Relatório SO

Série de Exercícios 2

Eng.º João Pedro Patriarca
Semestre 2015/2016 Verão

Alunos:

Pedro Tavares	41490
Pedro David	41530
Filipe Coelho	41873

1

a)

Foi adicionado um enum State {Running, Ready, Blocked}; que representão os 3 estados possíveis de uma UThread. Quando uma Uthread é criada o estado é iniciado a Ready e volta para este estado no UtActivate() e no UtYield() só a RunningThread (antes da troca), O estado de Running só dado á única Uthread a correr (RunningThread), por ultimo o estado Blocked é atribuído quando uma Uthread deixa de competir pelo CPU por exemplo quando esta faz Join a outra,

b)

Foi acrescentado `LIST_ENTRY AliveLink; ás Uthreads para os ligar na` LIST_ENTRY AliveQueue; que é um campo global da Uthread.c assim como a RunningThread, este guarda todas as threads que ainda estão Alive.

c)

Para implementar o UtTerminate() tiveram que ser acrescentados dois UtExits complementares, `UtExitAux(HANDLE uThread) e UtExitAux2(HANDLE uThread)`, a primeira termina a Uthread passada como parâmetro e continua a execução da próxima Ready thread, isto é utilizado no caso da thread passar do estado Blocked para Ready onde é logo terminada, o segundo apenas termina a thread passada e continua a execução da RunningThread, foi também acrescentado um BOOL ToTerminate; que é iniciado a FALSE e se for chamado o Uterminate e a UThread passada não for a RunningThread este campo é passado a TRUE, quando forem trocadas as thread e este campo for TRUE a thread é terminada.

d)

Foram acrescentados os campos DWORD NWaiting; DWORD NRelease; ás Uthreads estes campos servem para saber quantas threads estão á espera que a própria termine (NRelease), o outro campo serve para saber quantas threads a própria está á espera, quando este valor chega a 0 ela é ativada utilizando o UtActivate() conseguindo desta maneira só uma transição de Blocking->Running.

3

Á estrutura fornecida foi acresentada uma

```
typedef struct {
	TCHAR			*path;
	TCHAR			*fileMatch;
	DirectoryProcessor	dp;
	FileProcessor		fp;
	LPVOID			ctx;
	HANDLE			semaphore;
	BOOL			recursive;
} Parameters, *PParameters;
```

Esta struct é o contexto passado quando é criada uma nova thread, estas são criadas quando é encontrada uma nova diretoria e o nº de threads criadas não excede maxThreads caso contrario o processo continua recursivamente.

Para o programa não se prender num estado global teve que ser adicionado um semáforo (`HANDLE semaphore;`) que é criado com um nome para poder ser aberto (caso já exista) por outras instâncias de FindFilesPar.