

Investigación acerca de los Pipes.

Pipes.

Conocido como pipe o fifo, es un mecanismo de comunicación entre procesos que provee UNIX con las siguientes características generales, es un canal de entrada y salida de datos en el que se puede escribir y leer, permite que 2 o más procesos envíen información a otro. Existen 2 tipos, las pipes o unnamed fifos y las named fifos.

Las pipes o unnamed fifos sólo pueden ser empleados entre procesos relacionados (padre-hijo, hijo-hijo).

Algunas características de las "unnamed fifos" o pipes sin nombre son:

Un proceso crea una tubería o pipe y luego ejecuta la llamada fork, después, el proceso de lectura cierra el lado de escritura de la tubería y el proceso de escritura cierra el lado de la lectura de la tubería, se intercambian datos y se cierra el proceso.

Otras características son, un pipe sin nombre usa un buffer gestionado por el núcleo, la sincronización entre los accesos de escritura y lectura la realiza el núcleo, cuando el buffer está lleno, la llamada write se bloquea y cuando está vacío la llamada read se bloquea.

Las ventajas son fácil manejo de la tubería (read, write, close). Desventajas son permiten comunicaciones de tipo half duplex y son usadas en procesos relacionados.

Para los pipes con nombre o "named pipes", permiten comunicar procesos no relacionados, existen en el sistema ficheros, se crean utilizando `mknod` y existen hasta que se borren con `rm`. La lectura y escritura se realiza con las llamadas `read` y `write`, apertura con `open`.

La mayor ventaja es la comunicación entre procesos que no estén relacionados.

- No se tiene que iniciar la escritura y lectura al mismo tiempo.
- Se pueden tener múltiples lectores y escritores que no necesitan un ancestro.
- Al igual que un archivo se puede tener el control de propietario y los permisos.
- Son persistentes (existen en el sistema de archivos).

Desventajas con otros mecanismos.

- Los pipes no pueden ser utilizados para transmitir datos a múltiples procesos receptores de forma simultánea, ya que al leer los datos de la tubería estos son borrados.
- Si existen varios procesos que desean leer en un extremo de la tubería, un proceso que desea escribir en el otro extremo no puede dirigir los datos a un proceso concreto. Asimismo, si existen varios procesos que desean escribir en la tubería, no existe forma de determinar cuál de ellos envía los datos.

- Si un proceso envía varios mensajes de diferente longitud en una sola operación de escritura en la tubería, el proceso que lee el otro extremo de la tubería no puede determinar cuántos mensajes han sido enviados, o donde termina un mensaje y donde empieza el otro, ya que los datos en la tubería son tratados como un flujo de bytes no estructurados de tamaño fijo.

Ventajas.

Son una forma eficiente de transferir datos de un proceso a otro.

Son sencillas de usar.

Son bidireccionales.

Pruebas de funcionamiento de los Pipes sin nombre y Pipe con nombre.

```
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$ gcc UnamedPipe.c -o UnamedPipe
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$ ./UnamedPipe
Lectura en proceso padre: Cadena de lectura
Bytes leidos:16
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$
```

Figura 1. Pipe sin nombre.

```
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$ gcc NamedPipeW.c -o NamedPipeW
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$ ./NamedPipeW
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$
```

Figura 1.1. Pipe con nombre, proceso escritor.

```
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$ gcc NamedPipeR.c -o NamedPipeR
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$ ./NamedPipeR

Mensaje leído por proceso 2: Procesos externos
Numero de bytes: 18
luisfc@luisfc-X541NA:~/Documentos/Sistemas Operativos/Pipes$
```

Figura 1.2. Pipe con nombre, proceso lector.