

© 2004-2011 WolnyBernal 1

Exercício Programa Mini Web Server

PSI 2653
Meios Eletrônicos Interativos I



© 2004-2011 WolnyBernal 2

Mini Web Server

- ❑ **Objetivo**
 - ❖ Desenvolvimento de um programa servidor WEB
- ❑ **Composição do grupo**
 - ❖ 5 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: 16 de junho
 - ❖ Entrega do trabalho escrito em sala de aula
 - ❖ Execução do programa em sala de aula
 - ❖ Serão descontados 2 pontos da nota para cada dia de aula em atraso

© 2004-2011 WolnyBernal 3

Mini Web Server

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

© 2004-2011 WolnyBernal 4

Mini Web Server

- ❑ **Detalhamento**
 - ❖ Servidor TCP concorrente multithreaded
 - ❖ Deve aguardar requisições na porta 8080
 - ❖ Configuração da base da hierarquia de páginas
 - O programa deve possuir um arquivo de configuração que contém o caminho do "diretório base" das páginas oferecidas

© 2004-2011 WolnyBernal 5

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
 - ❖ Integração com uma Câmera USB para geração de imagens JPG

© 2004-2011 WolnyBernal 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

© 2004-2011 WolnyBernal 7

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** = DNS domain name ou endereço IP
 - **Port** = valor numérico, se a porta for vazia, porta 80 é assumida
 - **abs_path** = caminho, se abs_path não estiver presente na URL, deve ser utilizado "/"

© 2004-2011 WolnyBernal 8

Dicas



© 2004-2011 WolnyBernal 9

Dicas

- Dica: Desenvolva seu projeto em etapas:
 - ❖ 1ª etapa
 - Modificar o servidor `echoTCP multiThreaded`
 - Teste:
 - Utilizar um browser WEB para testar o servidor
 - Responder cada requisição com um texto fixo, formato txt, sem cabeçalho HTTP da resposta.
 - ❖ 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.
 - ❖ 5ª etapa
 - Integração com programa USB para gerar imagens em um determinado diretório.

© 2004-2011 WolnyBernal 10

Dicas

- Sintaxe da URL
 - ❖ A sintaxe relacionada ao uso mais comum da URL é
 - `scheme://domain:port/path`
 - Onde:
 - Scheme: equivale ao protocolo
 - 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`
 - Onde:
 - 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"

© 2004-2011 WolnyBernal 11

Dicas

- Base da hierarquia de páginas
 - ❖ A partir do "diretório base" (BASE) deverão ser depositados os arquivos
 - ❖ 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

© 2004-2011 WolnyBernal 12

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 "outfd".

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')