

IPC (Inter-process Communication)

Es una función básica que nos dan los sistemas operativos y lo que hace es que yo pueda enviar información (datos o estados) entre dos procesos distintos.

Señales termina siendo un mecanismo de IPC muy sencillo que me da el estado de otro proceso, por ej. porque parte va.

La necesidad de usar IPC es que podemos implementar una tarea dividiéndola en subtareas y cada una de estas subtareas la podemos implementar como un proceso. Por lo tanto, voy a tener a varios procesos que están haciendo parte de la tarea que quiero hacer y en algún momento van a tener que compartir información. Es decir que si tendría más de un núcleo sería mucho más rápido realizar una tarea ya que estos procesos trabajan en simultaneo me permite modularidad que me permite remplazar un proceso por otro respetando su función en dicha tarea.

Clasificación de distintas IPC

Paso de mensajes:

- Usando llamados al kernel
- Pequeñas cantidades de información
- Fácil de implementar
- Estos son:
 - Pipe (Tubería anónimas)
 - FIFO (tuberías con nombre)
 - Cola de mensajes
 - Sockets

Memoria Compartida:

- Comparte espacio de memoria
- Rápida
- Requiere Sincronizar

En común Fifo y Pipe

- Orientado a flujo de bytes (no segmentados)
- Procesos comparten información que residen en el kernel
- No tiene persistencia en el kernel
- Se necesita una llamada a sistema
- Persistencia de proceso
- Unidireccional
- Se pueden abrir con la configuración bloqueante o no bloqueantes
- No discrimina que proceso escribió los datos que se escribió en la tubería
- Lee el primer proceso el dato y el resto no ya que se quita del buffer

Pipe – Tuberías anónimas

- Comunicación entre procesos relacionados
- Mensajes orientados a “stream” de datos (flujo de bytes), (no segmentados)
- No permiten compartir información entre procesos no relacionados

- Se puede abrir configuradas como bloqueante (por defecto) o como no bloqueante
- Para crear y abrir una tubería se utiliza pipe ()

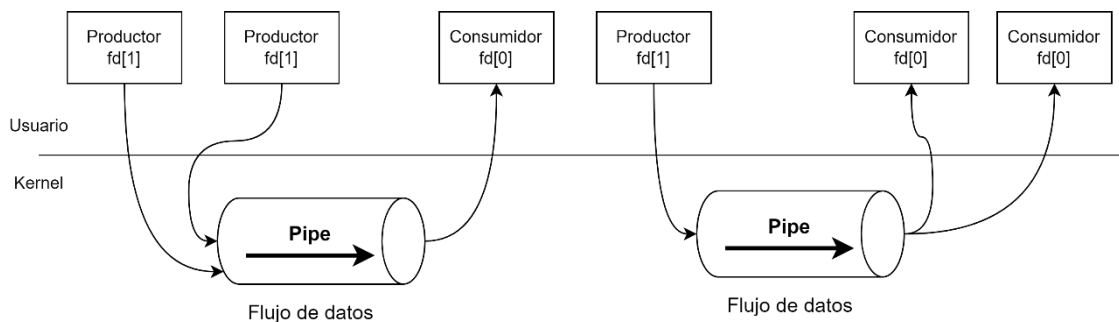
FIFO – Tuberías con nombre

Semánticamente, una FIFO es similar a una tubería. La principal diferencia es que una FIFO tiene un nombre dentro del sistema de archivos y se abre de la misma forma que un archivo normal. Esto permite que se use un FIFO para la comunicación entre procesos no relacionados (por ejemplo, un cliente y un servidor).

- Comunicación entre procesos no relacionados
- Mensajes orientados a “stream” de datos (flujo de bytes), (no segmentados)
- La tubería persiste con un nombre en filesystem
- Se puede abrir configuradas como bloqueante o como no bloqueante
- Para crear y abrir una fifo mkfifo () seguido de open

Problema de Stream de datos

- No discrimina que productor escribió los datos consumidos (ósea no sabe quién fue el primero en escribirlo).
- El primer consumidor lee los datos, el resto no.



Cola de Mensajes (posix)

Para solucionar las limitaciones de flujo de bytes utilizamos cola de mensajes. Algo que no tenía pipe ni fifo es que el tiempo de convivencia no es el mismo esto significa un proceso podría escribir un mensaje en una cola de mensajes y en un futuro otro proceso lo puede leer, ya que el buffer se almacena en kernel debido a que los mensajes están segmentados, únicamente si reiniciamos la maquina se borran.

- Procesos comparten información que residen en el kernel
- Se necesita una llamada a sistema
- Se puede abrir configuraciones bloqueantes y no bloqueantes
- Unidireccional

No se identifica quien envía cada mensaje para cada receptor

A cada mensaje se le asigna una prioridad

Lee el mensaje por antigüedad dependiendo la prioridad

Memoria Compartida

Persistencia del sistema de archivos (archivos de memoria)

Persistencia de kernel (objetos de memoria)

Requiere sincronización

No requieren llamadas a sistemas para intercambiar datos

Estados de una tubería

- Si se llena la tubería se bloquea
- Si no hay nada la tubería también se bloquea hasta que escriba
- Retorna cero a la lectura cuando no haya nada en el buffer y no haya ningún extremo de escritura conectado a algún proceso.
- Cuando cierro los descriptores de archivo de lectura ambos el sistema operativo le manda una señal SIGPIPE 7.