REAL TIME OPERATING SYSTEM PROGRAMMING-II: Windows CE, OSEK and Real time Linux

Lesson-6:
WCE Exceptions, Notifications and IPC Objects

1. Exceptions and Notifications

Exceptions and Notifications

- For which there is no waiting process or thread

Exceptions and Notifications

- Exception handling signals and notification signals
- Notification examples— notifications from timer, serial device detect, power up, system event.
- Notification can generate dialog or sound or run a notification responding function

Signaling (Exception) Functions

- RaiseException using the 32-bit exception code, exception flags, number of arguments, and 32-bit constant for array of long pointer for the arguments.
- Each argument passes the data to the exceptional responding routine (like ISR)

Signaling (Notification) functions

- CeSetUserNotificationEx using Handle CE_NOTIFICATION_TRIGGER object for notification, and CE_USER_NOTIFICATION object (object pointer defines action flags, dialog title, dialog text, sound and other details)

Signaling (Notification) functions

- CeGetUserNotificationEx using the Handle for notification and long pointer for CE_USER_NOTIFICATION object

Signaling (Notification) functions

- CeClearUserNotificationEx using the Handle for notification to acknowledge a notification by notification responding function

2. Critical Section Functions

Critical Section Functions

- InitializeCriticalSection

—to initialize a critical section

- EnterCriticalSection

— to enter a critical section

- LeaveEnterCriticalSection

— to exit a critical section

Critical Section Functions

- TryEnterCriticalSection

– to try to enter a critical section

- DeleteEnterCriticalSection

–to delete a critical section

3. IPC Functions

IPC Objects

- For which there is waiting process or thread

An IPC object release

- A process (thread or scheduler, task or ISR) generates some information by or value and sends event or semaphore or message into queue or a single IPC or multiple objects as output so that it lets another process waiting for that object in order to take note or use the object

An IPC object (s) wait

- A process waits for an IPC or object (s) in order to take note or use the object (s)

WCE IPC Objects

- Events,
- semaphores including mutex-semaphores
- message-queues for threads synchronization

4. Semaphore Functions

Semaphore functions

- CreateSeamphore

—to create the semaphore

ReleaseSemaphore

– to release semaphore to let waiting thread code unblock

- CreateMutex

– to create the mutex

- ReleaseMutex

– to release mutex to let waiting thread code unblock

5. Message Queue Functions

Message Queue Functions

- CreateMsgQueue

– to create the message queue)

- OpenMsgQueue

–to open a message queue

- ReadMsgQueue

–to read from the queue

- GetMsgQueueInfo

– to query the queue

Message Queue Functions

- WriteMsgQueue

–to write into the queue

- CloseMsgQueue

–to close an open message-queue

6. Event Functions

Event functions

- CreateEvent

—to create the event

- SetEvent

— event set on single occurrence of the event and don’t auto-reset till waiting thread unblocks

- ResetEvent

– Event auto-resets on unblocking of thread This function forces reset of the event and unblock the thread waiting for it

Event functions

- PulseEvent

—to set the event and then reset the event by unblocking all waiting threads for that event

- SetEventData

— using event-handle and 32-bit data in the arguments

- GetEventData

— using event-handle to get the event data

4. IPC Object Functions

Wait Single and multiple object

- WaitForSingleObject

using object Handle and 32-bit waiting time value in milliseconds in the arguments

- WaitForMultipleObjects

using count number for objects (events or mutexes), pointer to array of object Handles, boolean

Wait all

- WaitAll (if true then wait for all, in WCE must be set to false) and 32-bit waiting time value in milliseconds in the arguments
- Each object-handle is a long pointer. waiting time value = INFINITE disables the timeout specification for wait for the multiple objects.

Wait for multiple message objects

- MsgWaitForMultipleObjectsEx using count number for message objects, long pointer to array of object Handles
- boolean WaitAll (if true then wait for all, in WCE must be set to false), 32-bit waiting time value in milliseconds in the arguments and 32-bit flags for WakeMask.

Summary

We learnt

- WCE provides for exception handling signals, notification signals, event functions (for threads of a process), semaphores, message queues, wait single object, multiple object and multiple message-object functions.

End of Lesson-6 of chapter 10 on

WCE Exceptions, Notifications and IPC Objects