

Objetivos: prática de utilização de IOCP no suporte à execução de operações assíncronas. Para cada questão do enunciado deve realizar programas que demonstrem o correto funcionamento. A entrega deverá incluir um pequeno relatório que explique as decisões de implementação tomadas.

- 1) Realize uma biblioteca (DLL) de suporte a operações assíncronas de cópia de ficheiros suportadas em *I/O Completion Port*. A biblioteca deve exportar as seguintes operações, passíveis de serem invocadas de forma concorrente (*thread-safe*).

BOOL AsyncInit();	Constrói a infraestrutura necessária ao suporte de operações assíncronas de cópia de ficheiros. Na eventualidade de invocação múltipla apenas a primeira invocação deve ter efeito.
<pre>typedef VOID(*AsyncCallback)(LPVOID userCtx, DWORD status, UINT64 transferredBytes); BOOL CopyFileAsync(PCSTR srcFile, PCSTR dstFile, AsyncCallback cb, LPVOID userCtx);</pre>	<p>A função CopyFileAsync inicia a cópia assíncrona do ficheiro de nome srcFile para o ficheiro de nome dstFile. Na conclusão da operação, ou na ocorrência de algum erro durante a sua execução, a função de <i>callback</i> (cb) é invocada com o argumento userCtx, o <i>status</i> da operação(status) e o total de <i>bytes</i> transferidos (transferredBytes). Na ausência de erro o valor de <i>status</i> é zero, caso contrário corresponde ao erro dado pela chamada à função GetLastError().</p> <p>A função retorna TRUE se a operação foi iniciada com sucesso, FALSE caso contrário.</p>
VOID AsyncTerminate();	<p>Termina e destrói todos os recursos associados à infraestrutura de suporte a operações assíncronas. Na eventualidade de invocação múltipla, apenas a primeira invocação deve ter efeito.</p> <p>Valorizam-se soluções que garantem a execução de todas as operações de cópia pendentes</p>

- 2) Com base na infra estrutura anterior, realize a função **DWORD CopyFolder(PCSTR origFolder, PCSTR dstFolder)**, que copia os ficheiros presentes na pasta de nome **origFolder**(ignorando sub-pastas), para a pasta de nome **dstFolder** (que se supõe já criada).
- As cópias de ficheiros serão realizadas em paralelo através da função **CopyFileAsync** desenvolvida no ponto anterior.
- A função **CopyFolder** é síncrona, retornando apenas quando todos os ficheiros forem processados.
- O retorno indica o número de ficheiros em que a cópia não foi executada com sucesso.

Data limite de entrega: 13 de Junho de 2018

João Patriarca,
João Trindade,
Jorge Martins