



Redes de Computadores - 2014.2

Trabalho 1 Servidor Web

Profa. Hana Karina S. Rubinsztein

Descrição:

Este trabalho tem por objetivo a implementação da funcionalidade básica de um servidor Web. O servidor deve permitir que clientes HTTP (Firefox, IE, etc.) se conectem ao servidor e façam *downloads* de arquivos.

O servidor para o protocolo HTTP deve implementar os métodos de consulta GET e POST usando os protocolos TCP/IP para transferência das páginas HTML.

Como funciona o protocolo HTTP

A implementação esperada deve ser baseada no HTTP 1.1 (RFC 2616), em que cada consulta contém algumas informações na requisição feita pelo cliente e na resposta esperada pelo servidor.

Um cabeçalho padrão de requisição feita por um navegador de internet pode ser algo como:

```
GET /somedir/page.html HTTP/1.1
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language:pt_br
```

Nesse cabeçalho temos informações que são consideradas pelo servidor para trazer o *site* desejado.

A função do servidor HTTP é analisar a requisição acima, identificar o arquivo sendo solicitado e enviar o arquivo pela rede para o cliente. Entretanto, antes de enviar o



documento, o servidor HTTP deve enviar um cabeçalho de resposta para o cliente. Uma resposta típica de um servidor HTTP pode ser vista abaixo:

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html
```

```
data data data data data ...
```

Neste caso o arquivo foi encontrado no servidor (200) . A seguir estão as respostas que devem estar implementadas no seu programa:

- **200 OK:** A Requisição foi bem sucedida e a informação foi retornada ao requisitante.
- **301 Moved Permanently** (Movido Permanentemente): O objeto requisitado foi movido permanentemente; a nova URL é especificada no campo `Location` do cabeçalho de resposta da mensagem. O cliente será redirecionado automaticamente para a nova URL.
- **400 Bad Request** (Má requisição): Um código de erro genérico para uma requisição que não é entendida pelo servidor.
- **404 Not Found** (Não encontrado): O documento requisitado não existe neste servidor.
- **505 HTTP Version Not Supported** (Versão de HTTP não suportada): A versão do protocolo de requisição não é suportada por este servidor.

Ao executar o servidor deve-se indicar o número da porta em que ele receberá conexões. Se nenhuma porta for especificada, o servidor deve escutar na porta 8080.

Método GET (4 pts)

Seu servidor deve ter suporte ao método GET para requisitar sites armazenados no diretório que conterá sites na linguagem HTML. Deve ser implementado um site em HTML: `index.html`, contendo um pequeno texto e uma foto.

Método POST (2pts)

Seu servidor deve ter suporte ao método POST para enviar dados de formulários para páginas WEB. Deve ser implementada uma página de exemplo para testes, chamada de `post.html` que contenha o formulário. Ao tratar essa página o



servidor deve retornar outra página como resposta contendo as informações enviadas no formulário.

Dica: Veja um exemplo em http://www.w3.org/MarkUp/html-spec/html-spec_8.html#SEC8.2

Diretórios (1 pt)

Você deve adicionar ao seu servidor a capacidade de navegação em diretórios. Se o documento requisitado for um diretório, seu servidor HTTP deve retornar um documento HTML com *links* para os arquivos/diretórios presentes no diretório. Você deve permitir navegação recursiva de diretórios. Estudem as páginas de manual das funções `opendir` e `readdir`.

Concorrência (2pt)

O servidor deve ser capaz de atender vários clientes simultaneamente. Assim deve ser implementado um modelo de concorrência, seja usando processos ou *threads*. Todos os tratamentos devem ser realizados de forma a não deixar filhos *zumbis*, ou *threads* paradas.

Arquivo de Log (1pt)

Seu servidor deve manter um registro das requisições atendidas, indicando o IP do cliente solicitante, o método e objeto requisitado e a data e hora.

Entrega do Trabalho:

O trabalho deverá ser submetido eletronicamente utilizando o **moodle**. O prazo de entrega se encerra no dia **23 de outubro às 23:55h**. O sistema de submissão será bloqueado nesse horário e você não poderá submeter por outro meio.

Cada grupo (de até 3 participantes) deve entregar um arquivo tipo zip ou tar.gz, contendo os arquivos do trabalho (.c, .cpp, .h, **makefile** e etc...) e um relatório. Esse relatório deve conter o nome dos autores e descrever como o servidor Web foi implementado; uma seção descrevendo se tudo que foi especificado/solicitado no o trabalho foi ou não implementado; e, por fim, outra seção com as funcionalidades extras que não foram solicitadas e foram implementadas.

Os grupos irão apresentar o trabalho no dia 24 de outubro, durante o horário da aula de laboratório.



Observações:

- a) Seu programa deve compilar no Linux utilizando gcc;
- b) Comente todas as linhas do seu código;
- c) Na entrega não inclua os arquivos objeto (.o) nem os executáveis utilizados!