Facultad de Ingeniería Universidad de Buenos Aires

75.59 Técnicas de Programación Concurrente I

Trabajo Práctico N°2

Integrantes:

Padrón	Nombre	Email
87991	Ramonda, Juan Pablo	juanpr@gmail.com
89542	Hurtado, Pablo Nicolás	hurtado.pani@gmail.com
89579	Torres Feyuk, Nicolás R. Ezequiel	ezequiel.torresfeyuk@gmail.com

Índice

1. Introducción

El presente trabajo consiste en el desarrollo de una aplicación *concurrente*. El objetivo de la misma es permitir el almacenamiento de datos centralizado y su posterior recuperación por parte de los clientes.

Debido a que la materia apunta a obtener conocimientos sobre los mecanismos de concurrencia vistos hasta el momento en la cátedra, el proyecto debe correr en una única computadora y funcionar bajo un ambiente *Unix/Linux* dado que los mecanismos vistos son los implementados por *System V*.

La implementación de la aplicación consiste en un esquema *Cliente-Servidor*. El servidor se encuentra escuchando peticiones del cliente. Los clientes deben conectarse al servidor, y luego de esto pueden utilizar los servicios provistos por el mismo.

2. Modo de Operación

2.1. Cómo compilar y correr la aplicación

Para poder correr la aplicación correctamente, lo primero que debe realizarse es compilar el programa. Para esto se ha provisto de un *Makefile* que se encagará de realizar el proceso de compilación.

Para compilar la aplicación servidor se debe ingresar el siguiente comando:

```
$ make server
```

Para compilar la aplicación cliente se debe ingresar el siguiente comando:

```
$ make client
```

En el caso de que desee compilar todas las aplicaciones juntas puede hacerlo por medio de alguno de los siguientes comandos:

```
$ make server client
$ make all
```

2.2. Casos de Uso

Como bien se dijo, la aplicación está compuesta por un *Servidor* que escucha peticiones y *Clientes* que realizan las mismas. De esta forma, el *Servidor* es un proceso que no tiene interacción con el usuario. En cambio, el *Cliente* si tiene interacción con el usuario. La aplicación *Cliente* despliega un menú con las tareas que puede realizar el mismo. A continuación se explica el modo de uso de cada una de las opciones del menú:

- **1-Leer un Registro:** Mediante esta opción, el cliente le indica al servidor qu/'e registro desea recuperar de la base de datos.

  Precondición: El servidor se encuentra corriendo.

  - El usuario solicita la lectura de un registro.
  - El programa cliente pregunta el número de registro deseado.

- El usuario ingresa el número de registro.
- El programa cliente lo envía al servidor.
- El servidor recibe el pedido, valida el número pedido y responde enviando el registro solicitado o error (E1).
- El cliente recibe la respuesta y muestra el registro al usuario.

E1) El número de registro es inválido. Corresponde a un registro inexistente en la base de datos.

- El programa cliente muestra un mensaje de error y vuelve al menú de selección de operación.

- **2-Agregar un registro a la Base de Datos:** Mediante esta opción, el cliente le indica al servidor que desea agregar un registro. Precondición: El servidor se encuentra corriendo.

  - El usuario solicita agregar un registro a la base de datos.
  - El programa cliente pregunta el campo nombre del nuevo registro.
  - El usuario ingresa el nombre.
  - El programa cliente valida el tama no del nombre (E1).
  - El programa cliente pregunta el campo dirección del nuevo registro.
  - El usuario ingresa la dirección.
  - El cliente valida el tama no de la dirección (E1).
  - El programa cliente pregunta el campo iteléfono del nuevo registro.
  - El usuario ingresa el teléfono.
  - El cliente valida el tama no del teléfono. (E1).
  - El cliente envía el registro a agregar al servidor.
  - El servidor recibe el mensaje, agrega el registro a la base de datos, y envía al cliente la respuesta.
  - El cliente recibe el mensaje y notifica al usuario del éxito de la operación.
  - El cliente muestra el menú principal.

  E1) El tamaño del campo es inválido.

  - El programa cliente muestra un mensaje de error y solicita nuevamente el ingreso del valor del campo.
  - El usuario debe ingresar el valor del campo hasta que la validación sea correcta.

- **3-Modificar un registro:** Mediante esta opción, se permite la modificación de un registro de la base de datos.

  - El usuario solicita la modificación de un registro.
  - El cliente pregunta el número de registro a modificar.
  - El usuario ingresa el número.

- El cliente pregunta el valor del campo nombre del registro.
- El usuario ingresa el nombre.
- El programa cliente valida el tamaño del nombre (E1).
- El cliente pregunta el valor del campo dirección del registro
- El usuario ingresa la dirección.
- El programa cliente valida eltama no del campo dirección (E1).
- El cliente pregunta el valor del campo teléfono del registro
- El usuario ingresa el teléfono.
- El programa cliente valida el tama no del campo teléfono (E1).
- El cliente notifica al usuario el resultado de la operación.
- El cliente muestra el menú principal.

  E1) El tama no del campo es inválido.
  - El programa cliente muestra un mensaje de error y solicita nuevamente el ingreso del valor del campo.
  - El usuario debe ingresar el valor del campo hasta que la validación sea correcta.

- **4-Eliminar un registro de la Base de Datos:** Mediante esta opción, el cliente puede eliminar un registro de la base de datos.

  Precondición: El servidor se encuentra corriendo.

  - El usuario solicita al programa cliente la eliminación de un registro.
  - El programa cliente solicita el número de registro que se desea eliminar.
  - El programa cliente envÃa la petición de eliminación al servidor.
  - El servidor realiza la petición y envía el resultado al cliente.
  - El cliente recibe la respuesta del servidor y la muestra al usuario (E1).
  - El cliente regresa al menú principal.

  E1) El número de registro ingresado no existe.

  - El cliente notifica el error al usuario.
  - El cliente regresa al menú principal.

- **5-Salir de la Aplicación:** Mediante esta opción, el cliente abandona la aplicación.

  Precondición: El servidor se encuentra corriendo.

  - El usuario solicita salir del programa cliente.
  - Finaliza el programa cliente.

3. Análisis de la Solución

3.1. División de la Aplicación en Procesos

Debido a la naturaleza de la solución adoptada, se pueden distinguir fácilmente los siguientes procesos:

- **Servidor:** Este proceso se encarga de escuchar las peticiones los clientes. Mientras ningún usuario le envíe ningún mensaje, el mismo se bloquea. Cuando un cliente le envía algún mensaje, despierta al mismo y este se encarga de responderle al usuario.
- **Cliente:** Al abrir un usuario una aplicación Cliente, se crea un nuevo proceso el cual puede comunicarse con el servidor. El Cliente envía peticiones al servidor esperando una respuesta del mismo.

3.2. Comunicación entre Procesos

A continuación se detalla como es la comunicación entre los procesos según los casos de uso:

- **Leer Registro de la Base de Datos:**
  - El servidor y un cliente se comunican.
  - El cliente envía un mensaje al servidor para leer un registro.
  - El servidor recibe el mensaje y contesta al cliente con el registro solicitado o un error.
- **Agregar Registro al Servidor:**
  - El servidor y un cliente se comunican.
  - El cliente envía un mensaje al servidor para agregar un registro a la base de datos, junto con el registro a agregar.
  - El servidor recibe el mensaje y contesta con el resultado de la operación.
- **Modificar un Registro:**
  - El servidor y un cliente se comunican.
  - El cliente envía un mensaje al servidor pidiendomodificar un registro, con el registro modificado.
  - El servidor recibe el mensaje y contesta al cliente con el resultado de la operación.
- **Eliminar un registro de la base de datos:**
  - El servidor y un cliente se comunican.
  - El cliente envía un mensaje al servidor pidiéndo la eliminación de un registro de la base de datos junto con el número de registro a eliminar.
  - El servidor le responde con el resultado de la operación.

4. Protocolo de Comunicación

A continuación se detalla el protocolo utiizado en la aplicación. Para transmitir mensajes se utilizó una estructura con los siguientes elementos:

- mtype: código necesario por el mecanismo de comunicación para identificar el receptor del mensaje.
- pid: pid del cliente que envía el mensaje al servidor.
- numReg: número de registro de la base de datos por el cual se solicita una operación (lectura/modificación/eliminación).
- reg: registro de la base de datos (en caso de querer agregar/modificar).

Las acciones posibles y los valores para las mismas son:

1. **Conexión**: Este comando se envía automáticamente cuando se abre un cliente.
   - Cliente:
     mtype = 1
     pid = pidCliente
     comando = CON
     numReg = indefinido
     reg = indefinido
   - Servidor:
     mtype = pidCliente
     pid = pidCLiente
     comando = CONOK
     numReg = indefinido
     reg = indefinido
2. **Desconexión**: Este comando se envía automáticamente cuando se desea cerrar un cliente.
   - Cliente:
     mtype = 1
     pid = pidCliente
     comando = DESCON
     numReg = indefinido
     reg = indefinido
   - Servidor:
     mtype = pidCliente
     pid = pidCLiente
     comando = DESCONOK
     numReg = indefinido
     reg = indefinido

3. **Lectura de un registro de la BD**: Comando ejecutado cuando se desea consultar un registro de la base de datos.
   - Cliente:
     mtype = 1
     pid = pidCliente
     comando = LEERRG
     numReg = número de registro a leer
     reg = indefinido
   - Servidor:
     mtype = pidCliente
     pid = pidCLiente
     comando = LEERRGOK
     numReg = indefinido
     reg = registro leido
4. **Modificación de un registro de la BD**: Comando ejecutado cuando se desea modificar un registro de la base de datos.
   - Cliente:
     mtype = 1
     pid = pidCliente
     comando = MODRG
     numReg = número de registro a modificar
     reg = registro a reemplazar
   - Servidor:
     mtype = pidCliente
     pid = pidCLiente
     comando = MODREGOK
     numReg = indefinido
     reg = indefinido
5. **Baja de un registro de la BD**: Comando ejecutado cuando se desea eliminar un registro de la base de datos.
   - Cliente:
     mtype = 1
     pid = pidCliente
     comando = BAJARG
     numReg = número de registro a eliminar
     reg = indefinido
   - Servidor:
     mtype = pidCliente
     pid = pidCLiente
     comando = BAJARGOK

numReg = indefinido
reg = indefinido

Al no concluir exitosamente una operación, el servidor responde con un mensaje de error, el cual se detalla a continuación:
Servidor:

mtype = pidCliente

pid = pidCLiente

comando = ERROR

numReg = indefinido

reg = indefinido