5/11/2019

Tarea 3 Entrega 1

Sistemas Operativos

Profesor : Claudio Alvarez

Francisco Alvarez

Este proyecto consiste en hacer una base de datos no relacional que usa la arquitectura cliente servidor a través del internet, así permitiendo múltiples conexiones simultáneas.

El sistema a usar consiste en un cliente que manda un mensaje con un comando a un servidor, y el servidor responde con un código asociado. Estos comandos y respuestas pueden llevar información adicional donde fuera necesario.

Comandos:

Comandos	**Parametros**	**Descripción**
Connect		Busca conectarse al servidor.
Disconnect		Se desconecta del servidor.
Insert	Key, Value	Inserta un valor en la clave dada.
Insert	Value	Inserta un valor sin especificar clave.
Get	Key	Busca un valor asociado a una clave.
Peek	Key	Busca si una clave esta disponible en el servidor.
Update	Key, Value	Actualiza el valor asociado a una clave.
Delete	Key	Elimina el valor asociado a una clave.
List		Busca una lista de todos los valores.

Respuestas:

Respuestas	Código	Descripción	Retorna
Connection Lost	00	Conexión con el servidor se ha perdido.	
Bad Request	01	Petición mal formada o mal escrita.	
Key Not Found	02	Clave valor no coincide con la base de datos.	
Key Taken	03	Clave valor ya existe en la base de datos.	
Connected	10	La conexión se ha generado con éxito.	
Inserted	20	Se ha insertado con éxito.	Clave
Got	30	Se ha buscado el valor con éxito.	Valor
Peeked	40	Se ha hecho "peek" con éxito.	Boolean
Updated	50	Se ha actualizado con éxito.	
Deleted	60	Se ha eliminado con éxito.	
List	70	Se ha encontrado la lista con éxito.	Lista
Accepted	100	La peticion ha sido aceptada, pero está en cola.	

Mensaje:

El mensaje mandado por el cliente debe tener un formato definido para que el cliente pueda interpretarlo de manera correcta en cada momento. Para este proyecto se eligió un formato basado en el usado en HTTP, pero mejor ajustado para el sistema.

Request Message	**Example1**	**Example2**
COMMAND (parameters)	GET (1337)	UPDATE(313, "Best")
Client Port	25565	3000
Server Port	80800	80800

Response Message	**Example1**	**Example2**
Code RESPONSE	30 GOT	02 KEY NOT FOUND
Client Port	25565	3000
Server Port	80800	80800
Return	"Elite"	

En este caso, cada fila es una nueva línea en el mensaje. Como podemos ver, este formato de mensaje permite enviar y recibir sin mala interpretación, y cualquier error en el envío será informado al cliente a través de un Bad Request. Además se envían los puertos necesarios para mantener la conexión entre los clientes y el servidor.

En el primer ejemplo, vemos como el cliente con puerto 25565 hace un GET, pidiendo el valor localizado en la clave “1337”. A esto, el servidor responde con 30 GOT y retorna el valor “Elite”. Por el otro lado vemos en el ejemplo 2 como el cliente intenta actualizar un valor con una clave inexistente, por lo que el servidor responde con 02 KEY NOT FOUND.