Tuberías

Eduardo C. Garrido Merchán

Sistemas Operativos. Práctica 1. Semana 3.

Definición

- Una tubería es un mecanismo de comunicación entre procesos padre-hijo.
- Una tubería está formada por dos descriptores de fichero.
- Un descriptor de fichero *(fd[0])* permite leer de la tubería.
- Un descriptor de fichero *(fd[1])* permite escribir en la tubería.
- Por lo tanto, mediante **read()** y **write()** se pueden comunicar dos procesos usando la tubería.

Definición

- Una tubería es un mecanismo de comunicación entre procesos padre-hijo.
- Una tubería está formada por dos descriptores de fichero.
- Un descriptor de fichero *(fd[0])* permite leer de la tubería.
- Un descriptor de fichero *(fd[1])* permite escribir en la tubería.
- Por lo tanto, mediante **read()** y **write()** se pueden comunicar dos procesos usando la tubería.

Definición

- Una tubería es un mecanismo de comunicación entre procesos padre-hijo.
- Una tubería está formada por dos descriptores de fichero.
- Un descriptor de fichero *(fd[0])* permite leer de la tubería.
- Un descriptor de fichero *(fd[1])* permite escribir en la tubería.
- Por lo tanto, mediante **read()** y **write()** se pueden comunicar dos procesos usando la tubería.

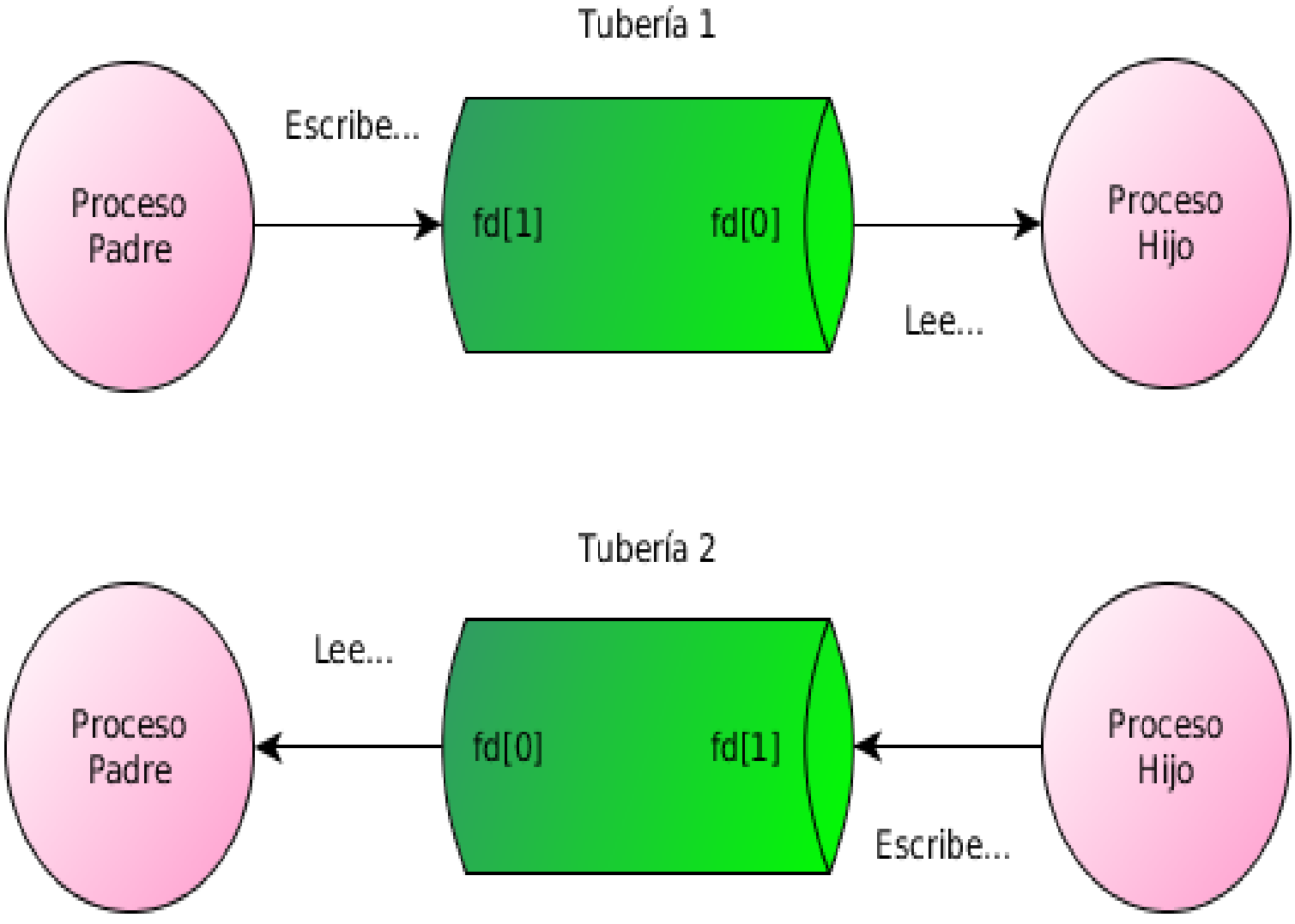
Definición

- Una tubería es un mecanismo de comunicación entre procesos padre-hijo.
- Una tubería está formada por dos descriptores de fichero.
- Un descriptor de fichero *(fd[0])* permite leer de la tubería.
- Un descriptor de fichero *(fd[1])* permite escribir en la tubería.
- Por lo tanto, mediante **read()** y **write()** se pueden comunicar dos procesos usando la tubería.

Definición

- Una tubería es un mecanismo de comunicación entre procesos padre-hijo.
- Una tubería está formada por dos descriptores de fichero.
- Un descriptor de fichero *(fd[0])* permite leer de la tubería.
- Un descriptor de fichero *(fd[1])* permite escribir en la tubería.
- Por lo tanto, mediante **read()** y **write()** se pueden comunicar dos procesos usando la tubería.

Descripción gráfica de la tubería



Uso de tuberías

- Se usa la llamada al sistema **pipe()** desde C.
- **pipe()** recibe como argumento de entrada un array de dos enteros: *int pipe(int fd[2])*.
- La llamada devuelve -1 en caso de error y si tiene éxito devuelve dos descriptores de fichero (que son insertados en la tabla de descriptores de fichero de los procesos).

Uso de tuberías

- Se usa la llamada al sistema **pipe()** desde C.
- **pipe()** recibe como argumento de entrada un array de dos enteros: *int pipe(int fd[2])*.
- La llamada devuelve -1 en caso de error y si tiene éxito devuelve dos descriptores de fichero (que son insertados en la tabla de descriptores de fichero de los procesos).

Uso de tuberías

- Se usa la llamada al sistema **pipe()** desde C.
- **pipe()** recibe como argumento de entrada un array de dos enteros: *int pipe(int fd[2])*.
- La llamada devuelve -1 en caso de error y si tiene éxito devuelve dos descriptores de fichero (que son insertados en la tabla de descriptores de fichero de los procesos).

Como emplear **pipe()**

- La creación de la tubería siempre debe preceder a la del proceso hijo.
- El proceso hijo hereda la tabla de descriptores del proceso padre, por tanto, el uso de la tubería.
- Ya que la tubería es unidireccional, se emplea para que un proceso lea y el otro escriba.
- Cada proceso cerrará el extremo de la tubería que no use para empezar la comunicación.
- El proceso hijo debe cerrar *(fd[1])* si quiere recibir datos del padre, y el padre deberá cerrar *(fd[0])* para enviar datos al hijo.
- Y viceversa.

Como emplear **pipe()**

- La creación de la tubería siempre debe preceder a la del proceso hijo.
- El proceso hijo hereda la tabla de descriptores del proceso padre, por tanto, el uso de la tubería.
- Ya que la tubería es unidireccional, se emplea para que un proceso lea y el otro escriba.
- Cada proceso cerrará el extremo de la tubería que no use para empezar la comunicación.
- El proceso hijo debe cerrar *(fd[1])* si quiere recibir datos del padre, y el padre deberá cerrar *(fd[0])* para enviar datos al hijo.
- Y viceversa.

Como emplear **pipe()**

- La creación de la tubería siempre debe preceder a la del proceso hijo.
- El proceso hijo hereda la tabla de descriptores del proceso padre, por tanto, el uso de la tubería.
- Ya que la tubería es unidireccional, se emplea para que un proceso lea y el otro escriba.
- Cada proceso cerrará el extremo de la tubería que no use para empezar la comunicación.
- El proceso hijo debe cerrar *(fd[1])* si quiere recibir datos del padre, y el padre deberá cerrar *(fd[0])* para enviar datos al hijo.
- Y viceversa.

Como emplear **pipe()**

- La creación de la tubería siempre debe preceder a la del proceso hijo.
- El proceso hijo hereda la tabla de descriptores del proceso padre, por tanto, el uso de la tubería.
- Ya que la tubería es unidireccional, se emplea para que un proceso lea y el otro escriba.
- Cada proceso cerrará el extremo de la tubería que no use para empezar la comunicación.
- El proceso hijo debe cerrar *(fd[1])* si quiere recibir datos del padre, y el padre deberá cerrar *(fd[0])* para enviar datos al hijo.
- Y viceversa.

Como emplear **pipe()**

- La creación de la tubería siempre debe preceder a la del proceso hijo.
- El proceso hijo hereda la tabla de descriptores del proceso padre, por tanto, el uso de la tubería.
- Ya que la tubería es unidireccional, se emplea para que un proceso lea y el otro escriba.
- Cada proceso cerrará el extremo de la tubería que no use para empezar la comunicación.
- El proceso hijo debe cerrar *(fd[1])* si quiere recibir datos del padre, y el padre deberá cerrar *(fd[0])* para enviar datos al hijo.
- Y viceversa.

Como emplear **pipe()**

- La creación de la tubería siempre debe preceder a la del proceso hijo.
- El proceso hijo hereda la tabla de descriptores del proceso padre, por tanto, el uso de la tubería.
- Ya que la tubería es unidireccional, se emplea para que un proceso lea y el otro escriba.
- Cada proceso cerrará el extremo de la tubería que no use para empezar la comunicación.
- El proceso hijo debe cerrar *(fd[1])* si quiere recibir datos del padre, y el padre deberá cerrar *(fd[0])* para enviar datos al hijo.
- Y viceversa.

Ejercicios

- Revisar el ejemplo de uso incluido en el documento Practica1 para ver un ejemplo del uso de tuberías.
- Realizar Ejercicio 9.

Ejercicios

- Revisar el ejemplo de uso incluido en el documento Practica1 para ver un ejemplo del uso de tuberías.
- Realizar Ejercicio 9.