

Entregable Trabajo Práctico N° 2

Tomás Vidal

Sistemas Operativos y Redes

Facultad de Ingeniería, UNLP, La Plata, Argentina.

14 de Octubre, 2024.

I. PROBLEMAS PRESENTADOS

I-A. Problema 10

I-B. Problema 11

Se debe crear un programa que pueda actuar de *dos formas diferentes*, como: **proceso A** o **proceso B**. Ambos deben poder acceder a un *bloque de memoria compartida*, para poder leer y escribir datos en común. Ambos procesos deben mantener una cierta sincronía, ya que uno no puede leer mientras el otro escribe y viceversa.

II. PROBLEMA 11

II-A. 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, ya que se registran manejadores de señales que esperan a ser llamados por **SIGUSR1** y **SIGTERM**.

II-B. Funciones principales

- *handle_termination*: Manejador que se encarga de liberar los recursos y terminar ambos procesos.
- *handle_write_read.a*: Realiza la escritura y lectura acorde a las especificaciones del **proceso A**.
- *handle_write_read.b*: Realiza la escritura y lectura acorde a las especificaciones del **proceso B**.
- *cypher*: Función de encriptación de datos.
- *decypher*: Función de desencriptación de datos.
- *setup_shared_memory*: Prepara la memoria compartida para ser usada.
- *save_pid_to_shared_mem*: Guarda en la memoria compartida el PID del proceso actual (A o B).
- *get_pid_from_shared_mem*: Obtiene el PID del proceso opuesto (A o B).
- *destroy_memory_block*: Libera la memoria compartida.

II-C. Explicación general del algoritmo

Se registran los manejadores para la señal **SIGUSR1** en ambos procesos; estos manejadores se encargan de la escritura y lectura correspondiente al proceso. Al inicio el *proceso A* se envía a sí mismo la señal **SIGUSR1**, de modo que si el *proceso B* existe, comienza un bucle entre ambos procesos de lectura y escritura, ya que al finalizar estas operaciones, cada proceso envía al otro la misma señal **SIGUSR1**.

Para poder enviar señales entre ambos procesos, se emplea

la memoria compartida para almacenar el PID correspondiente a ambos procesos, puesto que se guarda un arreglo de **int** y los PID son del mismo tipo.

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 $f^{-1}(x)$

III. DIAGRAMAS

Se adjuntan junto al presente las figuras de los diagramas de flujo, para que el lector pueda analizarlas precisamente.

Las responsabilidades del proceso A y el B son muy similares, la diferencia está en que en A debe finalizar el B en SIGTERM, y los bloque de memoria que leen y escriben son diferentes

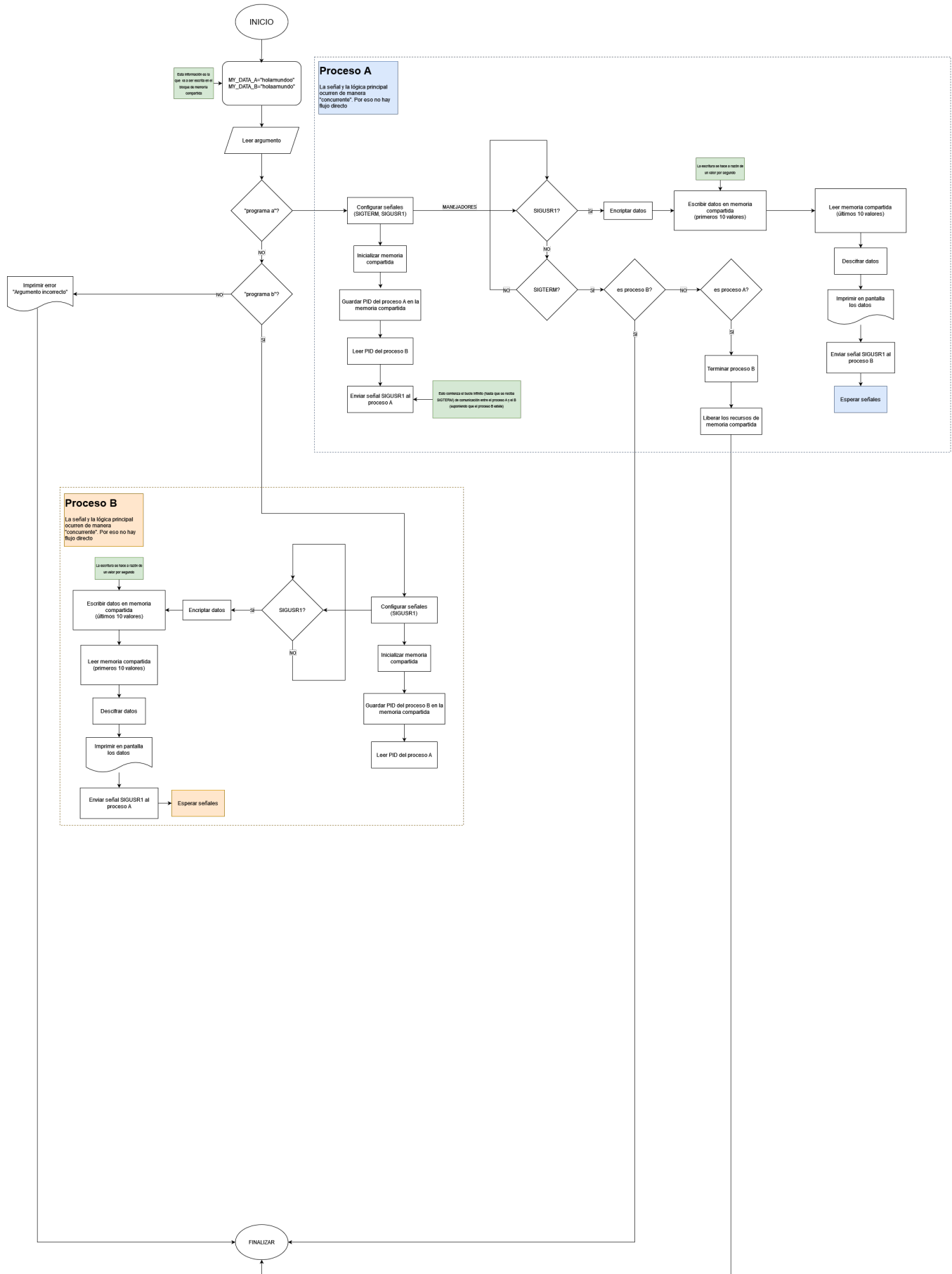


Fig. 1. Diagrama de flujo del problema 11.