Tarea 2 - Sistemas Operativos

Lucas Escalona Samson

Item 1

Productor y consumidor

Para el desarrollo de este programa se modeladon los productores como generdores valores enteros aleatorios en el rango de 0 a 99. Estos valores representan los productos que serán almacenados temporalmente en el buffer. Los consumidores se modeladon como agentes procesan estos productos calculando su raíz cuadrada, lo cual simula el "consumo" del producto.

Monitor

El monitor está impremendo como una clase que gestiona el acceso concurrente al buffer. Este lleva un conteo de los productos presentes en el buffer a través del atributo contador. Además, los índices in y out indican las posiciones de inserción y extracción dentro del buffer, respectivamente. Estos índices se ajustan en cada operación de producción y consumo.

El acceso al buffer está protegido mediante un mutex que evita que varios hilos accedan al buffer al mismo tiempo. Para gestionar la espera de los hilos en condiciones donde el buffer esté lleno o vacío, se utiliza una variable de condición (cv). Los productores esperan si el buffer está lleno y los consumidores esperan si el buffer está vacío. Una vez que un hilo realiza una acción (ya sea producir o consumir), se notifica a otros hilos sobre el cambio en las condiciones, permitiendo que reanuden su ejecución.

El redimensionamiento está implementado como indica el enunciado de la tarea. Durante este, los productos ya presentes en el buffer se copian a un nuevo arreglo con el tamaño adecuado, y los índices in y out se ajustan para mantener la integridad del buffer circular.

Runtime

1) Los valores clave para la simulación (número de productores, consumidores, tamaño inicial del buffer y tiempo de espera de los consumidores) se obtienen de los argumentos pasados al ejecutar el programa.

2) Se inicializan vectores para almacenar los hilos de productores y consumidores.

3) Se crean múltiples hilos productores, cada uno encargado de generar un número fijo de productos.
4) Paralelamente, se crean hilos consumidores, que intentan consumir productos del buffer hasta que se agoten.

Ejecución de los hilos

Los hilos productores y consumidores operan en paralelo, coordinándose mediante el Monitor. Los productores insertan productos en el buffer mientras que los consumidores los extraen, sincronizándose a través de mutexs y variables de condición.

Una vez que todos los productores y consumidores completan su trabajo, se llama a join() en cada hilo para asegurarse de que todos terminen correctamente.
Finalmente, el programa se cierra, liberando recursos y cerrando el archivo de log.

Item 2

Modelo de Páginas y Estructura del Sistema

El sistema simula la gestión de memoria virtual, implementando algoritmos de reemplazo de páginas (FIFO, LRU, Reloj simple y Óptimo). Las páginas representan bloques de datos que deben mantenerse en un número limitado de marcos de memoria física. Los fallos de página ocurren cuando una página solicitada no está en memoria, lo que requiere cargarla y, en algunos casos, reemplazar otra.

Tabla de Páginas

La clase Tabla_de_paginas es el núcleo de la simulación. Esta estructura organiza y controla las páginas presentes en memoria, garantizando un manejo eficiente según el algoritmo de reemplazo seleccionado. A continuación, se detallan sus componentes principales:

Estructura Interna

Hash Table (tabla): Almacena las páginas actualmente en memoria para acceso rápido.

Listas o Vectores de Control:
FIFO y LRU: Usan una lista (orden_de_acceso) para registrar el orden de inserción o acceso.

Reloj Simple: Gestiona los marcos mediante un vector (marcos) y un puntero (puntero_reloj).

Métodos de Reemplazo

FIFO: Reemplaza la página que ingresó primero.
Óptimo: Busca la página cuyo uso está más lejano en el futuro.
LRU: Reemplaza la página menos recientemente utilizada.
Reloj Simple: Evalúa los bits de referencia para reemplazar páginas de forma cíclica.

Control de Fallos de Página

Cada vez que una página no está en memoria y se debe cargar, el sistema registra un fallo de página.

Simulación y Algoritmos

Inicialización

Se obtienen los parámetros de entrada: número de marcos (-m), algoritmo (-a), y archivo con referencias (-f). Las referencias de página se leen desde el archivo especificado y se almacenan en un vector para su procesamiento.

Simulación de Accesos a Memoria

Cada referencia de página se evalúa. Si ya está en memoria, se actualizan estructuras internas del algoritmo de reemplazo correspondiente. Si no está en memoria, ocurre un fallo de página, y se utiliza el algoritmo seleccionado para determinar qué página reemplazar.

Repositorio: https://github.com/Escalona1/SistemasOperativos2024-2