

# Servidor FTP

por Julio Kenji Ueda 9298281 e Ricardo Akira Tanaka 9778856

O servidor FTP foi implementado para se comportar de forma similar a como o servidor proFTPD se comporta em algumas situações, sabendo processar requisição válidas, inválidas, comandos de login, logout e listagem de arquivos. O servidor foi escrito em C, compilado e testado no GNU/Linux.

## Compilação, Remoção e Inicialização

Para compilar, basta executar:

```
$ make
```

Para remover arquivos oriundos do processo de compilação, execute:

```
$ make clean
```

Para remover arquivos oriundos do processo de compilação e execução, como a pasta e arquivos remotos, execute:

```
$ make cleanall
```

Para iniciar o servidor, execute:

```
$ ./server <endereço ip> <porta>
```

Onde:

- **<endereço ip>**: endereço ip do servidor na rede, onde pode ser descoberto por algum cliente. Para *localhost*, use 127.0.0.1. Para redes externas, utilize o ip externo.
- **<porta>**: porta de escuta do servidor.

Exemplo:

```
$ ./server 127.0.0.1 1221
```

## Comandos e respostas

O servidor é implementado para responder aos seguintes comandos:

- **QUIT**: finaliza a conexão com o servidor. O servidor responde com *"221 Goodbye"*.
- **SYST**: informa o sistema operacional do servidor e modo de transferência de dados padrão. O servidor responde com *"215 Unix Type: L8"*.

- **TYPE <modo>**: seleciona o modo de transferência de dados. Se o modo for definido como "I", então o servidor responde como *"200 Type set to I"* e a transferência de dados utiliza o modo binário; caso definido como "A", então o servidor responde como *"200 Type set to A"* e a transferência de dados utiliza o modo ASCII; em qualquer outro caso, o servidor responde como *"504 Type not implemented for <modo> parameter"*.
- **USER <nome do usuário>**: qualquer nome de usuário é válido. O servidor responde com *"331 Password required for <nome de usuário>"*.
- **PASS <senha do usuário>**: qualquer senha do usuário é válida. Se o usuário executar o comando *USER <nome do usuário>* anteriormente, então o servidor responde com *"230 <nome do usuário> logged in"* e o usuário se encontra autenticado no sistema; caso contrário o servidor responde com *"503 Login with USER first"*.

Os comandos a seguir necessitam de autenticação do usuário; caso contrário o servidor responde com *"530 Please login with USER and PASS"*.

- **DELE <nome do arquivo>**: remove um arquivo do servidor. Se o arquivo <nome do arquivo> existir, então o servidor responde com *"250 DELE command successful"* e o arquivo é removido do servidor; caso contrário responde com *"550 <nome do arquivo>: No such file or directory"*.
- **PASV**: habilita o modo passivo do servidor para uma nova conexão de dados, onde esta nova conexão deve ser solicitado a partir do cliente. O servidor responde com *"227 Entering passive mode (<ip1>, <ip2>, <ip3>, <ip4>, <p1>, <p2>)"*, informando o endereço da nova conexão de escuta para transferência de dados, onde <ip1>.<ip2>.<ip3>.<ip4> é o <endereço ip> informado no momento da execução do servidor e <p1> = <porta de escuta para dados>/256 e <p2> = <porta de escuta para dados>%256.

Os comandos a seguir necessitam de autenticação do usuário e também da conexão de dados; caso contrário o servidor responde com *"425 Unable to build data connection: Connection refused"*.

- **LIST**: lista o diretório remoto. O servidor responde com *"150 Opening ASCII mode data connection for file list"* antes da transferência de dados e responde com *"226 Transfer complete"* ao término da transferência, fechando a conexão de dados logo em seguida.
- **RETR <nome do arquivo>**: recupera um arquivo remoto. Se o arquivo <nome do arquivo> existe, então o servidor responde com *"150 Opening BINARY mode data connection for <nome do arquivo>"* antes da transferência e responde com *"226 Transfer complete"* ao término da transferência do arquivo, fechando a conexão de dados logo em seguida; caso contrário responde com *"550 <nome do arquivo>: No such file or directory"*.
- **STOR <nome do arquivo>**: envia um arquivo ao servidor. Se o arquivo <nome do arquivo> existe, então o servidor responde com *"150 Opening BINARY mode data connection for <nome do arquivo>"* antes da transferência e responde com *"226 Transfer complete"* ao término da transferência do arquivo, fechando a conexão de

dados logo em seguida; caso contrário responde com “*451 Requested action aborted. Local error in processing*”.

## Execução e teste do servidor

É possível executar o servidor utilizando *telnet*, mas é recomendado utilizar o cliente FTP do shell, pois o *telnet* não permite transferência de arquivos binários. Para iniciar a conexão utilizando o cliente FTP do shell, digite:

```
$ ftp -p <endereço ip> <porta>
```

Onde:

- **p**: flag que indica o modo passivo para transferência de dados.
- **<endereço ip>**: endereço ip do servidor na rede, informado no momento da execução do servidor.
- **<porta>**: porta de escuta do servidor.

Exemplo:

```
$ ftp -p 127.0.0.1 1221
```

O cliente FTP do shell executa a seguinte pipeline de comandos ao iniciar a conexão com o servidor:

1. USER <nome do usuário>
2. PASS <senha do usuário>
3. SYST

O cliente deve informar seu nome e sua senha, e logo após encontra-se autenticado no servidor com o modo de transferência de dados padrão definido como binário.

```
Connected to 127.0.0.1.
200 MAC0352 EP1 Server
Name (127.0.0.1:root): username
331 Password required for username
Password:
230 User username logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

O servidor está pronto para interagir com o cliente. O comando PASV é automaticamente executado para comandos que exigem conexão de dados como STOR, RETR e LIST.

Agora é possível:

- Enviar um arquivo que existe:

```
ftp> put photo.jpg
local: photo.jpg remote: photo.jpg
227 Entering Passive Mode (127,0,0,1,177,27)
150 Opening BINARY mode data connection for photo.jpg
```

```
226 Transfer complete
541189 bytes sent in 0.00 secs (408.6446 MB/s)
ftp>
```

- Tentar enviar um arquivo que não existe:

```
ftp> put null.jpg
local: null.jpg remote: null.jpg
local: null.jpg: No such file or directory
ftp>
```

- Recuperar um arquivo que existe:

```
ftp> get photo.jpg
local: photo.jpg remote: photo.jpg
227 Entering Passive Mode (127,0,0,1,177,27)
150 Opening BINARY mode data connection for photo.jpg
226 Transfer complete
541189 bytes received in 0.00 secs (155.5509 MB/s)
ftp>
```

- Tentar recuperar um arquivo que não existe:

```
ftp> get null.jpg
local: null.jpg remote: null.jpg
227 Entering Passive Mode (127,0,0,1,177,27)
550 null.jpg: No such file or directory
ftp>
```

- Listar arquivos:

```
ftp> ls
227 Entering Passive Mode (127,0,0,1,177,27)
150 Opening ASCII mode data connection for file list
total 540
drwxr-xr-x 2 root root 4096 set  2 14:03 .
drwxr-xr-x 3 root root 4096 set  2 14:03 ..
-rw-r--r-- 1 root root 541189 set  2 13:54 photo.jpg
226 Transfer complete
ftp>
```

- Apagar arquivo que existe:

```
ftp> del photo.jpg
250 DELE command successful
ftp>
```

- Tentar apagar arquivo que não existe:

```
ftp> del null.jpg  
550 null.jpg: No such file or directory  
ftp>
```

- Finalizar sessão:

```
ftp> exit  
221 Goodbye.  
ftp>
```

## Referências

RFC 959 - File Transfer Protocol. Disponível em <<https://tools.ietf.org/html/rfc959>>

D. J. Bernstein. FTP: File Transfer Protocol. Disponível em <<http://cr.yp.to/ftp.html>>