



INSTITUTO POLITÉCNICO NACIONAL  
UNIDAD PROFESIONAL INTERDISCIPLINARIA  
INGENIERÍA Y TECNOLOGÍAS AVANZADAS



## **TUBERÍAS CON NOMBRE Y SIN NOMBRE**

PROFESORA:

Susana Sanchez Najera

INTEGRANTES DEL EQUIPO:

Vargas Sánchez Andrea Liliana  
Cisneros Martínez Daphne Liliana  
Escala Acosta Andres Rafael

GRUPO: 2TV3

Fecha de entrega: 9 de noviembre del 2021

# TUBERÍAS

Una tubería (pipe, cause o “[”) consiste en una cadena de procesos conectados de forma tal que la salida de cada elemento de la cadena es la entrada del próximo. Permiten la comunicación y sincronización entre procesos. Es común usar un buffer de datos entre elementos consecutivos.

La comunicación al ocupar tuberías se basa en la interacción producto/consumidor, los procesos productores, aquellos que envían los datos, se comunican con los procesos consumidores, aquellos que reciben los datos, siguiendo un orden FIFO.

Una vez que un proceso consumidor recibe un dato este será eliminado de la tubería.

## TUBERÍAS SIN NOMBRE

Las tuberías sin nombre tienen asociado un fichero en memoria principal, por lo tanto, son temporales y se eliminan cuando no están siendo usadas por productores ni por consumidores.

Es el método más básico para comunicar procesos. Para poder crear una tubería sin nombre se utiliza:

*int pipe (int fildes[ ] );*

Donde *pipe* crea la tubería, se tienen dos casos:

1. *fildes[0]*: descriptor de un fichero de sólo lectura de la tubería.
2. *fildes[1]*: descriptor de un fichero de sólo escritura en la tubería.

Lo que se escribe en *fildes[1]* se va a leer en *fildes[0]*.

Un proceso va a crear una tubería y luego va a ejecutar el fork. Después de haber hecho el fork y pipe, el proceso de lectura cierra el lado de escritura de la tubería, se hace el intercambio de datos y se cierra el proceso.

Las tuberías utilizan un buffer que es gestionado por el núcleo donde se llevan a cabo las sincronizaciones entre los accesos de lectura y escritura, en el caso de que el buffer esté lleno se bloquea *write* y en caso contrario *read* se bloquea. El núcleo también se encarga de gestionar la tubería según FIFO.

Cuando se intenta escribir en el momento en que el extremo lector se cierra, se va a generar la señal *sigpipe*, cuando se cierre el extremo escritor se va a recibir un *EOF*.

## VENTAJA

- Se tiene un fácil manejo de la tubería.
- Permiten la comunicación entre el proceso que crea un cauce y procesos hijos tras la creación de la tubería.

## DESVENTAJAS

- Permiten comunicaciones de tipo half duplex.
- Solo pueden ser utilizadas entre procesos relacionados.

## TUBERÍAS CON NOMBRE

Son conocidas como *Tuberías FIFO*, ya que emplean un método de comunicación FIFO entre procesos ( First In First Out). Es decir, el primero que llega es el primero que se marcha, como en una cola, el primero que llega es el que antes termina.

Su diferencia respecto a las tuberías sin nombre radica en que el cauce se crea en el sistema de archivos, y por lo tanto no tienen carácter temporal.

Permiten la comunicación entre dos procesos que no tengan relación, son una clase de ficheros que tienen entrada en un directorio, existen en el sistema de archivos.

Se pueden crear tuberías con nombre desde shell con el comando *mkfifo*:

*int mkfifo(const char \*pathname, mode\_t mode);*

Donde *pathname* es la ruta que va a crear la tubería con nombre y *mode* es la máscara de permisos.

Los comandos para una tubería con nombre son:

- OPEN (abrir)
- CLOSE (cerrar)
- UNLINK (eliminar)
- READ (leer)
- WRITE (escribir)

Una ventaja que poseen las tuberías con nombres es que permiten la comunicación entre los procesos que usen dicha tubería, aunque no exista una conexión jerárquica entre ellos.

## CONCLUSIÓN

El uso de las tuberías, en los sistemas operativos multitarea, permite implementar, de forma muy eficiente, la creación de procesos; iniciando todos los procesos al mismo tiempo, y atendiendo automáticamente los requerimientos de lectura de datos para cada proceso cuando los datos son escritos por el proceso anterior. De esta manera el planificador de corto plazo va a dar el uso de la CPU a cada proceso a medida que pueda ejecutarse minimizando los tiempos muertos. Para mejorar el rendimiento, se pueden implementar las tuberías usando búferes, lo que permite al proceso proveedor generar más datos que lo que el proceso consumidor puede atender inmediatamente.

Al momento de implementar una tubería, es necesario tener en cuenta el propósito y la duración que tendrá dicha tubería; si será temporal (sin nombre) o si será “permanente” (con nombre). Puesto que las tuberías sin nombres se asocian con un fichero en memoria principal y, por lo tanto, se eliminan cuando no están siendo usadas ni por productores ni por consumidores. En contraste, las tuberías con nombres se crean en el sistema de archivos, y por lo tanto no tienen carácter temporal. Se manejan mediante llamadas al sistema, como el resto de ficheros del sistema.

## REFERENCIAS

Suárez, O. D., Quesada, A., & Santana, F. J. (2006). *Tema 5 : Comunicación entre Procesos mediante Tuberías*. PDF. Recuperado 9 de noviembre de 2021, de [http://sopa.dis.ulpgc.es/progsis/material-didactico-teorico/tema5\\_1transporpagina.pdf](http://sopa.dis.ulpgc.es/progsis/material-didactico-teorico/tema5_1transporpagina.pdf)

Juarez, D. (s. f.). *Tuberías - Pipes*. Scribd. Recuperado 9 de noviembre de 2021, de <https://es.scribd.com/doc/308736168/Tuberias-Pipes>