

Práctica: Simulación de TreadMarks Distribuido con Docker

Introducción

En esta práctica se desarrolló una simulación de TreadMarks distribuido usando contenedores Docker. TreadMarks es una implementación de memoria compartida distribuida (DSM, por sus siglas en inglés), que permite a múltiples procesos cooperar sobre estructuras compartidas sin necesidad de un sistema de memoria física unificada. Esta técnica es útil en sistemas distribuidos, ya que facilita la programación paralela sin tener que manejar la complejidad completa de paso de mensajes.

Objetivos

- Simular el comportamiento de TreadMarks usando memoria distribuida coordinada con sockets TCP.
- Implementar mecanismos de sincronización como barreras y exclusión mutua distribuida.
- Ejecutar múltiples nodos (coordinador y workers) en un entorno contenido y aislado con Docker.

Actividades Realizadas

1. Se desarrolló una biblioteca (`tmk_lib.c`) que proporciona funciones básicas para:
 - Inicialización distribuida (`Tmk_startup`)
 - Barrera de sincronización (`Tmk_barrier`)
 - Distribución de estructuras (`Tmk_distribute`)
 - Exclusión mutua con un servidor central (`Tmk_lock_acquire`, `Tmk_lock_release`)
2. Se crearon dos programas:
 - `nodo.c`: valida barreras y lock distribuido.
 - `suma_tmk.c`: distribuye una estructura con un arreglo y suma cooperativa entre nodos.
3. Se diseñaron un Dockerfile, Makefile y docker-compose.yml para construir automáticamente los contenedores de cada nodo con su propio ID de proceso. La red definida permite comunicación directa entre ellos.

Resultados

Al ejecutar el entorno, se observaron los siguientes comportamientos correctos:

- Los nodos esperan al coordinador mediante una barrera TCP.
- Se distribuye correctamente una estructura de datos compartida.

- Cada nodo calcula una suma parcial del arreglo y la envía de forma segura al coordinador.
- El coordinador acumula las sumas y presenta un resultado correcto de la suma global.

Conclusión

Esta práctica permitió comprender los fundamentos de TreadMarks como sistema de memoria distribuida y cómo simular su comportamiento en entornos reales con Docker (no tuvimos acceso al clúster del labred). Se exploraron conceptos importantes como sincronización, exclusión mutua, y transmisión de estructuras entre procesos. El uso de herramientas modernas como Docker facilitó la creación de un entorno distribuido.