



RtCreateSemaphore(...)

Funkce `RtCreateSemaphore` vytvoří synchronizační objekt typu semafor.

Prototyp

HANDLE

```
RtCreateSemaphore(  
    LPSECURITY_ATTRIBUTES    lpEventAttributes,  
    LONG                     lInitialCount,  
    LONG                     lMaximumCount  
    LPCTSTR                 lpName  
);
```



RtCreateSemaphore(...)

Parametry

lpEventAttributes v RTSS ignorováno.

lInitialCount určuje počáteční stav čítače objektu semaforu. Tato hodnota musí být v rozmezí $\langle 0, lMaximumCount \rangle$. Stav objektu je signalizován, když hodnota jeho čítače je větší než 0; objekt je v nesignálním stavu, když je hodnota čítače rovna 0. Stav čítače je snížen pokaždé, když je uvolněno vlákno z funkce čekání (`RtWaitForSingleObject` nebo `RtWaitForMultipleObjects`). Stav čítače je zvětšen o definovanou hodnotu při zavolání funkce `RtReleaseSemaphore`.

lMaximumCount maximální hodnota, kterou může čítač objektu dosáhnout. Tato hodnota musí být větší než 0.

lpName ukazatel na řetězec označující název objektu. Řetězec může obsahovat libovolný znak kromě znaku backslash '\'. Maximální délka řetězce je `RTX_MAX_PATH`. Pokud je tento parametr `NULL`, pak je objekt vytvořen bez názvu. Názvy objektů jsou case-sensitive.



RtCreateSemaphore(...)

Návratová hodnota

Pokud funkce uspěje, návratovou hodnotou je handle na objekt typu událost. Pokud funkce selže, pak vrátí *NULL*; voláním funkce `GetLastError` lze získat další informace o chybě.

Poznámky

Pokud funkce `GetLastError` vrátí *ERROR_ALREADY_EXISTS*, pak objekt tohoto názvu (parametr *lpName*) již existuje a parametry *lInitialCount* a *lMaximumCount* jsou ignorovány, protože již byly nastaveny procesem, který objekt vytvořil. Již existující objekt typu událost lze otevřít funkcí `RtOpenSemaphore`.

Pokud parametr *lpName* obsahuje jméno již existujícího objektu, který však není typu semafor, pak funkce selže a `GetLastError` vrátí *ERROR_INVALID_HANDLE*; objekty, které lze pojmenovat (tj. událost, mutex, semafor a sdílená paměť) totiž sdílejí v RTSS stejný prostor jmen.



RtCreateSemaphore(...)

Poznámky

Libovolné vlákno v daném procesu může pomocí funkcí čekání

`RtWaitForSingleObject` nebo `RtWaitForMultipleObjects` čekat na signalizaci objektu typu semafor. Objekt je v signálním stavu, když je hodnota jeho čítače větší než nula. V tom případě případná funkce čekání skončí, čímž dojde k uvolnění příslušného vlákna a hodnota čítače objektu typu semafor se automaticky sníží o 1.

Pokud na semafor čeká více jak jedno vlákno ve funkci čekání a vnitřní čítač objektu umožňuje uvolnění dalších vláken (tj. hodnota čítače je větší než nula), uvolňují se další vlákna v pořadí určené jejich prioritou - vlákno s vyšší prioritou před vláknem s nižší prioritou. V případě shodné priority se vlákna uvolňují v pořadí určené délkou jejich čekání - to které čekalo déle je uvolněno dříve.

Procesy mohou mezi sebou sdílet objekty typu semafor. Proces otevře objekt typu semafor, který před tím vytvořil jiný proces, pomocí funkce `RtOpenSemaphore`.

Handle na objekt typu semafor je možné uzavřít pomocí funkce `RtCloseHandle`. Operační systém uzavře handle na objekt typu semafor automaticky při ukončení procesu, který handle vytvořil. Objekt typu semafor je vymazán, když jsou uzavřeny **všechny** handle, které se na něho odkazovaly.



RtOpenSemaphore(...)

Funkce `RtOpenSemaphore` vrátí handle na existující pojmenovaný objekt typu semafor.

Prototyp

HANDLE

`RtOpenSemaphore (`

 DWORD *DesiredAccess*,
 BOOL *bInheritHandle*,
 LPCTSTR *lpName*

`);`

Parametry

DesiredAccess v RTSS ignorováno.

bInheritHandle v RTSS ignorováno.

lpName ukazatel na řetězec obsahující název existujícího
pojmenovaného objektu typu semafor, který chceme otevřít.
Názvy objektů jsou case-sensitive.



RtOpenSemaphore(...)

Návratová hodnota

Pokud funkce uspěje, návratovou hodnotou je handle na objekt typu událost. Pokud funkce selže, pak vrátí *NULL*; voláním funkce `GetLastError` lze získat další informace o chybě.

Poznámky

Funkce uspěje pouze tehdy, pokud některý proces objekt typu semafor daného názvu již vytvořil pomocí funkce `RtCreateSemaphore`.

Handle na objekt typu semafor je možné uzavřít pomocí funkce `RtCloseHandle`. Operační systém uzavře handle na objekt typu událost automaticky při ukončení procesu, který handle vytvořil. Objekt typu událost je vymazán, když jsou uzavřeny *všechny* handlery, které se na něho odkazovaly.



RtReleaseSemaphore(...)

Funkce `RtReleaseSemaphore` zvětší stav čítače objektu typu semafor o definovanou hodnotu.

Prototyp

BOOL

```
RtReleaseSemaphore(  
    HANDLE    hSemaphore ,  
    LONG      lReleaseCount ,  
    LPLONG    lpPreviousCount  
);
```



RtReleaseSemaphore(...)

Parametry

hSemaphore

handle na objekty typu semafor, který byl získán buďto voláním funkce `RtCreateSemaphore` nebo `RtOpenSemaphore`.

lReleaseCount

určuje o kolik má být zvětšena hodnota vnitřního čítače objektu. Hodnota **musí** být větší než 0. Pokud by zvětšení stavu čítače objektu o tuto hodnotu způsobilo překročení max. povolené hodnoty čítače (definované parametrem *lMaximumCount* při vytváření objektu funkcí `RtCreateSemaphore`), pak hodnota čítače **nebude** změněna a funkce vrátí **FALSE**.

lpPreviousCount

ukazatel na místo (32 bitů), kam bude uložena hodnota vnitřního čítače objektu před jeho zvětšením o hodnotu *lReleaseCount*. Pokud je tento parametr **NULL**, hodnota čítače není požadována a tudíž ani vrácena.



RtReleaseSemaphore(...)

Návratová hodnota

Pokud funkce uspěje, návratová hodnota je *TRUE*.

Pokud funkce selže, návratová hodnota je *FALSE*; voláním funkce `GetLastError` lze získat další informace o chybě.

Poznámky

Objekt typu semafor se nachází v signálním stavu, pokud je hodnota jeho čítače *větší* než nula. Objekt typu semafor se nachází v nesignálním stavu, pokud je hodnota jeho čítače rovna nule.



RtReleaseSemaphore(...)

```
while (i<limit) {
    //wait forever for the Semaphore to become signaled
    dw = RtWaitForSingleObject(
        Semaphore,           //the handle of object to wait for
        INFINITE             //wait forever
    );
    if (dw==WAIT_OBJECT_0) {
        ptr[1] ++;
        //release the Semaphore
        RtReleaseSemaphore(Semaphore, 1, 0);
    } else {
        //the Semaphore was abandoned
        printf("Semaphore was abandoned\n");
        totalFailures ++;
    }
    i++;
}
}
```