

Sommario

SY10InitializeCriticalSection 3
IMPLEMENTAZIONE 3

SY10InitializeCriticalSection

SY = Funzione di sistema

La funzione SY10InitializeCriticalSection inizializza il pool informazioni di una sezione critica.

IMPLEMENTAZIONE

SY10InitializeCriticalSection(pool)

Parametri:

IN/OUT	TIPO VARIABILE	NOME DI ESEMPIO	DIM
IN	ARRGBL	pool	B Pool informazioni sezione critica

Descrizione:

Ogni sezione critica basa il proprio funzionamento interno su una struttura dati proprietaria chiamata *pool* informazioni. Ogni sezione critica avrà un proprio *pool* che dovrà essere sempre inizializzato prima di poter essere utilizzato. Il *pool* è formato da un array di bytes dove il numero di elementi dipende dal numero di unità task che lo utilizzeranno più un header fisso di due byte.

Se per esempio una determinata sezione critica viene utilizzata in solo due unità task basterà creare un pool di 4 byte stando attenti che una unità dovrà utilizzare l'ID 1 mentre la seconda l'ID 2.

Gli ID sono progressivi partendo da 1 ed identificano la posizione delle informazioni nel pool per cui passare un ID con indice non disponibile nel pool significa invalidare la chiamata e il relativo funzionamento.

NB: Per evitare problemi si consiglia di dimensionare i pool informazioni con dimensioni sufficienti ad indirizzare tutte le unità task.

Esempio

Nell'esempio viene inizializzato un pool informazioni utilizzato poi per una sezione critica.

unità configurazione:

```

; critical section consts
CONST
  CS_POOL_HEADER      2      ; fixed header in critical section pool info
  CS_POOL_SIZE        10     ; critical section pool info size
ARRGBL
  LockPool            B      CS_POOL_HEADER + CS_POOL_SIZE

```

unità inizializzazione:

```

;===
; Initialize system
;
GLOBAL
  Initialized          F      OUT
; main entry point
BEGIN
  CALL TASK_INIT
  WHILE TRUE
    CALL TASK_EXECUTE
    WAIT A_LOOP
ENDWHILE
END
;===
; Task initialization
;
SUB TASK_INIT
  SY10InitializeCriticalSection(LockPool)
  Initialized = TRUE
ENDSUB
;===
; Task execution
;
SUB TASK_EXECUTE
  SUSPEND
ENDSUB

```

Note

- La funzione deve essere eseguita una sola volta e nella fase di inizializzazione dell'applicativo.

- Le funzioni [SY10EnterCriticalSection](#) e [SY10LeaveCriticalSection](#) non devono essere chiamate prima di aver inizializzato la sezione critica con [SY10InitializeCriticalSection](#).

Documento generato automaticamente da **Qem Wiki** - <http://wiki.qem.it/>

Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.