

DOWNLOADER

Estrutura Geral

O projeto é um **cliente FTP** que se conecta a servidores FTP, faz autenticação e faz o download de arquivos. O fluxo de trabalho principal envolve três partes principais:

1. **Conectar ao servidor FTP**
2. **Autenticar o usuário**
3. **Baixar o arquivo no modo passivo**

Estruturas e Definições Importantes

1. `URL` Struct

- **Objetivo:** Armazenar informações sobre a URL FTP.

- **Campos:**

- `host`: Nome do servidor FTP.
- `ip`: Endereço IP resolvido do servidor.
- `user`: Nome de usuário para autenticação.
- `password`: Senha para autenticação.
- `resource`: Caminho do arquivo no servidor.
- `file`: Nome do arquivo a ser baixado.

2. Constantes

- `MAX_LENGTH`: Define o tamanho máximo de buffers utilizados no programa.
- `FTP_PORT`: Porta padrão do FTP (21).
- `DEFAULT_USER` e `DEFAULT_PASSWORD`: Valores padrão para o usuário e senha caso não sejam fornecidos na URL.

Funções Principais

1. `parseURL`

- **Objetivo:** Processa a URL FTP fornecida, extraindo informações sobre o servidor, usuário, senha, caminho do arquivo e IP.

- **Como funciona:**

- Verifica se a URL começa com `ftp://`.
- Extrai as credenciais de login (se fornecidas).
- Resolve o nome do host para um endereço IP usando `resolveHostnameToIP`.
- Divide o caminho do recurso para identificar o arquivo a ser baixado.

2. connectToServer

- **Objetivo:** Estabelece uma conexão TCP com o servidor FTP.

- **Como funciona:**

- Configura a estrutura ``sockaddr_in`` com o endereço IP e a porta do servidor.
- Cria um socket e tenta conectar-se ao servidor.
- Retorna o descritor de arquivo do socket em caso de sucesso, ou ``-1`` em caso de erro.

3. ``readResponse`` e ``readFullResponse``

- **Objetivo:** Lê a resposta do servidor FTP após cada comando enviado.

- **Como funciona:**

- ``readResponse`` lê a resposta e extrai o código de resposta (ex: 220, 331, 230).
- ``readFullResponse`` lida com respostas multi-linhas, como mensagens de status detalhadas.

4. ``passiveMode``

- **Objetivo:** Envia o comando ``PASV`` ao servidor para entrar no modo passivo e obter um IP e uma porta para a transferência de dados.

- **Como funciona:**

- Envia o comando ``PASV`` e interpreta a resposta para obter o IP e a porta de dados.
- Retorna as informações necessárias para a transferência de dados.

5. ``downloadFile``

- **Objetivo:** Gerencia o processo de download do arquivo.

- **Como funciona:**

- Conecta-se ao servidor e faz a autenticação com os comandos ``USER`` e ``PASS``.
- Entra no modo passivo para obter um socket de dados.
- Solicita o arquivo com o comando ``RETR`` e recebe os dados via o socket de dados.
- Salva os dados no arquivo local.

6. ``resolveHostnameToIP``

- **Objetivo:** Resolve o hostname para um endereço IP.

- **Como funciona:**

- Utiliza a função ``gethostbyname`` para obter o endereço IP associado ao nome do host.
- Converte o endereço IP de formato binário para uma string legível com ``inet_ntoa``.

7. ``closeConnection``

- **Objetivo:** Fecha uma conexão de socket.

- **Como funciona:**

- Chama a função ``close`` para encerrar a conexão TCP.

Fluxo de Execução

1. O programa começa na função ``main``, onde recebe uma URL FTP como argumento.
2. A URL é analisada pela função ``parseURL``, que preenche a estrutura ``URL`` com os detalhes do servidor, usuário, senha, caminho do recurso e nome do arquivo.
3. A conexão ao servidor é estabelecida por meio da função ``connectToServer``.
4. A autenticação é realizada enviando os comandos ``USER`` e ``PASS``.

5. O modo passivo é ativado com o comando `PASV`, e o IP/porta de dados são extraídos.
6. O arquivo é solicitado com o comando `RETR`, e os dados são recebidos e salvos localmente.
7. Após o download, as conexões são fechadas com `closeConnection`.

Propósito Geral

O código tem como propósito fornecer uma **interface simples para baixar arquivos de servidores FTP**, realizando a autenticação, gerenciando a comunicação com o servidor e salvando os arquivos localmente, tudo isso em modo passivo (ideal para NATs e firewalls).

3LEIC06

Guilherme Coelho - up202000141@edu.fc.up.pt

Sofia Reis - up201905450@edu.fc.up.pt