MyFTP - Cliente e Servidor FTP Simples

O objetivo desse projeto é desenvolver uma aplicação cliente e servidor que implementa uma versão simplificada do protocolo **FTP**. As duas componentes principais do módulo são os executáveis do cliente `myftp` e do servidor `myftpserver`. Os comandos FTP implementados são:

1. **get** (`get <arquivo_remoto>`) - Copia o arquivo com o nome <arquivo_remoto> do diretório remoto para o diretório local.
2. **put** (`put <arquivo_local>`) - Copia o arquivo com o nome <arquivo_local> do diretório local (pasta principal do projeto) para o diretório remoto.
3. **delete** (`delete <arquivo_remoto>`) - Deleta o arquivo com o nome <arquivo_remoto> do diretório remoto.
4. **ls** (`ls`) - Lista os arquivos e subdiretórios no diretório remoto.
5. **cd** (`cd <diretorio_remoto>` ou `cd ..`) - Muda para o <diretorio_remoto> na máquina remota ou muda para o diretório pai do diretório atual.
6. **mkdir** (`mkdir <diretorio_remoto>`) - Cria um diretório com nome <diretorio_remoto> como um subdiretório no diretório de trabalho atual na máquina remota.
7. **pwd** (`pwd`) - Imprime na tela o diretório de trabalho atual da máquina remota.
8. **quit** (`quit`) - Termina a sessão FTP.

Executando a aplicação

Clone o repositório e instale os pacotes necessários. É recomendável criar antes um ambiente virtual no python.

No linux, basta executar:

```
$ python3 -m venv <nome>
$ source <nome>/bin/activate
$ pip install -r requirements.txt
```

Para iniciar o programa servidor execute:

```
$ python myftpserver.py <port_number>
```

Para iniciar programa cliente execute:

```
$ python myftp.py <server_name> <port_number>
```

Ao utilizar Windows é possível que surja o erro `erro: Microsoft Visual C++ 14.0 is required`. Nesse caso recomendamos instalar o *Visual Studio 2019 Community* (link a seguir):

> https://visualstudio.microsoft.com/pt-br/downloads

Durante a instalação selecione **Desktop development with C++**. Na parte superior, selecione **extensions** e procure por C++, com isso o download será feito e basta prosseguir com a instalação.

Especificação da aplicação

O funcionamento do cliente e servidor são explicados a seguir:

- Servidor FTP (programa `myftpserver`) - O programa servidor recebe um único parâmetro de linha de comando que é o número da porta a qual o servidor irá executar. Uma vez que o programa `myftpserver` for invocado, ele cria um *socket* TCP, associa o número da porta do servidor ao seu *socket* e passa a escutar conexões de clientes. Quando uma conexão com um cliente é estabelecida, o servidor começa a aceitar comandos e executá-los. Mensagens de erro apropriadas são enviadas ao cliente sempre que os comandos falham. Ao receber o comando `quit`, o servidor encerra a conexão com o cliente.

- Cliente FTP (programa `myftp`) - O programa do cliente FTP recebe dois parâmetros de linha de comando: o endereço da máquina que o servidor reside e o número da porta. Uma vez que o cliente começa a rodar ele exibe um *prompt* `myftp>`. A partir daí aceita e executa comandos enviando-os para o servidor e exibindo os resultados e mensagens de erro quando apropriado. O cliente deve cessar sua execução quando o usuário entrar o comando `quit`.

Implementação do protocolo

Para um detalhamento do protocolo proposto ver arquivo `PROTOCOLO.md`.

Execução do projeto

A filosofia principal foi conceber um servidor FTP simples que opera com os comandos apresentados, transferindo arquivos entre hospedeiros diferentes. As principais caracteríticas em mente foram:

1. O servidor implmentado é concorrente (*multi-threaded*), isto é suporta a conexão de múltiplos clientes simultaneamente.
2. Assumimos que o usuário sempre digita um comando com a sintaxe correta (i.e. não existe checagem da sintaxe).

As principais dificuldades surgiram em implementar os comandos de transferência de arquivos (**get & put**) onde uma atenção especial precisou ser dada, tendo em vista às restrições impostas pela transferência de dados através da rede (principalmente no tamanho do *buffer* da ordem de poucos *kilobytes*).

Também devemos citar que somente foram testados a transferência de arquivos localizados na pasta principal do projeto, como está descrito neste issue do repositório.

Algumas dificuldades foram encontradas ao testar aplicação num ambiente de diferentes sistemas operacionais, onde o servidor rodava no Windows e os clientes no Linux (o servidor simplesmente não respondia às requisições).

Como perspectiva futura, as principais melhorias que podem ser feitas seriam:

- Implementação de credenciais de acesso e permissões de usuário, o que tornaria a aplicação bem mais robusta num ambiente em que múltiplos usuários manipulam o diretório remoto.
- Implementação de checagem da sintaxe para tratar casos em que o usuário digita comandos inválidos.

Referências

A principal inspiração para esse trabalho foi o material encontrado em Programming-Project1 da Universidade da Geórgia, a qual deixamos os devidos créditos.