



ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

Relatório 3ª Serie de SO

Docente: João Pedro Patriarca

Semestre Verão – 16/17

Turma LI42D

Alunos:

Rui Franco - 36210

Ana Baptista - 41487

Miguel Arroja - 42161

1)

As desvantagens da implementação da aplicação servidora em anexo são:

- Poderão ser esgotadas todas as threads disponíveis para o processamento dos pedidos, aumentando imenso o tempo de resposta dos mesmos.
- É aumentado o número de threads associados à CP descontrolada, em que não só aumenta o tempo de resposta, mas o tamanho de memória a ser utilizada (Como por exemplo a partir de stacks das threads). As comutações das threads também é um problema inerente nesta implementação.
- A diretoria raiz de pesquisa é determinada por parâmetro em linha de comando.
- Os acessos aos dispositivos de I/O envolvidos no processamento de uma pesquisa são realizados exclusivamente com acessos síncronos aos respetivos dispositivos.

2) Alterações realizadas na implementação:

Connection.h: adicionamos um overlapped no início desta estrutura. Adicionamos também a declaração de um novo método: ConnectionGetLineAsync

Connection.cpp : definimos o método ConnectionGetLineAsync para substituir a utilização do método ConnectionGetLine da implementação original, para não alterarmos a função origem pois a mesma pode estar a ser utilizada por outras funções/classes. Esta nova função tem o mesmo comportamento que a anterior mas é aplicado ao buffer que se encontra na connection em vez de ler através da função CGetChar.

Jpg_SearchService.cpp: Aqui foram feitas a maioria das alterações.

Main : Iniciamos um CompletionPort e uma thread para fazer a gestão de pedidos. Iniciamos também uma connection por cada socket e um buffer para alocação de dados. Chamamos a função WSARecv para receber os dados mas de forma assíncrona.

ProcessConnection : Modificamos o comportamento desta função. Num ciclo infinito e através da função GetQueuedCompletionStatus esperamos que a thread cliente seja acordada quando houver trabalho. De seguida o trabalho é processado mais uma vez através da função WSARec.

Handler.ProcessRequest: Esta função passa a utilizar a função ConnectionGetLineAsync em vez de ConnectionGetLine