


© 2004-2014 Volnys Bernal 1

Exercício Programa Mini Web Server

PSI 2653
Meios Eletrônicos Interativos I



© 2004-2014 Volnys Bernal 2

Mini Web Server

- ❑ **Objetivo**
 - ❖ Desenvolvimento de um programa servidor WEB
- ❑ **Composição do grupo**
 - ❖ 4 pessoas (obrigatório)
- ❑ **Formato do trabalho**
 - ❖ Papel A4, folhas grampeadas (não encadernar!!)
 - ❖ Página de rosto informando:
 - Nome da disciplina, título do trabalho e nome dos autores
- ❑ **Entrega:**
 - ❖ Data entrega: 26 de junho
 - ❖ Entrega do trabalho escrito em sala de aula
 - ❖ Execução do programa em sala de aula

© 2004-2014 Volnys Bernal 3

Mini Web Server

- ❑ **Descrição**
 - ❖ Desenvolver um mini servidor WEB
 - ❖ Servidor TCP concorrente multithreaded
 - ❖ Deve aguardar requisições na porta 8080
 - ❖ Configurações do programa "mini web server":
 - As configurações do programa "mini web server" devem ser armazenadas em um arquivo de configuração, a ser lido no início de sua execução, que deve conter, no mínimo:
 - PORTA: Número da porta de trabalho (ex: 8080)
 - BASE: Caminho do diretório "base" da hierarquia de objetos a ser apresentada pelo servidor

© 2004-2014 Volnys Bernal 4

Mini Web Server

- ❑ **Funcionalidades a serem suportadas:**
 - ❖ Requisições e respostas HTTP
 - Versão: HTTP 1.0
 - Métodos: HTTP GET
 - ❖ Objetos:
 - Páginas HTML
 - Arquivos texto
 - Imagens formato JPEG (JPG)
 - ❖ Listagem de diretório
 - Listar todas as entradas do diretório, incluindo a entrada para voltar ao diretório pai ("..")
 - Configurar hyperlink em cada entrada do diretório listado

© 2004-2014 Volnys Bernal 5

Mini Web Server

- ❑ **Funcionalidades a serem suportadas:**
 - ❖ Caso a URL referencie um diretório que contenha o arquivo "index.html" este deve ser retornado, ao invés da listagem do diretório.
 - ❖ URL: O mini web server deve suportar o seguinte formato de URL:
 - http://host:port/path

© 2004-2014 Volnys Bernal 6

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


© 2004-2014 Volnys Bernal 7

Mini Web Server

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

© 2004-2014 Volnys Bernal 8

Dicas



© 2004-2014 Volnys Bernal 9

Mini Web Server

- ❑ **URL**
 - ❖ O formato de URL utilizado no protocolo HTTP (RFC 2068) possui a seguinte 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 "/"

© 2004-2014 Volnys Bernal 10

Dicas

- ❑ **Dica: Desenvolva seu projeto em etapas:**
 - ❖ 1ª etapa
 - Modificar o servidor *echo* TCP *multithreaded* que responde com um objeto fixo (um texto fixo, formato txt, sem cabeçalho HTTP da resposta).
 - Teste:
 - Utilizar um browser WEB para testar o servidor
 - Responder cada requisição com o objeto fixo.
 - ❖ 2ª etapa
 - Implementação de decodificação de requisições HTTP GET
 - Implementação de decodificação de URL, para extrair o *path*
 - Busca de arquivos a partir de um "diretório base"
 - Teste:
 - Utilizar um browser WEB para testar o servidor
 - Responder somente páginas tipo texto (.txt) sem se preocupar com o cabeçalho HTTP da resposta
 - ❖ 3ª etapa
 - Identificar o tipo de objeto requisitado: texto, html, imagem ou diretório
 - Gerar cabeçalho de resposta HTTP
 - ❖ 4ª etapa
 - Gerador de páginas HTML referente a listagem de diretório.

© 2004-2014 Volnys Bernal 11

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:80/test`
 - Scheme: "http"
 - Domain: "example.org"
 - Port: "80"
 - Path: "test"

© 2004-2014 Volnys Bernal 12

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')