Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores
Programação Concorrente, Verão de 2016/2017

Terceira série de exercícios

Este enunciado é acompanhado pela implementação em C# de um servidor *single-threaded* que realiza o *tracking* de ficheiros disponibilizados por máquinas que participam num sistema de partilha de ficheiros entre pares (e.g., `BitTorrent`). O servidor deve manter uma tabela com a informação relativa aos ficheiros conhecidos; para cada ficheiro deve ser memorizado o nome e a lista dos *network endpoints* (e.g., endereço IP e porto) onde o mesmo pode ser obtido.

A comunicação entre o servidor e os participantes no sistema de partilha de ficheiros 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 respectivo cliente não solicitar a sua terminação, de modo a permitir usar a mesma ligação para trocar vários pares pedido resposta. 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. Por simplificação, a classe `Store` mantém em memória a informação de *tracking* dos ficheiros. 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 operação executada pelo servidor apresenta um registo na consola.

1. Inspirando-se na estrutura do servidor *single-threaded* fornecido, implemente uma versão *multi-threaded* do servidor, com os seguintes requisitos:

   - Utilização das variantes assíncronas das operações da API de *sockets*, de acordo com o *Asynchronous Programming Model* da plataforma .NET, incluindo no método que aceita pedidos dos clientes (`TcpListener.AcceptTcpClient`).
   - Limitação, por concepção, do número máximo de pedidos que o servidor pode processar simultaneamente.
   - Limitação, por concepção, do número máximo de chamadas recursivas ao método de *callback* por cada IOCP *thread*. As chamadas recursivas podem ocorrer quando a operação invocada assincronamente é concluída pelo método `BeginXxx`. Nota: a invocação recursiva do método de *callback* é reportada através da propriedade `IAsyncResult.CompletedSynchronously`.
   - 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 outra versão do servidor do exercício 1, baseado exclusivamente nas operações assíncronas da API de *sockets* ao estilo TAP (*Task-based Asynchronous Pattern*). Tire partido dos métodos assíncronos disponíveis a partir do C# 5.0.

3. Usando a TPL (*Task Parallel Library*) e, se achar conveniente, os métodos assíncronos do C#, implemente um método utilitário para pesquisa de ficheiros no disco que determine quais os ficheiros que contêm 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
- 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. Utilize a TPL e do método utilitário da alínea anterior, modificando-o se necessário.

Data limite: 9 de Julho de 2017

ISEL, 2 de Junho de 2017