

Exercício Programa Mini Web Server

Volnys Borges Bernal

Depto. de Engenharia de Sistemas Eletrônicos
Escola Politécnica da USP



Mini Web Server

□ Objetivo

- ❖ Desenvolvimento de um programa mini servidor WEB

Mini Web Server

□ Descrição

- ❖ Desenvolver um mini servidor WEB
- ❖ Servidor TCP concorrente multithreaded
- ❖ Deve aguardar requisições na porta 8080
- ❖ Configurações da instalação:
 - Armazenadas em um arquivo de configuração.
 - Este arquivo deve ser lido no início da execução
 - Contém:
 - PORTA: Número da porta TCP (ex: 8080)
 - BASE: Caminho do diretório “base” da hierarquia de objetos a ser apresentada pelo servidor

Mini Web Server

❑ Funcionalidades a serem suportadas:

- ❖ Requisições e respostas HTTP
 - Versão: HTTP 1.0
 - Métodos: HTTP GET
- ❖ Objetos:
 - Páginas HTML
 - Arquivos texto
 - Imagens no formato JPEG e PNG
- ❖ Quando a URL referenciar um objetivo “diretório”:
 - Se no diretório existir o arquivo index.html, retornar este arquivo
 - Caso contrário, gerar a página html do diretório com hiperlinks.

Mini Web Server

□ Detalhamento

- ❖ Implementar, no mínimo, os seguintes códigos de resposta HTTP:
 - 200 OK
 - sucesso, objeto pedido segue mais adiante nesta mensagem
 - 400 Bad Request
 - mensagem de pedido não entendida pelo servidor
 - 404 Not Found
 - documento pedido não se encontra neste servidor
 - 505 HTTP Version Not Supported
 - versão de HTTP do pedido não usada por este servidor

Mini Web Server

- **Ambiente e linguagem**
 - ❖ Ambiente Linux
 - ❖ Linguagem C
 - ❖ Biblioteca pthreads
 - ❖ Interface sockets

Dicas



Mini Web Server

□ URL

- ❖ O formato simplificado de URL:

- Utilizado no protocolo HTTP (RFC 2068)

- ❖ Sintaxe:

- ❖ `http_URL = "http:" "://" host [":" port] [abs_path]`

- **host** = nome do host (nome DNS) ou endereço IP
- **Port** = opcional, valor numérico da porta (default 80)
- **abs_path** = opcional, caminho absoluto,
- se não presente, deve utilizar o diretório raiz "/"

Dicas

□ Sintaxe da URL

- ❖ A sintaxe relacionada ao uso mais comum da URL é:
 - `scheme://domain:port/path`
 - sendo:
 - Scheme: equivale ao protocolo (ex: HTTP)
 - Domain: DNS hostname ou um endereço IP
 - Port: Porta do serviço (porta do protocolo da camada de transporte)
- ❖ Porém, a sintaxe geral da URL é:
 - `scheme://username:password@domain:port/path?query_string#anchor`
 - sendo:
 - Query: para programas tipo CGI
 - Anchor: geralmente utilizado em HTTP, indica o ponto de início para ser apresentado.
- ❖ Exemplo: `http://example.org:8080/test/mypage.html`
 - Scheme: “http”
 - Domain: “example.org”
 - Port: “8080”
 - Path: “test/mypage.html”

Dicas

□ Base da hierarquia de páginas

- ❖ Servidor deve permitir a configuração do diretório “diretório base”.
- ❖ Os objetos apresentados pelo Servidor WEB são aqueles presentes a partir do “diretório base” (variável BASE).
- ❖ Exemplo de uma hierarquia de diretórios e arquivos:

```
/
  home
    jose
      programas
      ....
      listador
        principal.htm
        projeto (diretorio)
          nomes.txt
          logo.jpeg
        teste.htm
```

- BASE=/home/jose/webserver

Dicas

❑ Função transferfile()

- ❖ Para transferência do arquivo utilize a função transferfile():

```
int transferfile(char *path,int output_fd)
```

- ❖ A função transferfile realiza a leitura do conteúdo de um arquivo, identificado por seu caminho (path), transferindo seu conteúdo para outro arquivo ou socket identificado pelo descritor de arquivos "output_fd".

Dicas

❑ Função lista_diretório:

❖ Para listar o diretório utilize a função lista_diretório():

```
void lista_diretorio(char *path, char *buffer, int buffersize)
```

❖ Sendo:

- path: diretório a ser listado
- buffer: conterá string com a seqüência ASCII
- buffersize: tamanho do buffer
- resultado da listagem do diretório (finalizada por '\0')