengan acceso a éste.

La función regresa `0` en caso de éxito y `-1` si hay un error (en cuyo caso regresa también el código de error -variable `errno`-).



Espera, Señal y destrucción de un semáforo POSIX no nombrado

```
int sem_wait(sem_t *sem);
```

Esta función ejecuta un Espera sobre el semáforo.

```
int sem_post(sem_t *sem);
```

Esta función ejecuta un Señal sobre el semáforo.

```
int sem_destroy(sem_t *sem);
```

Esta función destruye el semáforo referenciado. Se puede volver a inicializar y usar.



Ejemplo: Productor-Consumidor con semáforos POSIX no nombrados (archivo ProductorConsumidor.C)

```
#include <pthread.h>
#include <semaphore.h> //Archivo de cabecera

#define TAM_BUFFER 5
#define MAX_PROD 10

void Produce( void *ptr );
void Consume( void *ptr );

sem_t mutex, espacios, productos; //declaración semáforos

int buffer[TAM_BUFFER];
int a_vaciar=0, a_llenar=0;
```

Ejemplo: Productor-Consumidor con semáforos POSIX no nombrados

```
int main()
{
    pthread_t HiloProd, HiloCons;
    srand ( time(NULL) );

    sem_init(&mutex,0,1);
    sem_init(&espacios,0,TAM_BUFFER);
    sem_init(&productos,0,0);

    pthread_create( &HiloProd, NULL, (void *) &Produce, NULL);
    pthread_create(&HiloCons, NULL, (void *) &Consume, NULL);

    pthread_join(HiloCons, NULL);
    pthread_join(HiloProd, NULL);

    sem_destroy(&mutex);
    sem_destroy(&espacios);
    sem_destroy(&productos);

    return 0;
}
```

Se inicializan los semáforos.

Se crean los hilos.

Se espera la terminación de los hilos.

Se destruyen los semáforos.

Ejemplo: Productor-Consumidor con semáforos POSIX no nombrados

```
void Produce( void *ptr ) {
    int i, prod, indice;

    for (i = 0; i < MAX_PROD ; i++)
    {   prod = rand() % 5; // Se genera el producto
        sem_wait(&espacios); // Esperamos que haya espacio
        sem_wait(&mutex); // Se obtiene el derecho de acceso exclusivo al buffer y los índices
        buffer[a_llenar] = prod;
        indice = a_llenar;
        a_llenar = (a_llenar+1) % TAM_BUFFER;
        sem_post(&mutex); // Se libera el derecho de acceso al buffer
        sem_post(&productos); //Se señala que hay un producto disponible

        printf("Productor(%d): produjo Buffer[%d]:%d\n",i,indice,prod);
    }
    pthread_exit(0);
}
```

} Note la generación del producto fuera de la SC

} Note la E/S fuera de la SC

Ejemplo: Productor-Consumidor con semáforos POSIX no nombrados

```
void Consume( void *ptr ) {
    int i, consumible, indice;

    for (i = 0; i < MAX_PROD ; i++)
    {
        sem_wait(&productos); //Esperamos un producto para consumirlo
        sem_wait(&mutex); // Se obtiene el derecho de acceso exclusivo al buffer y los índices
        consumible = buffer[a_vaciar];
        indice = a_vaciar;
        a_vaciar = (a_vaciar+1) % TAM_BUFFER;
        sem_post(&mutex); // Se libera el derecho de acceso al buffer
        sem_post(&espacios);

        printf("Consumidor (%d): Buffer[%d]:%d\n",i,indice,consumible);
        sleep(consumible);
    }
    pthread_exit(0);
}
```

} Note el consumo fuera de la SC