

TC2006 Lenguajes de Programación

Tarea 7: Programación Distribuida en Erlang

Fecha de entrega: martes 18 de noviembre de 2014

(* Equipos de 3 o 4 integrantes *)

Esta tarea les dará la oportunidad de practicar y familiarizarse con el lenguaje de programación Erlang.

Deberán subir como asignación de Blackboard, el archivo texto con nombre **M_tarea7.erl** (donde sustituyan M por sus matrículas separadas con guiones bajos) que contenga el código Erlang que implementa un prototipo sencillo de un sistema distribuido de inscripciones a eventos.

Internamente, mediante comentarios de Erlang deberán incluir sus matrículas y nombres en el archivo, la descripción de la interface de usuarios (clientes), los protocolos de comunicación entre clientes y servidores, y los protocolos de comunicación entre comandos y clientes.

FUNCIONAMIENTO DEL SISTEMA DE INSCRIPCIONES

El sistema de inscripciones debe tener usuarios (*clientes*) interesados en inscribirse a determinados eventos y un usuario (*organizador*) que pueden administrar los eventos. Además, debe tener un servidor de clientes (*registro*) que administre el manejo de clientes y sus solicitudes de inscripción, un servidor de eventos (*programador*) que administre el manejo de eventos, y un grupo de servidores de evento (uno por cada *evento*) que administren las inscripciones a cada uno.

Cada cliente se debe **registrar** (*nombre y contraseña*) antes de **entrar al sistema** de inscripciones (*log on*) en un nodo del sistema distribuido (una terminal distinta corriendo Erlang en la misma computadora). Solo se permitirá un cliente por nodo, de forma que se debe regresar un error adecuado si ya existe un cliente en el nodo donde trata de entrar otro cliente. Un cliente puede entrar al sistema en varios nodos distintos a la vez.

El cliente también debe poder mandar solicitudes de **inscripción a los eventos**, **obtener una lista de los eventos** en los que está inscrito y **salir del sistema** de inscripciones (*log off*). Regresar un error adecuado si el cliente no está dentro del sistema o si no se aceptó su solicitud de inscripción.

El organizador tiene el nombre especial **organizador** (Guauu!!) que le debe permitir **crear nuevos eventos** con cierta capacidad de asistentes, **eliminar eventos** existentes, **modificar capacidades** de eventos (aumentarlas o disminuirlas), obtener la **lista de todos los eventos** y **obtener la lista de inscritos** a un evento particular.

Un nodo en el sistema distribuido debe de ejecutar al **registro** (servidor de clientes) que administre la información de los clientes del sistema. A través de este

servidor, los nuevos clientes se registrarán, y los clientes ya registrados harán sus solicitudes. El registro también debe interactuar con el programador (servidor de eventos) y los servidores de eventos para atender las solicitudes de inscripción de los clientes. El registro debe responder con errores adecuados si algo evita que se pueda atender una solicitud. Errores tales como: nuevos clientes con nombres ya registrados, solicitudes de clientes no registrados, eventos inexistentes, inscripciones denegadas, etc.

Un nodo en el sistema distribuido (probablemente el mismo del registro) debe ejecutar al **programador** (servidor de eventos). El programador interactuará con el organizador para la creación, actualización y eliminación de eventos. El programador es el responsable de crear los servidores de eventos y de comunicarles las solicitudes de actualización realizadas por el organizador. También es el intermediario entre el registro y los servidores de eventos.

Los **servidores de eventos** se deben crear en un nodo (comúnmente el mismo del programador) y deben mantener actualizada la información sobre la capacidad del mismo y sus clientes inscritos.

El sistema debe estar preparado para manejar adecuadamente situaciones problema tales como:

- Eventos inexistentes
- Eventos llenos
- Solicitudes de clientes no registrados
- Creación de eventos que ya existen
- Caída de nodos donde residen los servidores (registro, programador y/o eventos)

Pareciera demasiado trabajo, pero no iniciarán de cero!! Les estoy anexando el documento *Getting Started with Erlang*, donde se explican las facilidades básicas sobre programación secuencial, concurrente y distribuida en Erlang. Además, se incluye un ejemplo sobre la programación de un sistema distribuido para mandar mensajes entre usuarios del sistema, que incluye los elementos que Ustedes requieren para realizar esta tarea. Tomen como base ese programa para escribir el suyo.

Espero que incluyan comentarios del mismo tipo que los incluidos en el ejemplo, de forma que describan las funciones de interface de usuarios, los protocolos de comunicación, la descripción de las funciones que incluyan, y finalmente, pero no menos importante, una secuencia de comandos que ilustre el uso de su sistema y todas sus características:

- creación de clientes, organizador, registro, programador y servidores de eventos
- registro y solicitudes de clientes
- creación y actualización de eventos
- manejo de errores

ÉXITO!!