***3. Acrescente a função BOOL UtMultJoin(HANDLE handle[], int size) que espera passivamente pela terminação de todas as threads passados no array handle. Se o handle 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.***

A nossa implementação para esta solução consiste no seguinte pressuposto: é adicionado um campo ao descritor de todas as UThreads que funciona como um contador de UThreads que têm a responsabilidade de acordar a sua “main thread “de forma a esta última poder sair do estado **BLOCKED**.

O método UtMultJoin é responsável por percorrer o array de HANDLES recebido como parâmetro, criar para cada um deles uma estrutura capaz de armazenar a informação da Running Thread (neste caso a thread que futuramente vai ser acordada) e o campo de ligação na sua lista Joiners onde é inserida no fim a Running Thread.

De seguida incrementa-se contador da Running Thread por cada HANDLE válido presente no array recebido. Desta forma, quando uma das Worker Threads terminar (através da chamada ao método UtExit) é decrementado o contador de cada uma das threads presentes na sua lista Joiners e é verificado se esse contador chegou ao valor zero. Se sim é porque está no momento de acordar a thread que ficou bloqueada. Caso contrário, existe uma espera passiva para que esse contador chegue ao valor zero.