

TP1: Comunicaciones

Taller de Programación I - Curso Ing. Diego Azcurra - 1er cuatrimestre 2016

Objetivo

Desarrollar una aplicación Cliente-Servidor constituida por un programa llamado Servidor que recibe, procesa y responde mensajes a uno o más programas llamados clientes.

Desarrollo

Servidor:

Se deberá desarrollar un programa en C/C++ que lea desde un archivo en formato XML la definición de un puerto y la cantidad de clientes que pueden ser atendidos simultáneamente. Si el archivo no existe, se creará uno por defecto. El nombre del archivo será pasado por línea de comandos.

Una vez parseado el archivo XML, el servidor atenderá los mensajes enviados por las distintas instancias del programa cliente (la salida del programa deberá ser controlada). Teniendo en cuenta el máximo especificado, el servidor deberá poder procesar simultáneamente los mensajes de todos aquellos clientes que lo soliciten, mostrando por pantalla dicho procesamiento. Una vez procesado el mensaje deberá ser enviado de vuelta al cliente informando si el procesamiento ha sido exitoso o no. En caso de superar el máximo se deberá informar al programa cliente.

Además el servidor deberá almacenar en un archivo los resultados de las operaciones, errores encontrados (mensaje incorrecto), problemas de comunicación, etc. Se deberán loguear distintos tipos de mensajer.

Ejemplo de XML del servidor

```
<servidor>
    <CantidadMaximaClientes>7</CantidadMaximaClientes>
    <puerto>15636</puerto>
</servidor>
```

El cliente

Se deberá desarrollar un programa en C/C++ que lea desde un archivo en formato XML la definición de la dirección IP en la cual se encuentra ejecutando el servidor y el puerto en el cual está escuchando. Además este archivo posee una lista de mensajes a ser procesados. El nombre del archivo será pasado por línea de comandos.

Una vez que el cliente inicia, esté creará dinámicamente un menú por pantalla en el cual se podrá seleccionar una de las siguientes opciones:

1. **Conectar:** Se establecerá la conexión con el servidor.
2. **Desconectar:** Se cerrará la conexión con el servidor.
3. **Salir:** Se finalizará la ejecución del programa.
4. Por cada uno de los mensajes que se encuentra en el archivo de configuración se creará una opción de menú que dirá "Enviar el mensaje <id>", donde <id> es el identificador único del mensaje. Cada vez que un mensaje es enviado, el cliente esperará la respuesta del servidor y mostrará por pantalla el mensaje y su respuesta.
5. **Ciclar:** permite el envío de todos los mensajes en forma iterativa durante una cantidad determinada de milisegundos que deberá ser ingresado tras la selección de esta opción.

Al igual que el programa llamado servidor el cliente deberá informar los distintos eventos en un archivo de log. Este archivo además de tener información sobre errores de comunicación, pérdida de mensajes, mensajes recibidos, etc.

Mensaje

Un mensaje es una entidad que consta de un identificador único, el tipo de dato a ser enviado (INT, DOUBLE,STRING,CHAR) y el valor correspondiente.

Por lo tanto para cada mensaje se especifica:

- id
- Tipo
- Valor

Ejemplo XML Cliente

```
<Cliente>
  <conexion>
    <IP>192.168.0.1</IP>
    <puerto>65532</puerto>
  </conexion>
  <mensaje>
    <id>Mensaje1</id>
    <tipo> INT </tipo>
    <valor> i1 </valor>
  </mensaje>
  ....
  ....
  <mensaje>
    <id>Mensaje2</id>
    <tipo> String </tipo>
    <valor> hola mundo </valor>
  </mensaje>
</mensajes>
```

Restricciones

- La implementación deberá estar hecha en C/C++.
- Para la lectura y escritura de archivos XML debe utilizarse una biblioteca.
- Todo el código debe ser desarrollado íntegramente por cada grupo. No se permite la reutilización de código de cuatrimestres anteriores o de otras materias. Ante cualquier duda se deberá consultar con los docentes. La reutilización de código sin consulta previa será condición suficiente para la desaprobación de la materia.

Este enunciado no es definitivo. Si se realizan cambios en clase se respetarán y evaluarán los mismos.

Fechas

Semana #	Fecha	Tema
1	9 de marzo	Presentación de la materia
2	16 de marzo	Presentación enunciado TP 1
3	23 de marzo	Consultas
4	30 de marzo	Consultas
5	6 de abril	Consultas
6	13 de abril	Entrega TP1. Presentación enunciado TP 2
7	20 de abril	Primer recuperatorio TP1. Consultas TP 2
8	27 de abril	Segundo recuperatorio TP1. Consultas TP 2
9	4 de mayo	Consultas
10	11 de mayo	Entrega TP2. Presentación enunciado TP 3
11	18 de mayo	Primer recuperatorio TP2. Consultas TP 3
12	25 de mayo	Feriado
13	1 de junio	Segundo recuperatorio TP2. Consultas TP 3
14	1 de junio	Consultas
15	8 de junio	Entrega TP 3
16	22 de junio	Primer recuperatorio TP 3
17	29 de junio	Segundo recuperatorio TP 3