

# Fundamentos de Software de Comunicaciones

## E.T.S.I. Telecomunicación

### Prácticas de procesos (pipes y fifos)

En los ejercicios se tendrá especial cuidado en controlar las señales y los errores en las llamadas al sistema.

1. Implemente un programa que genere un proceso hijo encargado de mostrar por pantalla la información que le suministra su padre. El padre obtiene la información del usuario por teclado. Los procesos terminan cuando se introduce por teclado la cadena "fin\n" (es decir, "fin" más la pulsación de la tecla INTRO). Utilice una pipe para el trasiego de información entre padre e hijo.
2. Realice una variación del ejercicio anterior donde esta vez el hijo será el que ponga fin a los dos procesos cuando alcance la cuenta de X caracteres leídos por la pipe (este valor se puede obtener como argumento al programa). En esta ocasión, se creará una pipe adicional para que el hijo le pueda transmitir al padre si éste debe continuar leyendo mensajes del usuario por teclado o debe finalizar.  
El algoritmo es el que sigue:
  - a. Padre lee mensaje de teclado del usuario y lo manda al hijo por la pipe original.
  - b. Hijo lee mensaje por la pipe original, lo muestra por pantalla y actualiza su cuenta de bytes leídos. Si se alcanza o supera el máximo X entonces utiliza la pipe nueva para enviar al padre el carácter 'n' y termina. Si no se alcanza X, entonces utiliza la pipe nueva para enviar al padre el carácter 'y' y vuelve a (b).
  - c. El padre lee por la pipe nueva un carácter. Si es 'y' vuelve al paso (a). Si es 'n', hace wait() del hijo y termina la ejecución.
3. Basándose en los ejercicios 1 y 2, consiga la misma funcionalidad a través de dos programas independientes, que se ejecutarán a la vez en el sistema. En esta ocasión, hay que sustituir las pipes por fifos. Uno de los programas es el que hace la lectura por teclado y el envío a la fifo. El otro recibe por la fifo y muestra por pantalla. El nombre de la fifo puede obtenerse como argumento a los programas. Crea la fifo por consola, previamente.
4. Elabore un programa donde un padre le transmita el contenido de un fichero de texto a un hijo utilizando una pipe. La misión del hijo es recibir los datos de ese fichero y convertir todas sus mayúsculas a minúsculas para grabarlo con el nombre que decida el padre (NOTA: mira la función *tolower()*). El hijo recibirá del padre el nombre del fichero donde se grabarán los datos, porque el padre primero llama a *fork()* y luego pide el nombre del fichero de entrada y el de salida al usuario por teclado. El fichero de entrada lo abre el padre. El de salida lo abre el hijo cuando el padre le pasa el nombre a través de la pipe.

5. Aunque el programa anterior puede parecer correcto, no lo es.
  - a. Considera qué ocurre si al principio del padre introducimos una espera. Por ejemplo una función que duerma el proceso, como *sleep(10)*, antes del primer write. ¿Sigue funcionando bien? ¿Por qué?
  - b. Considera ahora qué ocurre si al principio del hijo introducimos una espera. Por ejemplo, un *sleep(10)* antes del primer read. ¿Sigue funcionando bien? ¿Por qué?
  - c. Arregla el programa para que funcione bien en ambos casos.
6. Implemente un programa que establezca una comunicación bidireccional entre dos procesos, padre e hijo, cuya especificación es la siguiente:
  - a. Proceso padre: lee de teclado una secuencia de operaciones aritméticas (+, - o \*) y dos operandos (tipo **short**), hasta que, al solicitar el operador, el usuario introduce la cadena “fin”. Este proceso comprueba que es una de las tres operaciones permitidas y la envía, junto con operandos, al hijo. Se debe usar para ello una única llamada a la función **write/write\_n**. El padre entonces espera a que el hijo le envíe el resultado de la operación, que viene en una variable tipo **int**.
  - b. Proceso hijo: en una única llamada a **read/read\_n**, este proceso obtiene de la tubería tanto el operador como los dos operandos, realiza la operación correspondiente, y envía el resultado de vuelta al padre en una variable entera. El proceso debe iterar hasta que el padre cierre la tubería por la que le envía la información al hijo.