

Dicas para implementação do mini servidor Web

PSI 2653
Meios Eletrônicos Interativos I

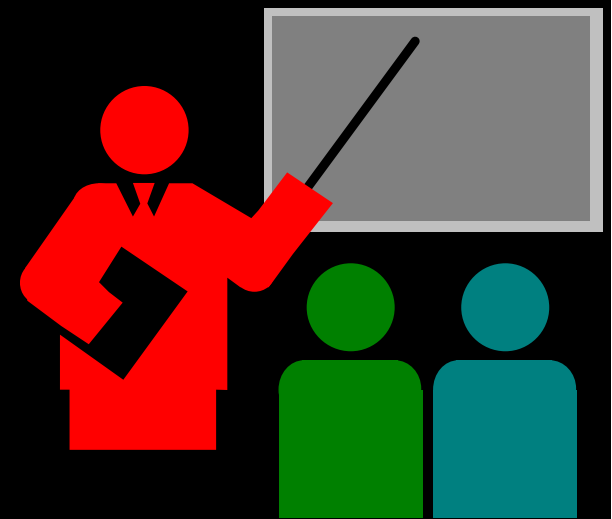


Dicas para implementação do mini servidor Web

- ❑ **As dicas aqui presentes são sugestões para facilitar a implementação do programa. Não é obrigatório sua utilização**

- ❑ **Sumário**
 - ❖ Diretório base do servidor WEB
 - ❖ Identificação da página a ser buscada
 - ❖ Chamadas ao sistema e rotinas biblioteca

Diretório base do servidor WEB



Diretório base do servidor WEB

❑ Base da hierarquia de páginas

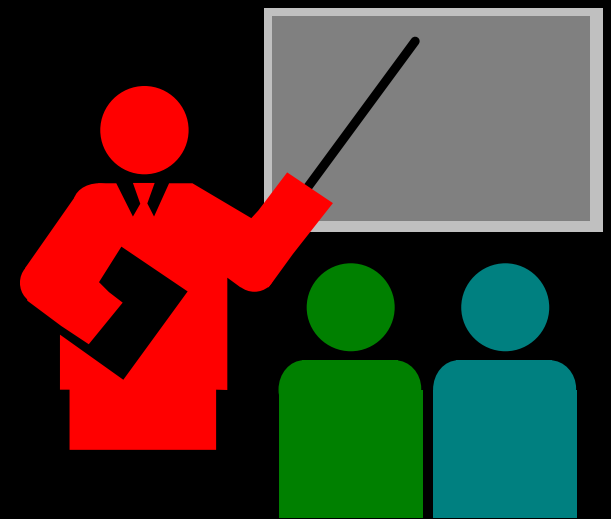
- ❖ Deve ser configurado no “Mini Servidor WEB” uma variável que deve conter o “diretório base” do webserver (variável BASE)
- ❖ A partir do “diretório base” (BASE) deverão ser depositadas as páginas HTML
- ❖ Quando não for informada uma página deve ser mostrada a página “index.html”

❖ Exemplo:

```
/
  home
    lumiar
      BaseWeb
        index.html
        homepage.html
        hobbies.html
```

BASE=/home/lumiar/BaseWeb

Identificação da página a ser buscada



Identificação da página a ser buscada

□ Identificação da página html (ou diretório) a ser transferido

- ❖ Aguardar mensagem HTTP
- ❖ Decodificar a linha de pedido (1ª linha da mensagem HTTP)
- ❖ Se for comando GET então
 - Obter URL
 - Extrair caminho da URL
 - Armazenar na variável CAMINHO
- ❖ Compor nome do arquivo/diretório
 - ARQUIVO = BASE / CAMINHO
- ❖ Exemplo:

```
GET /somedir/page.html HTTP/1.0
```

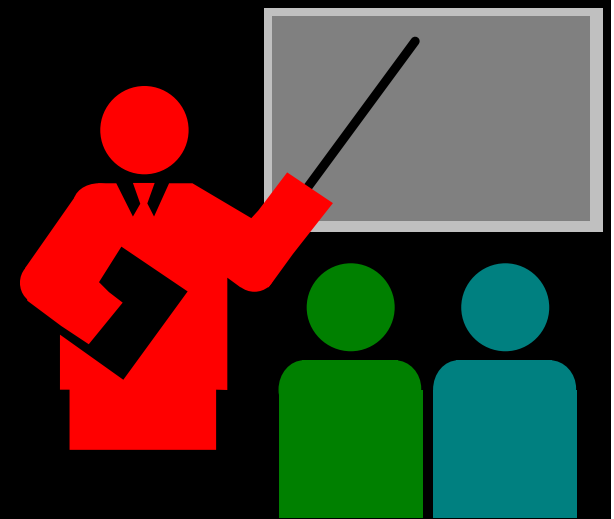
```
User-agent: Mozilla/4.0
```

```
Accept: text/html, image/gif, image/jpeg
```

```
Accept-language: fr
```

- BASE = /home/jose/webserver
- CAMINHO = /somedir/page.html
- ARQUIVO = /home/jose/webserver/somedir/page.html

Chamadas ao sistema e Rotinas de biblioteca



Chamadas ao sistema e Rotinas de biblioteca

❑ Manuais on line (utilitário man)

❖ Chamadas ao sistema

- São descritas na secção 2 do manual on-line do Unix
- Exemplo:
 - `open(2)`, `read(2)`, `write(2)`, ...
- Manual on-line
 - `man 2 tópico`

❖ Rotinas de biblioteca

- São descritas na secção 3 do manual on-line do Unix
- Exemplo:
 - `opendir(3)`, `readdir(3)`, `getcwd(3)`, ...
- Manual on-line
 - `man 3 tópico`

Chamadas ao sistema e Rotinas de biblioteca

□ **fstat(2)**

- ❖ Permite obter propriedades de um arquivo como, por exemplo, tamanho, tipo (diretório, normal, especial, ...)
- ❖ Sintaxe

```
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
int fstat(int filedes, struct stat *buffer);
```
- ❖ Para mais detalhes
 - `man 2 fstat`

Chamadas ao sistema e Rotinas de biblioteca

❑ opendir(3)

- ❖ Abre um diretório

- ❖ Sintaxe

 - #include <sys/types.h>

 - #include <dirent.h>

 - DIR *opendir(const char *name);

- ❖ Para mais detalhes

 - man 3 opendir

Chamadas ao sistema e Rotinas de biblioteca

❑ readdir(3)

- ❖ Lê 1 entrada de um diretório
- ❖ Sintaxe

```
#include <sys/types.h>
#include <dirent.h>
struct dirent *readdir(DIR *dir);
```
- ❖ Para mais detalhes
 - man 3 readdir

Chamadas ao sistema e Rotinas de biblioteca

□ **open(2)**

- ❖ Abre um arquivo
- ❖ Sintaxe

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

int open(const char *pathname, int flags);
int open(const char *pathname, int flags, mode_t mode);
```
- ❖ Exemplo:
 - Abre para leitura: `fd = open(path,O_RDONLY);`
 - Abre para escrita: `fd = open(path,O_WRONLY|O_TRUNC);`
- ❖ Para mais detalhes
 - `man 2 open`

Chamadas ao sistema e Rotinas de biblioteca

❑ read(2)

- ❖ Lê um arquivo ou socket

- ❖ Sintaxe

 - `#include <unistd.h>`

 - `int read(int fd, void *buffer, int count);`

- ❖ Para mais detalhes

 - `man 2 read`

Chamadas ao sistema e Rotinas de biblioteca

❑ **write(2)**

- ❖ Escreve em um arquivo ou socket

- ❖ Sintaxe

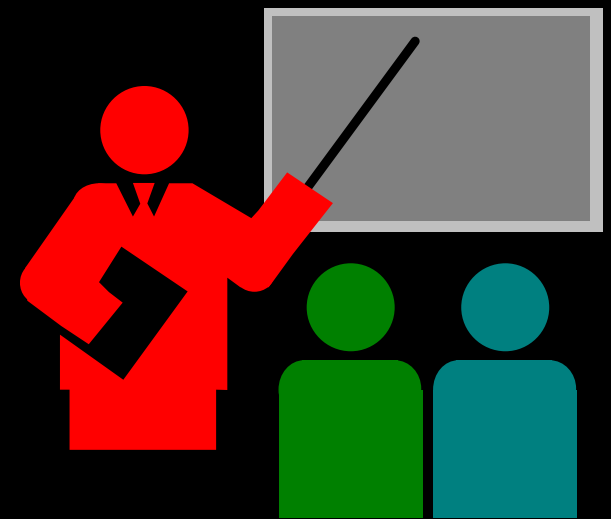
 - `#include <unistd.h>`

 - `int write(int fd, void *buffer, int count);`

- ❖ Para mais detalhes

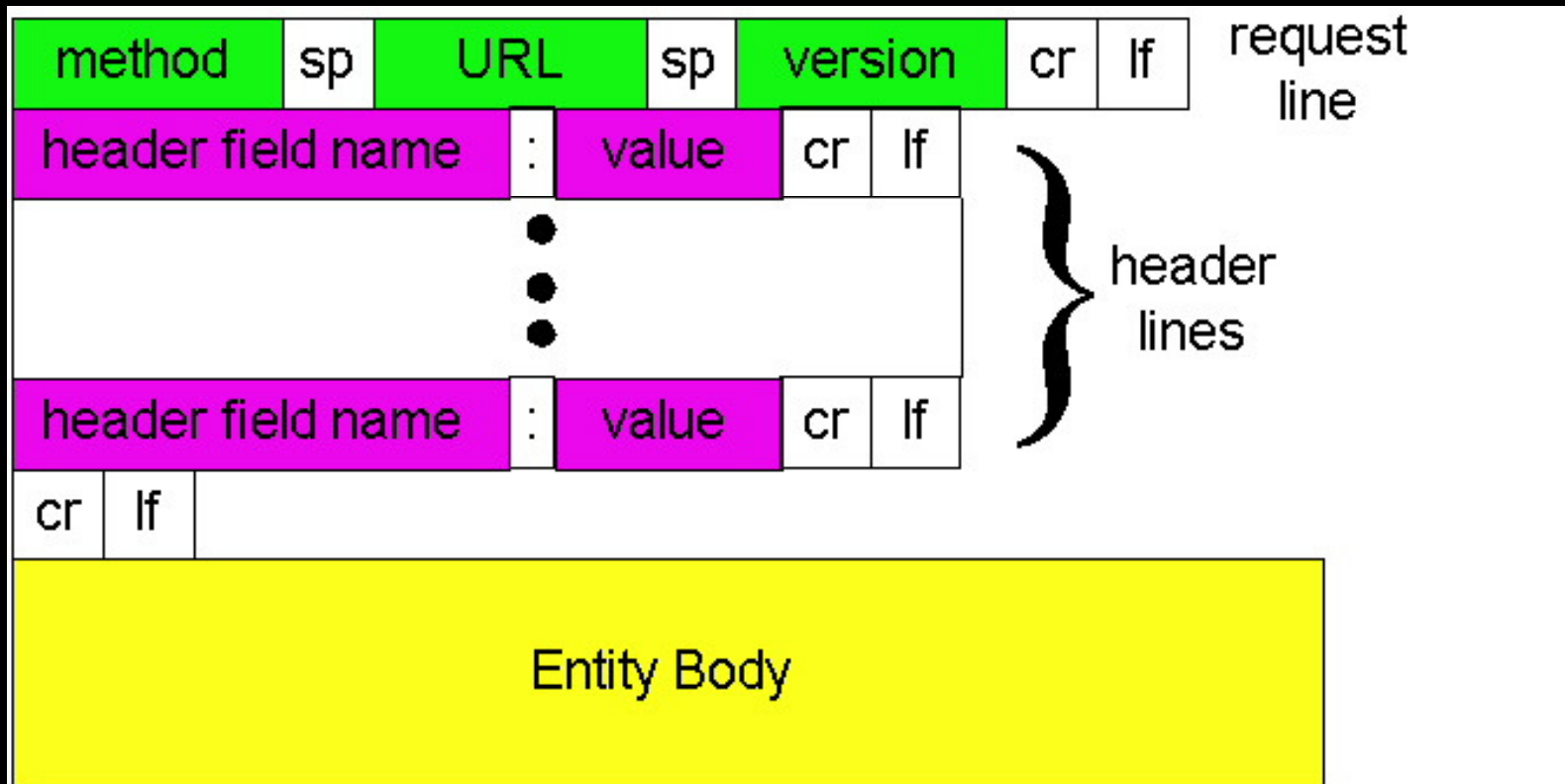
 - `man 2 write`

Dicas sobre HTTP



Dicas sobre HTTP

❑ Mensagem de pedido HTTP



Dicas sobre HTTP

❑ Pedido HTTP

linha do pedido
(comandos GET,
POST, HEAD)

linhas do
cabeçalho

```
GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
```

Carriage return,
line feed
indicam fim
de mensagem

(carriage return (CR), line feed(LF) adicionais)

Dicas sobre HTTP

□ Resposta HTTP

linha de status
(protocolo,
código de status,
frase de status)

linhas de
cabeçalho

```
HTTP/1.0 200 OK
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 .....
Content-Length: 6821
Content-Type: text/html
```

dados, p.ex.,
arquivo html
solicitado

```
dados dados dados dados ...
```

Dicas sobre HTTP

❑ Resposta HTTP GET

- ❖ Linha de estado da resposta
- ❖ Cabeçalho
 - Cabeçalho HTTP
- ❖ Corpo da mensagem
 - Página HTML

Dicas sobre HTTP

❑ Exemplo de geração da pagina HTML de resposta

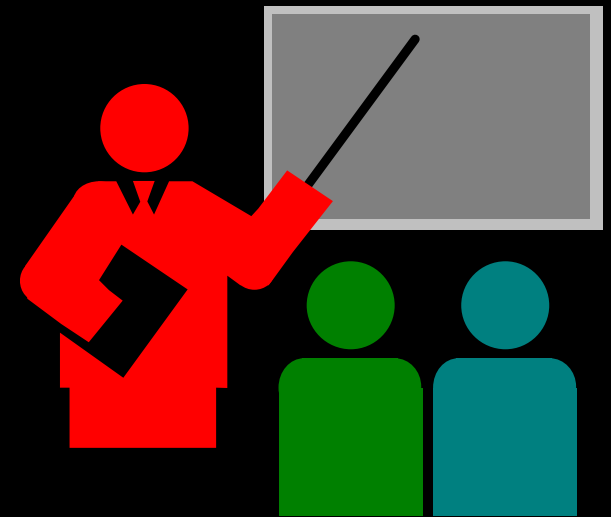
```
sprintf( msg,
    "<!DOCTYPE html> \
    <html> \
    <body> \
    <h1> LUMIAR </h1> \
    <form action=\"action_page.php\" method=\"GET\"> \
        <fieldset> \
            <legend> Configuração: </legend> \
            Valor:<br> \
            <input type=\"text\" name=\"valor\" value=\"50\"> \
            <br><br> \
            <input type=\"submit\" value=\"Submit\"> \
        </fieldset> \
    </form> \
    </body> \
    </html> \n");
```

Dicas sobre HTTP

❑ Exemplo de formatação da resposta HTTP GET

```
sprintf( buffer, "%s%s%s%d%s%s%s",  
    "HTTP/1.0 200 OK \r\n",  
    "Server: MiniWebServer 1.0\r\n",  
    "Content-Length: ",  
    size,          // tamanho exato da string msg  
    "\r\n",  
    "Content-Type: text/html\r\n",  
    "\r\n",  
    msg );  
  
status = write(newsd, buffer, strlen(buffer)+1);
```

Função transferfile()



Função transferfile()

❑ Função transferfile()

- ❖ Para transferência do arquivo (imagem, texto, ...) 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".