

C Programming in Linux Tutorial #046 – Semaphore pthread

En este tutorial, exploraremos el uso de semáforos en C utilizando pthreads en Linux. Los semáforos son herramientas de sincronización que controlan el acceso a recursos compartidos por múltiples hilos.

¿Qué es un semáforo?

Un semáforo es una variable o abstracto de programación utilizado para controlar el acceso a un recurso común en un entorno de programación concurrente como los hilos (threads). Un semáforo puede inicializarse con un valor no negativo y proporciona operaciones de espera (wait) y señalización (signal).

Bibliotecas necesarias

Para trabajar con semáforos y pthreads en C, necesitamos incluir las siguientes bibliotecas:

```
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

Ejemplo 1: Uso básico de semáforos con pthreads

En este ejemplo, utilizaremos un semáforo para sincronizar dos

hilos. Un hilo incrementará un contador y el otro hilo lo imprimirá.

```
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define NUM_ITERATIONS 10

sem_t semaphore;
int counter = 0;

void* increment(void* arg) {
    for (int i = 0; i < NUM_ITERATIONS; i++) {
        sem_wait(&semaphore); // Espera a que el semáforo
esté disponible
        counter++;
        printf("Increment thread: counter = %d\n", counter);
        sem_post(&semaphore); // Señaliza que el semáforo
está disponible
        sleep(1);
    }
    return NULL;
}

void* print(void* arg) {
    for (int i = 0; i < NUM_ITERATIONS; i++) {
        sem_wait(&semaphore); // Espera a que el semáforo
esté disponible
        printf("Print thread: counter = %d\n", counter);
        sem_post(&semaphore); // Señaliza que el semáforo
está disponible
        sleep(1);
    }
    return NULL;
}

int main() {
```

```
pthread_t inc_thread, print_thread;

// Inicializa el semáforo con un valor de 1
sem_init(&semaphore, 0, 1);

// Crea los hilos
pthread_create(&inc_thread, NULL, increment, NULL);
pthread_create(&print_thread, NULL, print, NULL);

// Espera a que los hilos terminen
pthread_join(inc_thread, NULL);
pthread_join(print_thread, NULL);

// Destruye el semáforo
sem_destroy(&semaphore);

return 0;
}
```

Compilar y ejecutar

Para compilar y ejecutar el programa, usa los siguientes comandos en la terminal:

```
gcc -o semaphore_example example1.c -lpthread
./semaphore_example
```

Este programa crea dos hilos. Uno incrementa un contador y el otro lo imprime. Ambos hilos están sincronizados utilizando un semáforo para asegurar que no accedan al contador al mismo tiempo.

Consideraciones de seguridad

Es importante asegurarse de que los semáforos se inicialicen correctamente y se destruyan después de su uso para evitar fugas de recursos. Además, el uso incorrecto de semáforos puede llevar a condiciones de carrera o bloqueos (deadlocks).

Conclusión

Los semáforos son una herramienta poderosa para la sincronización de hilos en programas concurrentes. Con los ejemplos proporcionados, ahora deberías tener una comprensión básica de cómo usar semáforos con pthreads en C bajo un entorno Linux.