

Protocolo Caixote

Projecto RC 2016/2016 - Grupo 15
75785 João Miguel Campos
78013 Bruno Henriques
78040 Sofia Reis

Introdução

O projecto Caixote sincroniza ficheiros entre cliente e servidor, de forma parecida ao *Dropbox* ou *rsync*. Foi implementado em Python, utilizando as APIs de ficheiros, threads e sockets.

Funcionamento da Aplicação

1. O Servidor abre na *port* indicada e fica à espera de ligações de clientes.
2. Assim que um Caixote(cliente) liga-se, o Servidor cria uma thread dedicada à ligação.
3. Estabelecido o socket de ligação, os Caixote-Servidor sincronizam os ficheiros.

A sincronização na aplicação resume-se em 3 partes:

1. Autenticação do Cliente (**LOGIN > LOGGED**)
2. Identificação dos ficheiros a sincronizar (**DIFF > DIFFS**)
3. Partilha de ficheiros. Caixote envia ficheiros ao Servidor > Caixote pede ficheiros ao Servidor. Servidor envia ficheiros ao Caixote. (**PUT, GET/GETN > FILE**)

Protocolo

O protocolo de comunicação Caixote-Servidor baseia-se em header lines e respectivo conteúdo opcional. Uma *header line* é uma qualquer sequência de caracteres terminada em \n. Nessa linha constam, separados por :, o método e argumentos da mensagem.

O método de uma *header line* é a palavra identificadora da informação que se segue. Exemplo:

LOGIN:utilizadorxpto:pastafixe\n > [LOGIN, utilizadorxpto, pastafixe]

Métodos (header lines) do protocolo reconhecidos pela Aplicação:

Método	Argumentos	Descrição
LOGIN	Username, Foldername	Método de autenticação do utilizador
DIFF	Number of lines	Método que informa ao servidor os ficheiros do utilizador
PUT/FILE	Filepath, Length, mtime	Envia o conteúdo dum ficheiro PUT: Caixote, File: Servidor
GET/GETN	Filepath/Number of lines	Requisita o conteúdo dum/vários ficheiros
LOGGED	[IN/OUT]	Resposta ao pedido de autenticação. IN=OK, OUT=Falhou
DIFFS	Number of lines	Resposta ao método DIFF. Informa os ficheiros diferentes entre o Caixote e Servidor (CLIOLD, SRVOLD)

