

CI244 - Fundamentos de Redes de Computadores

Sistemas Distribuídos

– Trabalho Prático*–

Prof. Michele Nogueira
michele@inf.ufpr.br

¹Departamento de Informática - NR2/UFPR

Data de Apresentação do Projeto: 20 e 21 de Junho de 2012.
Trabalhos fora dessas datas não serão aceitos!

1. Apresentação

Neste projeto, você irá criar um servidor web mínimo para servir arquivos de texto e html. O seu servidor terá um único componente dinâmico para listar os arquivos de um diretório. Para isto, ele precisará ler e enviar o conteúdo dos arquivos para os clientes, assim como aceitar dados de entrada.

Como o HTTP funciona?

O HTTP é um protocolo que segue o modelo cliente e servidor cuja comunicação entre este é realizada via requisição-resposta. Toda comunicação ocorre sobre uma conexão TCP (Transport Control Protocol). O servidor ouve em uma porta TCP, e aceita novas conexões. O cliente se conecta ao servidor e envia uma requisição HTTP. O servidor responde com uma resposta HTTP. Para este projeto, o seu servidor Web deve servir uma única requisição HTTP por conexão (um protocolo HTTP completo suporta múltiplas requisições por conexão). As requisições e respostas HTTP são em ascii. Abaixo, você encontra um exemplo de uma requisição HTTP recebida pelo servidor.

```
1. GET / HTTP/1.1
2. User-Agent: curl/7.18.0 (i486-pc-linux-gnu) libcurl/7.18.0 OpenSSL/
0.9.8g zlib/1.2.3.3 libidn/1.1
3. Host: google.com
4. Accept: */*
5.
```

Figura 1: Requisição a um servidor Web

A linha 1 dessa requisição contém o método HTTP (get), a requisição URI (/), e a versão do HTTP (HTTP/1.1). Para este projeto, o seu servidor web pode ignorar todos os outros campos do cabeçalho. Note que a requisição é finalizada por uma linha em branco (linha 5). Isto é usado para indicar ao servidor que o cliente terminou de enviar o cabeçalho HTTP.

```
1. HTTP/1.1 200 OK
2. Date: Wed, 29 Oct 2008 03:24:19 GMT
3. Server: Apache/2.2.8 (Fedora) DAV/2 mod_pubcookie/3.3.3 mod_ssl/2.2.8
   OpenSSL/0.9.8g
4. Accept-Ranges: bytes
5. Transfer-Encoding: chunked
6. Content-Type: text/html
7. Content-Language: en
8.
9. ... HTTP CONTENT ...
...
```

Figura 2: Resposta do Servidor Web

A resposta do servidor se assemelha ao que segue:

A linha 1 desta resposta contém a versão do protocolo HTTP (HTTP/1.1), o código de estado (200), e frase explicativa (OK). Em seguida, são encontradas algumas respostas. Para este projeto, seu servidor web não deve gerar qualquer resposta nos campos do cabeçalho. Note que como a requisição HTTP, a resposta é também finalizada com uma linha em branco (a linha 8). As linhas que seguem imediatamente a linha em branco (linha 9+) contém o conteúdo HTTP.

Seu servidor pode assumir que o cliente envia uma requisição HTTP única, e pode fechar a conexão TCP depois de enviar a resposta HTTP ao cliente.

Requisitos do projeto

Um servidor Web que ...

- Tem um único argumento de porta na linha de comando
- Funciona com firefox
- Suporta versões HTTP 1.0 e 1.1
- Serve arquivos fora do diretório corrente
- Mostra uma página listando os arquivos disponíveis de uma URI '/'

Por "funciona em firefox", eu quero dizer que se o servidor está executando na porta X e existe um arquivo onde o servidor estava executando F.txt, então digitando "http://server:X/F.txt" no firefox, ele deveria carregar o conteúdo do arquivo F.txt na janela. Seu servidor web precisa garantir apenas três códigos de estado e as suas respectivas frases explicativas:

- 200 OK
- 400 Bad Request
- 404 Not Found

O servidor web mínimo irá (1) ler a linha de requisição: método de requisição, requisição URI, versão do HTTP, (2) esperar encontrar uma linha em branco, (3) responder com uma mensagem de erro 'Bad Request' se uma requisição inválida/incompleta

*Este trabalho foi especificado com base no 'Exercício de Web Server' disponível no: ACM SIGCOMM education resources website.

é recebida; (4) fechar a conexão se o cliente expeira o tempo (depois de 5 segundos); (5) responder com o conteúdo do arquivo requisitado (ou a lista de conteúdo do diretório) usando uma mensagem de resposta HTTP 1.0 válida e (6) retornar o código de estado 200, 400 ou 404.

Servindo arquivos

O seu servidor Web deve servir arquivos do diretório corrente. Isto é, quando você executar o programa servidor web, o diretório corrente do programa será algum diretório (ex. um subdiretório do diretório home). Se o servidor recebe uma requisição como "GET /hello.html HTTP/1.0", ele deve retornar o arquivo "hello.html" que está localizado no diretório corrente de trabalho do servidor web. Para esta projeto, assume-se que o servidor web serve somente arquivos HTML e texto pleno.

2. Ambiente do projeto

1. Faça o download dos arquivos de referência através da URL: `www.nr2.ufpr.br/~michele/teaching/S1_2012/CI244/sockets.tar`
2. Descompacte o arquivo sockets.tar através do comando: `tar -xvz sockets.tar`
3. Você deve primeiro entender o que os arquivos c-server.c e c-client.c fazem. Para compilá-los e testá-los siga os passos da seção 'compilando e executando exemplo'.
4. Você deve tomar como base estes dois arquivos e adaptá-lo para criar seu servidor web como especificado anteriormente. Quando falo adaptar significa alterar o código, excluir partes no código, adicionar novas partes, etc.

Começando

Verifique também este tutorial:

Sockets

Compilando e executando exemplo

Para compilar os arquivos 'c-server.c' e 'c-client.c' siga os passos:

1. Abra 2 terminais (nós chamaremos nesta descrição de TERMINAL 1 e TERMINAL 2)
2. Em um dos dois terminais, entre no diretório em que os arquivos c-server.c e c-client.c estão armazenados.
3. Para compilar dê os comandos: `'gcc c-client1.c -o c-client1'` e `'gcc c-server1.c -o c-server1'`
4. Para executar o servidor, dê o seguinte comando no TERMINAL 1: `'./c-server1'`, após dê o seguinte comando no TERMINAL 2 para iniciar o cliente: `'./c-client1'`

3. Informações finais

- O projeto pode ser feito em pares (OBS: max. 2 pessoas)
- Pode ser feito usando seu computador
- O projeto vale de 0 a 100 pontos
- Se necessário, você pode trazer seu computador para a apresentação do projeto
- Os critérios de avaliação serão divulgados em breve.