**I**nstituto **S**uperior de **E**ngenharia de **L**isboa

Licenciatura em Engenharia Informática e de Computadores

Semestre de Inverno - 2015/2016



Série de Exercícios 3  
Sistemas Operativos

**Trabalho elaborado por:**

**Grupo 12**

**José Pica Nº 35455**

**Pedro Duarte Nº 36832**

**João Borges Nº 36848**

Grupo I



A adição da operação *WriteAsync* á infraestrutura foi baseada na operação *ReadAsync*. Para realizar esta operação é criada uma estrutura *IOBaseOper*, que representa a operação assíncrona, e é necessário também criar uma função para invocar a função de *callback* recebida para informar da conclusão da operação. A escrita no *device* desejado é realizado através da função *WriteFile*. Como a função *WriteFile* pode retornar antes de a escrita ter sido concluída, é preciso verificar se existiu erro *ERROR\_IO\_PENDING,* para a operação retornar o resultado de conclusão correto.

Ao realizar operação *CopyFile2Async*, sendo para ser usado as operações *ReadAsync* e *WriteAsync.* Foi necessário criar uma estrutura composta pela operação base, para operação de leitura do ficheiro origem, um *buffer* com o conteúdo do ficheiro e também associar um *device* para o ficheiro destino do conteúdo. A operação precisa ter a representação do acesso aos dois ficheiros, para isso é recorrido o uso das funções *OpenAsync*, para o ficheiro de leitura, e *CreateAsync* para o ficheiro onde é feita a escrita. Esta operação começa por ser efetuada a leitura do ficheiro origem, pela chamada a *ReadAsync*. Logo, só sabemos que esta operação foi concluída pela sua chamada á função de *callback* passada por parâmetro.

Visto isto, foi implementada uma função para passar como *callback* a *ReadAsync*, onde só ai é chamada a operação *WriteAsync*. É passado como contexto a este *callback,* a estrutura definida por CopyFile2Async, com o ficheiro associado para a escrita, o conteúdo do *buffer*, o número de bytes lidos e o *callback* recebido pela operação *CopyFile2Async* para sinalizar a finalização desta operação composta.