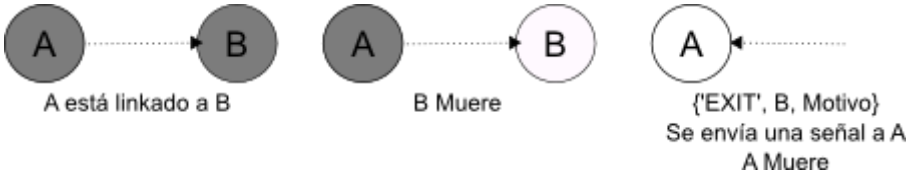
Reanimator

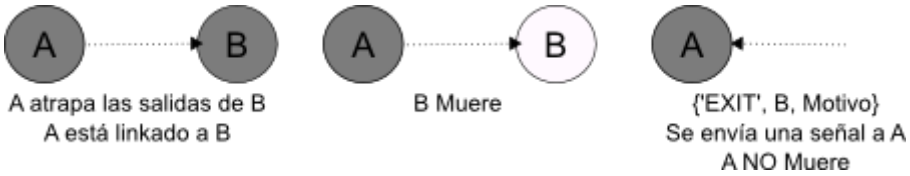
Control de errores en sistemas concurrentes

Los procesos enlazados tienen un mecanismo por el cual, dos procesos A y B quedan enlazados de tal forma que, si el proceso A muere envía una señal exit al proceso B y viceversa. Así se evita que haya procesos moribundos en nuestro sistema.



Como se puede ver en la ilustración, el proceso A esta linkado al proceso B y cuando el proceso B termina o muere envía una señal de salida al proceso A que a su vez termina.

Sin embargo, se puede variar este comportamiento según las necesidades. Para ello, el proceso A debe atrapar la señal de salida y procesarla decidiendo que es lo que desea hacer en ese momento.



En la ilustración anterior podemos ver el comportamiento de un proceso A que esta linkado a B y además, atrapa la señal de salida. Al proceso que atrapa la salida se le llama System Process. En este caso, el proceso A podría decidir terminar con su ejecución o continuar.

Pero, ¿Cómo sabe el proceso B a quién tiene que avisar? El proceso B tiene lo que se llama un conjunto de procesos linkados. Es decir, cuando un proceso, en nuestro caso A, se linka a otro proceso este, en nuestro caso B, registra su PID en un conjunto de procesos linkados a él. De esta forma cuando el proceso linkado termina, B, recorre el conjunto de procesos linkados enviando la señal de salida.

Objetivo:

Los procesos enlazados nos dotan de un mecanismo por el cual, dos procesos A y B quedaban enlazados de tal forma que, si el proceso A moría enviaba una señal exit al proceso B y viceversa. Con dicho mecanismo evitábamos que hubiera procesos moribundos creados en nuestro sistema. *Muere rápido para recuperarte lo antes posible*.

Puede que no deseemos un control tan férreo como nos propone el sistema de procesos enlazados. A lo mejor, sólo necesitamos saber si está vivo, o no, el proceso A para poder utilizarlo. A lo mejor, no nos supone ningún problema seguir trabajando sin el proceso A. En

estos casos, lo que necesitaremos es un monitor. Un monitor es un tipo de enlace asimétrico, mientras que los enlaces (link's) son de carácter simétrico. Es decir, si el proceso M monitoriza al proceso A, el proceso M sabrá sobre el proceso A, pero no al revés, como pasa con los procesos enlazados.

Para realizar este comportamiento tenemos que disponer de un servidor central y otro de apoyo. Si el servidor central decide caerse. ¿No estaría bien que los clientes pudieran monitorizar al servidor? Es lo suyo, ¿verdad? Para que en el caso de caídas del servidor central pasar automáticamente al servidor de respaldo. Un comportamiento muy deseable en muchos casos.

Voy a implementar un servidor genérico cliente-servidor, se caracteriza por un servidor central y un número indeterminado de clientes, sonde el estado del servidor vendrá dado por un conjunto de nombres.

Comportamiento:

- Módulo Paciente:

  Quizás hemos pasado por la desagradable experiencia de visitar una unidad de cuidados intensivos. Los pacientes en estos lugares están enganchados a unos monitores, o mejor dicho, los monitores están enganchados a los pacientes.

- Módulo Hospital/Reanimator

  Los pacientes irán dándose de alta en forma de procesos. El monitor se encarga de avisar, en caso de algún tipo de alarma o crisis del paciente. Dotado de la posibilidad de reanimar al paciente en caso de riesgo. Así, resultará más difícil que el paciente fallezca.

  Servidor de nombres de pacientes. Como interfaz tendrá:

  1. Nuevo: Inserta un nuevo paciente por su nombre en el servidor de forma síncrona.
  2. Borrar: Borra un nombre de paciente del servidor de forma asíncrona.
  3. Está: Comprueba si un nombre de paciente dado está registrado en el servidor.
  4. Listar: Lista de forma asíncrona los nombres de pacientes que tiene registrado el servidor.