

Objectivo: 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 de suporte a operações assíncronas sobre ficheiros de texto suportadas em *I/O Completion Port*. A única operação a suportar sobre ficheiros de texto corresponde à realização do histograma de todos os caracteres do alfabeto presentes no respectivo ficheiro. O histograma corresponde ao somatório das ocorrências de cada carácter do alfabeto no ficheiro. 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 (histograma) sobre ficheiros de texto. Na eventualidade de invocação múltipla apenas a primeira invocação deve ter efeito.
<pre>typedef VOID(*AsyncCallback)(LPVOID userCtx, DWORD status, UINT32 * histogram); BOOL HistogramFileAsync(PCSTR file, AsyncCallback cb, LPVOID userCtx);</pre>	<p>A função HistogramFileAsync inicia o processamento assíncrono do histograma do ficheiro de nome file. 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 histograma calculado (histogram).</p> <p>O histograma é constituído por uma sequência de inteiros igual em ordem e número ao alfabeto da língua inglesa. Cada entrada da sequência representa o somatório de ocorrências do respectivo carácter.</p> <p>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 histogramas pendentes.</p>

- 2) [Opcional] Com base na infra-estrutura anterior, realize a função **BOOL PrintHistogramFolder(PCSTR folder)**, que escreve no *stdout* o histograma para cada ficheiro presente na pasta de nome **folder**. A anteceder a escrita de um histograma deve ser escrito o nome do ficheiro correspondente.
- A produção do histograma é realizado através da função **HistogramFileAsync** desenvolvida no ponto anterior.
- A função **PrintHistogramFolder** é síncrona, retornando apenas quando todos os histogramas forem escritos no *stdout* (retornando TRUE), ou for encontrado algum erro (retornando FALSE).

Data limite de entrega: 3 de Janeiro de 2019

João Patriarca,

ISEL, 6 de Dezembro de 2018