

Acompanha este enunciado a implementação em C# de um servidor *single-threaded* que realiza *tracking* de ficheiros disponibilizados por máquinas que participam num sistema de partilha de ficheiros entre pares (e.g., *BitTorrent*). O servidor mantém informação relativa aos ficheiros (i.e., nome) existentes nos participantes e a sua localização (i.e., endereço IP e porto a serem usados para obter o ficheiro).

A comunicação entre o servidor e os clientes (i.e. participantes no sistema de partilha) é realizada através de protocolo proprietário baseado em pares pedido/resposta e sustentado no protocolo de transporte TCP. As ligações TCP são mantidas enquanto o cliente não solicita a sua terminação, existindo a possibilidade de vários pares pedido resposta serem trocados usando a mesma ligação. O protocolo proprietário, documentado na implementação fornecida, oferece suporte para as seguintes operações: adição e remoção de localização de ficheiro; obtenção de lista de ficheiros, e; obtenção de lista de localizações de um dado ficheiro.

Os principais elementos da implementação fornecida são as classes *Listener* e *Handler*, cujas instâncias participam no serviço de pedidos. A classe *Store* mantém em memória volátil (por simplificação) a informação de *tracking*. A definição destas classes está acompanhada da respectiva documentação.

O servidor mantém registo das acções realizadas (classe *Logger*), que pode ser apresentado na consola ou em ficheiro. Na implementação fornecida, cada execução do servidor cria um *log* na consola.

1. Inspirando-se na estrutura do servidor *single-threaded* fornecido, e tirando partido do *Asynchronous Programming Model* da *Framework .NET*, implemente uma versão *multi-threaded* do servidor, com as seguintes características:
 - Atendimento simultâneo de pedidos recorrendo à interface assíncrona da API de *sockets*.
 - Desistência por *timeout* na recepção de pedidos (uma vez estabelecida a ligação TCP).
 - Funcionalidade de registo (em *Logger.cs*) suportada por uma *thread* de baixa prioridade (*logger thread*), criada para o efeito. As mensagens com os relatórios devem ser passadas das *threads* que servem pedidos (produtoras) para a *logger thread* usando um mecanismo de comunicação que minimize o tempo de bloqueio das *threads* produtoras. A funcionalidade de registo deve ter o mínimo de influência no tempo de serviço, admitindo-se inclusivamente a possibilidade de ignorar relatórios.
2. Implemente uma versão minimalista de cliente com interface gráfica e que realize as operações suportadas pelo servidor, apresentando os respectivos resultados.
3. Implemente em C#, e usando a *framework* TPL, um método utilitário para pesquisa de ficheiros que contenham uma sequência arbitrária de caracteres. O método recebe como argumentos: o caminho absoluto da pasta raiz da pesquisa, a extensão dos ficheiros a considerar, e a sequência de caracteres a pesquisar nos ficheiros considerados. A pesquisa é realizada assincronamente tirando partido da eventual existência de vários processadores e é passível de ser cancelada. O resultado da pesquisa contém a lista dos nomes dos ficheiros (i.e. caminhos absolutos) que cumprem os critérios de pesquisa, o número de ficheiros encontrados com a extensão especificada e o número total de ficheiros encontrados.
4. Implemente uma versão minimalista de aplicação para a realização de pesquisas no sistema de ficheiros. A aplicação contém interface gráfica para recolha dos parâmetros de pesquisa e para apresentação dos resultados. Para melhorar a experiência de utilização, a apresentação dos resultados é realizada de forma incremental enquanto a pesquisa decorre (i.e. informação de progresso). A interface gráfica inclui botão para cancelamento da pesquisa em curso.

Nota de implementação: Faça uso da TPL e do método utilitário implementado na alínea anterior, modificando-o se necessário.

5. Implemente uma nova versão do servidor do exercício 1, tirando o máximo partido dos métodos assíncronos disponíveis no *Framework* .NET 4.5. Use o cliente implementado no exercício 2 para testar esta nova versão do servidor.

Data limite: 16 de Julho de 2014

ISEL, 18 de Junho de 2014