Universidad Carlos III

Sistemas Distribuídos

Curso 2023-24

Práctica 1

Colas de mensajes POSIX

Ingeniería Informática, Tercer curso

Adrián Fernández Galán (NIA: 100472182, e-mail: 100472182@alumnos.uc3m.es)
César López Mantecón (NIA: 100472092, e-mail: 100472092@alumnos.uc3m.es)

Prof . Félix García Caballeira
Grupo: 81

Índice

1 Introducción

El desarrollo de este proyecto consiste en implementar una aplicación cliente-servidor, donde los diferentes clientes podrán guardar información en tuplas a través del servidor, de forma transparente. Para que esto pueda darse se pide que la comunicación entre los clientes y el servidor se de a través de colas POSIX. Es importante destacar que para el tratamiento de las solicitudes el servidor será concurrente, utilizando hilos.

2 Diseño

La aplicación constará de dos partes diferenciadas: los clientes y el servidor.

2.1 Cliente y Biblioteca dinámica

El cliente estará formado por dos partes, el main de cliente, que corresponde con el fichero *src/cliente.c*, y la comunicación con el servidor, que corresponde con el fichero */src/clave.c*. Esta último se compilará como una librería dinámica *lib.so* de forma que la comunicación sea independiente de la implementación de cliente. De esta forma, logramos una aplicación que, en caso de cambiar la forma de comunicación o implementación de los servicios no sea necesario volver a compilar siempre que se respete la interfaz actual.

Para compilar el archivo *claves.c* se ha empleado el siguiente comando en un archivo *Makefile*.

```
gcc -c -fPIC -shared -o lib.so claves.c
```

El cliente podrá llamar a las las diferentes funciones ofrecidas por la interfaz, de esta manera se podrá interactuar con las distintas tuplas almacenadas. Los servicios, no obstante, estarán implementados en el servidor de forma transparente al usuario.

En esta implementación cada cliente creará una petición que corresponda con la funcionalidad invocada y creará una cola POSIX que le corresponde, donde se recibirá la respuesta, y abrirá la cola POSIX que le corresponde al servidor, donde volcará su petición y procederá a esperar la respuesta.

En el servidor se encuentran las funcionalidades que se encargan de la comunicación con el cliente, es decir *src/servidor.c*, y la correspondiente implementación de las tuplas, es decir *src/imp_clave.c*. En cuanto a la parte de comunicación con el cliente, el servidor abre su cola POSIX y espera de forma indefinida a que los clientes realicen sus peticiones. En el momento en el que llega una petición el servidor analizará la petición y llamará a la funcionalidad que el cliente haya pedido. Cuando la funcionalidad se haya realizado se volcará en contenido de respuesta en la cola de cliente. Para la implenetación de las distintas funcionalidades se ha optado por guardar las tuplas en ficheros, lo que nos permite acceder a las tuplas en distintas sesiones, ya que se guardan en la memoria permanente. Para poder crear y destruir simplemente creamos los ficheros y los borramos, y para leer y escribir abrimos el fichero y volcamos los datos de manera binaria.

2.2 Mensajes

Aquí explicar los argumentos de entrada y salida que nos llevan a desarrollar las estructuras que hemos usado.

2.3 Uso de ficheros

2.4 Servidor

Aquí explicar el diseño concurrente del servidor. Donde están los mutex y por qué. Diseño de pool de hilos o hilo por petición.

3 Descripción de pruebas

Pruebas que hemos ejecutado sobre nuestra aplicación: ejecución con varios clientes, validación de las funciones... Creo que lo mejor sería dividirlo en subsecciones, una para cada función, otra para las comunicaciones y otra para la concurrencia.

4 Conclusiones

Escribimos aquí