

TP3: Servidor para múltiplos clientes

Prática de programação usando select e timers

Trabalho individual

Data de entrega: 24/11/2013 até às 23:59

Submeter para: esub.para.loureiro@gmail.com

Objetivo

Do ponto de vista funcional, o objetivo do trabalho é implementar um sistema de envio de mensagens de um servidor que atenda a múltiplas requisições. Do ponto de vista didático, o objetivo é o domínio da programação orientada a eventos utilizando-se a primitiva *select* e a temporização por sinais.

O problema

A conjectura de Beal é uma conjectura em teoria dos números que generaliza o último teorema de Fermat. Beal formulou a seguinte conjectura:

$$\text{Se } A^x + B^y = C^z,$$

onde A, B, C, x, y, e z são inteiros positivos, com x, y, z > 2, então A, B, e C possuem um fator primo comum.

Por exemplo: $3^3 + 6^3 = 3^5$. Nesse exemplo as bases tem fator primo comum 3.

Você tentará encontrar um contra-exemplo para a conjectura desenvolvendo um servidor e um cliente simples com troca de mensagens em C ou C++, utilizando comunicação via protocolo UDP.

O protocolo

Você desenvolverá um cliente que irá se registrar no servidor enviando uma mensagem contendo um identificador do cliente.

O servidor irá confirmar o recebimento da solicitação de registro respondendo com quatro valores inteiros *bmin*, *bmax*, *pmin*, *pmax*. Onde *bmin* e *bmax* representam os valores mínimo e máximo para as bases (A, B, C) e *pmin* e *pmax* representam os valores mínimos e máximos para os expoentes (x, y, e z).

O cliente irá responder o servidor confirmando os valores recebidos. O cliente deverá procurar por um contra-exemplo da conjectura no intervalo dado testando todos os valores possíveis. Ao final da procura, o cliente irá notificar o servidor se encontrou ou não um contra-exemplo no intervalo dado, e se encontrou, quais os valores de A, B, C, z, y, z.

Caso algum mensagem seja perdida, é necessário fazer a retransmissão.

Vários clientes serão executados simultaneamente. O servidor precisa atender a todos esses clientes simultaneamente. Para cada cliente que conectar ao servidor, o servidor irá responder com um valor intervalo diferente de bases ou expoentes ($bmin$, $bmax$, $pmin$, $pmax$). O servidor deve armazenar quais intervalos já foram procurados.

Um navegador poderá ser utilizado para gerência do servidor. O servidor deverá responder, utilizando um outro porto definido por você, uma página HTML que mostra quais os intervalos já foram procurados e contra-exemplos encontrados.

O que deve ser apresentado

Cada aluno deve entregar junto com o código um relatório que deve conter uma descrição da arquitetura adotada para o servidor, os refinamentos das ações identificadas no mesmo, as estruturas de dados utilizadas e decisões de implementação.

Descrição dos testes utilizados, com imagens da tela mostrando o correto funcionamento com múltiplos clientes e o retorno da página HTML de gerência do servidor.

O código fonte do servidor e dos clientes devem ser submetidos com comentários que identifiquem no código as partes do algoritmo de alto nível e expliquem os detalhes das partes principais do programa. Junto do código fonte, um arquivo **Makefile** deve ser incluído, de forma a facilitar a compilação do programa e um arquivo **README** que descreva como usar os programas cliente e servidor o conteúdo de cada arquivo submetido. Não inclua arquivos objeto (.o) nem executáveis no seu arquivo de entrega. Inclua imagens e arquivos de testes utilizados.