Servidor HTTP

Eduardo Veiga
Thiago Martins

Índice

- Introdução
  Objetivo
  Desenvolvimento
  Conclusão
  Referências

Introdução

- **Servidor:** Em informática, um servidor é um sistema de computação que fornece serviços a uma rede de computadores.
  - Esses serviços podem ser de natureza diversa, como por exemplo, arquivos e correio eletrônico.

Servidor HTTP: Um programa de computador responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagens, etc.).

Introdução

- O tratamento é feito por demanda: Usuários recebem o que querem, quando querem, se estiver disponível.
  HTTP define como clientes web requisitam páginas web aos servidores e como eles as transferem aos clientes.
  - O processo se inicia com a conexão entre o computador onde está instalado o servidor web e o computador do cliente.
  - A partir daí é processado o pedido do cliente, e conforme as restrições de segurança e a existência da informação solicitada, o servidor devolve os dados.

Objetivo

“Implementar um servidor HTTP multithread com cache em RAM, em Linguagem C e com a biblioteca Pthreads”

Desenvolvimento

- Detalhamento:

O servidor aqui descrito, foi implementado em C e serve 1 (e não mais) usuário sequencialmente e com taxa controlada.

Desenvolvimento

- O Servidor serve corretamente os clientes quando lhe é solicitado os seguintes tipos de arquivo:
  - .pdf (aplicação)
  - .png (imagem)
  - .txt (texto)
  - .gif(imagem)
  - .jpeg(imagem)
  - .jpg(imagem)
  - .html (arquivo base de página web)

  Não foi implementado, todos tipos disponíveis de também grande utilização. (ex: mp3, flash, doc, ppt, etc).

Desenvolvimento

- O Cliente é responsável, somente pela requisição de um arquivo.
  - Ex: Telnet, utilizando o padrão a seguir:
    - Telnet 192.168.0.1 8000
      GET /home/usuario/docs/redes.pdf HTTP/1.1
  - Ex: Browser, utilizando uma URL válida de um arquivo, na central de arquivos do Servidor.
  - Onde o Browser é responsável por enviar requisições diretamente ao Servidor HTTP.
    192.168.0.1:8000/home/usuario/docs/redes.pdf

Desenvolvimento

- Não foi implementado a utilização de Threads para tratamento de serviço de multiusuários.
  O servidor executa em um único fluxo.
  É servido apenas um cliente, onde a conexão permanece ativa caso ele necessite de outra requisição.

Desenvolvimento

Esse cliente é servido á uma determinada taxa, pré-estabelecida por uma variável estática global.

- #define TAXA 1 //taxa de 1 MB
  #define TAM_MSG 1025 //número de bytes servidos

Desenvolvimento

Para o desenvolvimento da criação e gerenciamento de sockets, foi utilizado a biblioteca:

- socket.h

Para a conexão em si, foi utilizado a seguinte biblioteca:

- inet.h

Desenvolvimento

Validação:

- Foram realizados experimentos de envio de arquivo em 2 tipos diferentes de Browsers: Mozilla Firefox e Google Chrome.
- Ambos em ambiente Linux (Ubuntu).
- Foi obtido êxito em ambos.

Desenvolvimento

- O tratamento de toda mensagem (no caso de Browsers) de requisição é seqüencial, ou seja, é tratado cada linha da mensagem de requisição de uma vez.
  Ex:
  - Como a primeira linha da mensagem de requisição de um browser é o método GET, o servidor trata diretamente esse método, sem se preocupar com os outros cabeçalhos que possam de alguma forma serem úteis.

Desenvolvimento

- O servidor só trata o Método GET. Não foi implementado os outros métodos HTTP como: POST, HEAD, PUT, DELETE, etc.

Desenvolvimento

- Para melhor gerenciar o arquivo, foi utilizado uma biblioteca chamada sys/stat.h, onde quando um arquivo é aberto utilizando-a, ela já carrega todos os dados necessários para um bom tratamento de mensagem HTTP.
  Ex:
  - Data da ultima modificação.
  - Tamanho de arquivo.

Desenvolvimento

Execução (linux):

- No terminal digite:
  - gcc –o servidor servidorHTTP.c

    ./server 8000 100

    8000 = porta de acesso ao socket do servidor.

    100 = provável taxa máxima de vazão do servidor.

Conclusão

- Mediante o trabalho proposto, que não é de grande dificuldade, acreditamos que ficamos devendo, pois é de nossa responsabilidade saber gerenciar melhor nosso tempo a fim de que tudo que nos é solicitado possa ser de alguma forma bem atendido.

Referências