

COMPROC 2

Pipes con nombre

Escribe un programa llamado `escritor.c` que lee de entrada estándar números en formato ASCII separados por un carácter de nueva línea (puedes asumir que la entrada de datos será correcta). Para cada número deberá convertirlo a formato entero (representación interna de la máquina) y escribirlo en una pipe con nombre, llamada “mipipe”. Si la pipe no existe la creará, sólo con permisos de lectura y escritura para el propietario y para el grupo del propietario.

Escribe un programa llamado `lector.c` que reciba como parámetro el nombre de un fichero. Si ese fichero no existe deberá crearlo con permisos de lectura y escritura para el propietario y los miembros de su grupo y sin ningún permiso para el resto de usuarios. En caso de que el fichero ya exista deberá truncar su contenido. A continuación, recibirá por la pipe con nombre llamada “mipipe” una lista de números enteros en su representación interna de la máquina. Deberá convertirlos a formato ASCII y guardarlos en el fichero que ha recibido como parámetro, usando un carácter de nueva línea como separador entre números.

Pipes sin nombre

Adjunto a este enunciado tienes los programas ejecutables `lector_v2` y `escritor_v2` que son una variación de lo que se pide en el enunciado anterior y en los que la pipe con nombre ya no se utiliza: en `lector_v2` hemos reemplazado las lecturas de la pipe con nombre con lecturas de entrada estándar, y en `escritor_v2` hemos reemplazado las escrituras sobre la pipe con nombre son escrituras en salida estándar.

Crea un programa llamado `launch.c` que reciba como parámetro un nombre de fichero y cree dos procesos hijos que se comuniquen con una pipe sin nombre. Uno de los hijos deberá mutar a `escritor_v2`. El otro hijo deberá mutar a `lector_v2` pasando como parámetro el nombre de fichero que ha recibido `launch`. Tienes que conseguir que las escrituras en salida estándar de `escritor_v2` vayan a la pipe sin nombre. Y las lecturas de entrada estándar de `lector_v2` sean de la pipe sin nombre donde `escritor_v2` ha escrito. De manera que al comportamiento sea equivalente al que hemos conseguido con la ejecución de `escritor` y `lector` en el apartado anterior. Por su parte, el padre deberá esperar a que los dos hijos acaben la ejecución.