

Entregable Trabajo Práctico N° 3

Tomás Vidal

Sistemas Operativos y Redes

Facultad de Ingeniería, UNLP, La Plata, Argentina.

30 de Octubre, 2024.

I. PROBLEMA PRESENTADO

Se debe crear un programa que deba ejecutar *dos hilos*: **hilo A** e **hilo B**. Ambos deben poder acceder a un *bloque de memoria compartida*, para poder leer y escribir datos en común (*un arreglo de 20 bits*). Ambos procesos deben mantener una cierta sincronía, ya que uno no puede leer mientras el otro escribe y viceversa.

II. ALGORITMO IMPLEMENTADO

En la figura 1 se muestra el flujo general del programa. El diagrama es *abierto*, en sentido de que hay caminos en paralelo, esto se debe a que el programa tiene un comportamiento concurrente, debido a que se ejecutan múltiples hilos.

III. FUNCIONES PRINCIPALES

- *handle_termination*: Manejador que se encarga de liberar los recursos compartidos.
- *handle_thread_a*: Realiza la escritura y lectura acorde a las especificaciones del **Hilo A**.
- *handle_thread_b*: Realiza la escritura y lectura acorde a las especificaciones del **Hilo B**.
- *cypher*: Función de encriptación de datos.
- *decypher*: Función de desencriptación de datos.

IV. EXPLICACIÓN GENERAL DEL ALGORITMO

Se registra el manejador para la señal **SIGTERM**. Luego se crean el *mutex*, *cond_a* y *cond_b*, los hilos *A* y *B*; y se ejecutan los hilos, que quedan en un bucle de ejecución de tiempo indefinido, realizando las escrituras y lecturas acordes a las especificaciones, la sincronización se logra con el *mutex* y las condiciones compartidas entre los hilos (*cond_a* y *cond_b*). Eventualmente cuando la señal **SIGTERM** se reciba, se cancelan los bucles de los hilos, y luego de que se terminan las tareas que están realizando finalizan el bucle, se liberan los recursos y finaliza el programa.

Al comienzo se definen los 20 **int**, que conforman las palabras 'holamundooholaamundo', encriptadas con las especificaciones del problema (es decir son *números del 0 al 26*, que representan las letras de la **a** a la **z**), posteriormente en la escritura de ambos procesos, se escriben estos datos nuevamente encriptados con la función provista $f(x)$, y cuando se leen se desencriptan con la función $f^{-1}(x)$

V. REPOSITORIO DE GITHUB

El trabajo se puede encontrar en el siguiente [repositorio de Github](#).

Para acceder al mismo se deben tener permisos

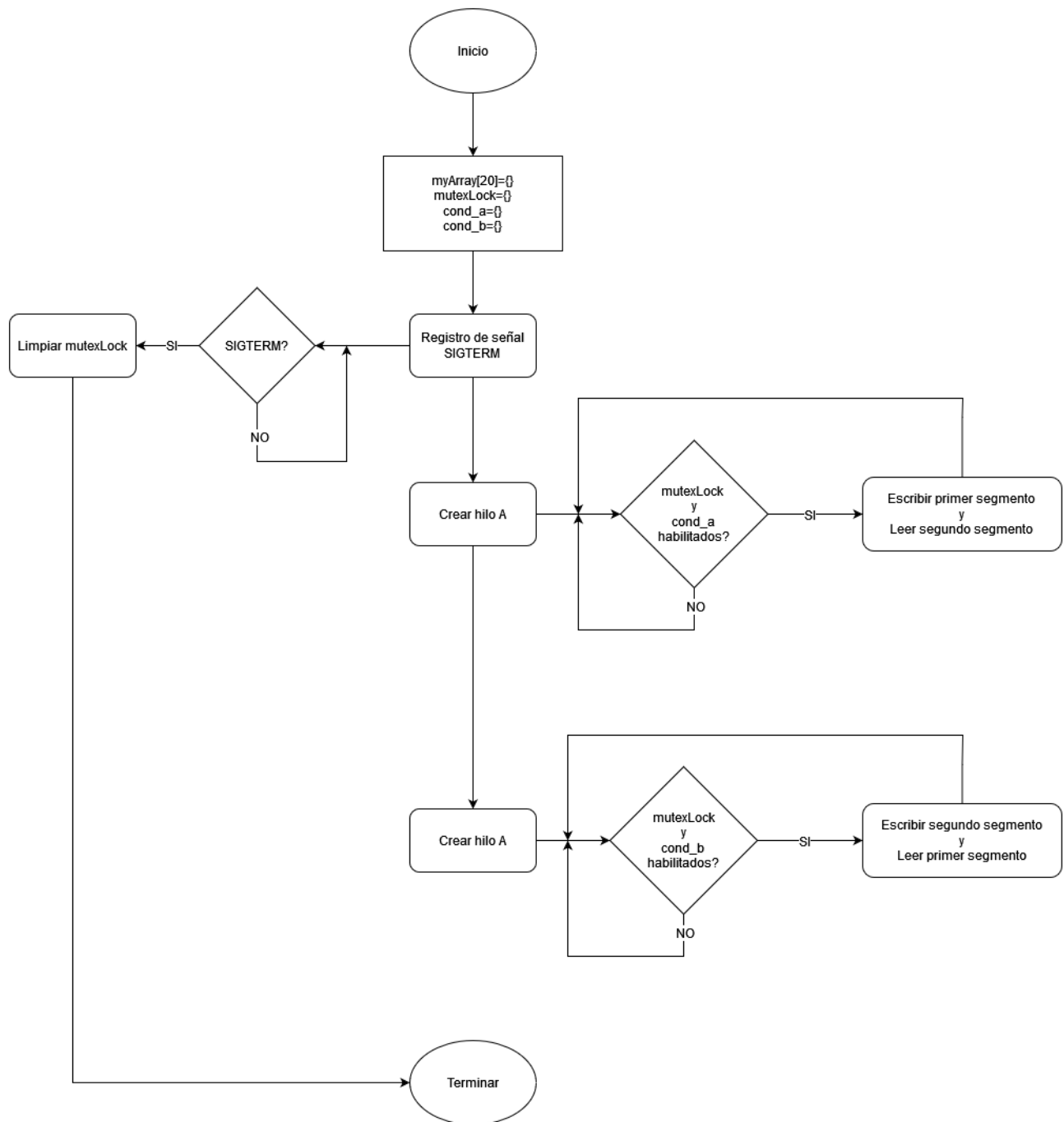


Fig. 1. Diagrama de flujo del algoritmo implementado.