Universidad de Carabobo
Facultad Experimental De Ciencias Y Tecnología
Departamento de Computación

Ejercicio 2

Profesora:
Mirella Herrera

Autores:
Luigi Quero, C.I: 30.009.785
Alejandro Cerpa, C.I: 30.334.870
Jeanmarco Alarcon, C.I: 27.117.926

Valencia, junio 2024

El siguiente resumen sintetiza los aspectos clave de cada módulo del programa, destacando los componentes de programación concurrente utilizados, las secciones críticas, los recursos críticos y los escenarios de concurrencia manejados.

Módulo Principal (DireccionMuseo)

- Descripción: Sistema concurrente para simular el flujo de visitantes en un museo, con opciones de audio guía o guía humana.
- Componentes Clave:
    - Hilos para roles como recepcionistas, audio guías, guías humanos y visitantes.
    - Uso de semáforos (semCola) y mutex (pthread_mutex_t) para sincronización.
    - Variables pthread_cond_t para condiciones.
    - Variables booleanas y enteras para estado y disponibilidad de recursos.
- Secciones Críticas: Acceso a colas de espera, modificación de variables globales, acceso a mutex.
- Recursos Críticos: Variables globales, colas de espera, hilos.
- Escenarios de Concurrencia: Atención de visitantes, asignación de guías, gestión de estadísticas.

Módulo Audio Guía

- Descripción: Gestión de audio guías en un museo.
- Componentes Clave:
    - Hilos para audio guía y posibles interacciones.
    - Semaforos y mutex para sincronización y control de acceso a recursos.
    - Condiciones (pthread_cond_t) para espera y notificación.
- Secciones Críticas: Actualización de estado de audio guía libre, manipulación de colas de espera.
- Recursos Críticos: Colas de espera, variables de estado.
- Escenarios de Concurrencia: Atención de visitantes por audio guía, sincronización con otros hilos.

Módulo Guía Humano

- Descripción: Gestión de guías humanos en un museo.
- Componentes Clave:
    - Hilos para guías y posibles interacciones.
    - Semaforos y mutex para sincronización y control de acceso a recursos.
    - Condiciones (pthread_cond_t) para espera y notificación.
- Secciones Críticas: Actualización de estado de guía libre, manipulación de colas de espera.
- Recursos Críticos: Colas de espera, variables de estado.
- Escenarios de Concurrencia: Atención de visitantes por guía humana, sincronización con otros hilos.

Módulo Recepcionista

- Descripción: Gestión de recepcionistas y asignación de guías en un museo.
- Componentes Clave:
    - Hilos para recepcionistas y posibles interacciones.
    - Semaforos y mutex para sincronización y control de acceso a recursos.
    - Condiciones (pthread_cond_t) para espera y notificación.
- Secciones Críticas: Actualización de estado de visitantes, envío de señales de atención.
- Recursos Críticos: Variables de estado, colas de espera.
- Escenarios de Concurrencia: Atención de visitantes, asignación de guías, sincronización con otros hilos.

Módulo Visitante

- Descripción: Procesamiento de visitantes en un museo.
- Componentes Clave:
    - Hilos para visitantes y sus interacciones.
    - Semaforos y mutex para sincronización y control de acceso a recursos.
    - Condiciones (pthread_cond_t) para espera y notificación.
- Secciones Críticas: Acceso y modificación de variables compartidas, sincronización de hilos.
- Recursos Críticos: Variables de estado, colas de espera.
- Escenarios de Concurrencia: Interacción de visitantes con recepcionistas y guías, espera y asignación de recursos.

Módulo Time

- Descripción: Simulación del flujo de visitantes a lo largo del día en un museo.
- Componentes Clave:
    - Proceso para simular el paso del tiempo y actualizar estadísticas.
    - Hilos para tareas paralelas como contar visitantes y actualizar estadísticas.
- Secciones Críticas: Acceso y modificación de variables compartidas, sincronización de hilos.
- Recursos Críticos: Variables de estado, colas de espera.
- Escenarios de Concurrencia: Conteo de visitantes, actualización de estadísticas, generación de informes.

Al realizar el programa descrito, hemos adquirido una comprensión profunda de varios conceptos y técnicas clave en programación concurrente y sistemas operativos, especialmente en el contexto de C.

Programación Concurrente

- Hilos y Threads: Hemos trabajado con hilos (threads) en C usando la biblioteca POSIX threads (pthread). Esto implica crear, sincronizar y comunicarse entre hilos, lo cual es fundamental para la programación concurrente.
- Sincronización: Hemos utilizado semáforos (semaphores), mutex (mutexes) y condiciones (condition variables) para sincronizar el acceso a recursos compartidos y coordinar la ejecución de hilos. Esto es crucial para evitar condiciones de carrera y asegurar la integridad de los datos en un entorno concurrente.
- Comunicación entre Hilos: Hemos experimentado cómo los hilos pueden comunicarse entre sí mediante variables compartidas y señales de condición. Esto nos ha enseñado sobre los mecanismos de sincronización y comunicación necesarios para coordinar acciones entre hilos.

Manejo de Recursos Compartidos

- Acceso Control: Hemos aprendido a usar mutex para proteger secciones críticas del código donde se acceden o modifican recursos compartidos. Esto es vital para prevenir condiciones de carrera y garantizar la consistencia de los datos.
- Gestión de Colas: Hemos implementado y utilizado colas para gestionar visitantes y guías, lo que nos ha dado experiencia en estructuras de datos y su uso en contextos de programación concurrente.

Diseño de Sistemas Concurrentes

- Modelado de Sistemas: Hemos diseñado un sistema simulado basado en el flujo de visitantes en un museo, lo que nos ha ayudado a entender cómo abstraer y modelar sistemas complejos.
- Patrones de Diseño: Aunque no se mencionan explícitamente, nuestro trabajo implica la aplicación de patrones de diseño concurrentes, como Productor-Consumidor, para organizar la lógica del mismo.

Estructura y Organización del Código

- Modularidad: Nuestro programa está dividido en módulos claramente definidos, cada uno encargado de una parte específica de la funcionalidad. Esto demuestra la importancia de la modularidad en el desarrollo de software.
- Encapsulación: Algunas partes del código, como las funciones de hilos, están encapsuladas en archivos separados, siguiendo buenas prácticas de organización y mantenimiento del código.

Experiencia Práctica

- Depuración y Optimización: Al enfrentarnos a problemas comunes en la programación concurrente, como deadlocks y condiciones de carrera, hemos adquirido habilidades prácticas en depuración y optimización de código.
- Resolución de Problemas: La necesidad de resolver problemas específicos relacionados con la sincronización y la gestión de hilos nos ha permitido mejorar nuestras habilidades de pensamiento analítico y resolución de problemas.