Engenharia Elétrica TE090 - Trabalho Sockets 2011/2

Pedroso

9 de dezembro de 2011

1 Introdução

O objetivo deste trabalho será realizar a implementação de um *servidor* Proxy HTTP básico que possa ser acessado por um browser comum (ex. Firefox, Google Chrome ou mesmo Internet Explorer), de modo a intermediar o acesso à outros servidores HTTP da Internet.

2 Descrição

O servidor Proxy HTTP a ser implementado deve pré-carregar o conteúdo solicitado pelo cliente para acelerar a resposta em futuros pedidos da mesma informação, que pode ser realizada por qualquer outro cliente. Funções adicionais que podem serem implementadas são: autenticação e implementação de *black-lists*.

O servidor deve ser implementado utilizando as rotinas disponíveis na biblioteca Sockets, utilizando a linguagem C para implementação. O servidor poderá ser *multithread*: quando um cliente inicia uma conexão o servidor deverá criar uma nova thread para atendê-la, de modo a não limitar o número de conexões simultâneas.

2.1 Requisitos básicos

Processar um pedido HTTP enviado pelo browser, buscando pela URL resultante em seu *cache* local. Se encontra, verifica a data e hora da versão da página com o servidor remoto. Se a informação não foi atualizada desde que foi gravada em cache, ela é enviada ao cliente imediatamente, caso contrário ou caso a informação não esteja no *cache*, o servidor Proxy irá contactar o servidor para obter a informação necessária, enviado a resposta ao cliente e armazenando na área de *cache*.

É necessário que o servidor Proxy processe pedidos com o protocolo HTTP. A especificação do protocolo HTTP é realizada pela

RFC2616 (ver http://tools.ietf.org/html/rfc2616 e o tutorial mais didático http://www.jmarshall.com/easy/http/). Para as transferências HTTP mais comuns, a sequência de mensagens é relativamente simples (é necessário ler a RFC, pelo menos o básico):

1. O cliente estabelece uma conexão TCP com o servidor.
2. O cliente envia uma requisição, composta de pelo menos uma linha de texto, para o servidor. Esta requisição consiste no método HTTP (frequentemente o método GET, solicitando a transferência de um arquivo; no entanto, são possíveis outros como POST, PUT, etc.). Juntamente com a requisição podem ser enviadas diversos campos separados por CRLF (*enter+linefeed*),
3. O servidor envia uma mensagem de resposta, com a linha inicial consistindo no *status*, indicando se a requisição teve sucesso. O status consiste em um código de resposta numérica (os códigos possíveis estão na RFC), e uma frase descrevendo a razão. Uma boa lista de códigos de resposta pode ser encontrada em http://en.wikipedia.org/wiki/List_of_HTTP_status_codes, e alguns exemplos do restante do cabeçalho HTTP podem ser encontrados em http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol.
4. O servidor envia a informação solicitada para o cliente.

Exemplo: Suponha que um servidor recebeu a seguinte mensagem:

```
GET /texto.txt HTTP/1.0
```

Isto indica que o cliente está solicitando o arquivo “texto.txt” do servidor. Caso a solicitação fosse feita simplesmente especificando “GET /”, o cliente está solicitando a página default (ou raiz) do servidor (normalmente o nome é *index.html*, mas isso depende do servidor)

A resposta do servidor deve ser a seguinte:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/plain
Content-Length: 42
some-footer: some-value
another-footer: another-value

abcdefghijklmnoprstuvwxyz1234567890abcdef
```

Note que a resposta foi enviada em duas partes: primeiro, iniciando com o código de resposta *HTTP/1.1 200 OK*, que é um código de sucesso - indicando que o servidor encontrou a informação desejada, seguindo-se pela data/hora e pelo *mime-tipe*, que neste caso especifica um arquivo texto, e o seu tamanho (campo *Content-Lenght*, que é de grande importância quando se está transferindo uma imagem. Ao cabeçalho HTTP, segue-se a infomação solicitada (no caso, *abc....def* é o conteúdo do arquivo).

Não será permitido o uso de bibliotecas (que não foram de autoria da equipe) que implementem total ou parcialmente as funcionalidades requisitadas. Caso seja constatado o uso indevido de bibliotecas o trabalho será considerado inválido.

3 Critérios de avaliação

O trabalho deve ser implementado em equipes de até 2 pessoas. Não serão admitidos trabalhos desenvolvidos por equipes maiores.

Será realizada uma defesa individual onde todos os componentes da equipe devem demonstrar conhecimento sobre a implementação apresentada. Caso um dos componentes não demonstre conhecimento sobre o trabalho implementado, sua nota será reduzida em proporção a sua participação no trabalho.

Em caso de cópias, as equipes envolvidas terão grau zero.

Para todos os trabalhos implementados, será considerado o seguinte critério para atribuição das notas:

Funcionalidade	Nota máxima
Requisitos básicos sem erros	100%
Requisitos básicos com erros que não impedem o funcionamento	75%
Requisitos básicos com erros que e funcionamento parcial	50%
Apenas requisitos básicos com erros que impedem o funcionamento	25%

4 Ponto Extra

Serão acrescentados pontos extras à nota geral como critério de criatividade, para equipes que apresentarem itens extras. Contactar o professor sobre a validade da idéia antes da implementação.