

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree Program in Computer Engineering)

Curso: CE-4302 Arquitectura de Computadores II

(Course: CE-4302 Computer Architecture II)

Estudiante: Gerardo Zeledon Martinez

(Student)

Carnet: 2016243863

(Identification)



Evaluación Taller 1: Hilos

(Workshop 1 evaluation: Threads)

Profesor: Ing. Luis Barboza Artavia

(Professor)

Entrega: 28 de febrero de 2020

(Date)

Investigación:

1. Métodos para el uso de hilos en GNU Linux

POSIX(pthread): Significa Portable Operating System Interface (for Unix). Es un estándar orientado a facilitar la creación de aplicaciones confiables y portables. La mayoría de las versiones populares de UNIX (Linux, Mac OS X) están cumpliendo este estándar en gran medida. Es más efectivo en sistemas multiprocesador o multinúcleo donde el flujo del proceso puede programarse para ejecutarse en otro procesador, ganando así velocidad a través del procesamiento paralelo o distribuido.

2. ¿Que es el concepto de mutex en multiprogramación y que busca hacer?

El término mutex en multiprogramación es la abreviatura a “mutual exclusion” y son la forma más común para implementar sincronización de threads y de proteger datos compartidos cuando deben ser accedidos por muchos hilos.

3. ¿Qué sucede cuando dos hilos quieren utilizar el mismo recurso? ¿Como se manejan estos casos?

Solo lo usa el primer thread y el otro tiene que esperar a que se desocupe. Cando un thread intenta adquirir un mutex que está siendo utilizado por otro thread se pone a “dormir” para no gastar ciclos de CPU inútilmente hasta que el mutex se libera.

El deadlock es el bloqueo permanente de un conjunto de procesos que compiten por los recursos del sistema o bien se comunican unos con otros. A diferencia de otros problemas de la gestión concurrente de procesos.

Problema 2:

Lo que sucede en el problema 2 es que al entrar el hilo 1 se bloquea hasta que el ciclo for termine y el segundo thread, aunque se ejecuta casi a la vez no puede ejecutar la función porque ya está siendo usada por el otro hilo, por lo cual debe esperar a que termine.