

IPC y Sockets

Niurka Vanesa Yupanqui

04 de julio de 2025

Introducción

Como parte de la tarea asincrónica sobre **IPC y sockets**, realicé la ejecución de distintos programas incluidos en las carpetas proporcionadas. Esta actividad me permitió observar de manera práctica cómo funciona la comunicación entre procesos, la creación de procesos hijos y la sincronización entre hilos.

Exploración de carpetas y programas

Carpeta `procesos`

Ejecuté el archivo `fork.c`, el cual utiliza la función `fork()` para crear procesos hijos. Al compilarlo y ejecutarlo, noté cómo el mismo código se ejecuta en paralelo tanto en el proceso padre como en el hijo.

También trabajé con `sumSec.c` y `sumPar.c`, que permiten comparar la ejecución secuencial con una paralela utilizando múltiples procesos.

Carpeta `cinco/`

Ejecuté el archivo `cinco.c`, una implementación del problema de los cinco filósofos. Este ejemplo me permitió analizar cómo se gestiona la sincronización entre procesos que comparten recursos, y cómo se evita el bloqueo o el acceso simultáneo no controlado.

Carpeta `threads/`

Trabajé con el archivo `tres_threads.c`, donde observé cómo se crean varios hilos que se ejecutan al mismo tiempo dentro de un mismo proceso. Para su compilación utilicé la librería `-lpthread`. También observé cómo los hilos imprimen mensajes de forma concurrente, mostrando un ejemplo claro de paralelismo.

Reflexión final

Esta práctica me ayudó a comprender mejor los conceptos de procesos, hilos y concurrencia. Me familiaricé con herramientas como `gcc`, `make` y el uso de bibliotecas para programación con hilos. A través de los ejemplos ejecutados, pude ver la importancia de la sincronización y la forma en que los procesos se comunican y trabajan en conjunto en un entorno de sistemas operativos.

Niurka Vanesa Yupanqui