

I Parte
Biblioteca Uthread

Para cada questão onde não for exigido explicitamente, apresente pelo menos um programa de teste que suporte a correção da solução proposta.

- 1) Modifique a biblioteca `Uthread` para suportar as seguintes funcionalidades:
 - a) Realize a função `BOOL UtAlive(HANDLE thread)` que retorna *true* se o *handle* passado como argumento corresponder ao de uma *thread* em actividade. Entende-se por *thread* em actividade qualquer *thread* que tenha sido criada e ainda não tenha terminado (não tenha invocado a função `UtExit`), independentemente do seu estado. Sugestão: mantenha uma lista de todas as *threads* em actividade.
 - b) Acrescente um campo ao descritor das *uthreads* para indicar o seu estado corrente. Os estados podem ser: `Running`, `Ready` e `Blocked`. Adicione à API a função `INT UtThreadState(HANDLE thread)` que retorna o estado da *thread* passada por parâmetro. Faça as alterações necessárias para manter o estado actualizado.
 - c) Acrescente a função `VOID UtSwitchTo(HANDLE threadToRun)`, que provoca uma comutação imediata de contexto para a *thread* `threadToRun`, se esta se encontrar no estado *ready*. Se não for esse o caso a função não tem nenhum efeito.
 - d) Acrescente a função `BOOL UtMultJoin(HANDLE handle[], int size)` que espera pela terminação de todas as *threads* passados no array *handle*. No caso de algum *handle* não corresponder a *thread* *alive* (de acordo com a função `UtAlive` proposta na alínea a)), ou corresponder à *thread* invocante, a função retorna de imediato com o valor `FALSE`. Caso contrário, espera (usando uma e uma só transição de *running*->*blocked*) que todas as *threads* terminem, retornando nesse caso o valor `TRUE`.
 - e) Escreva programas para determinar o tempo de comutação de *threads* na biblioteca `UThread`. Para a medição de tempos, utilize a função da Windows API `GetTickCount`.

II Parte
Modelo Computacional do Windows

<< A publicar brevemente >>