­­SETR2 -- Cosas a tener en cuenta de:

-CoOs:

-código:

SystemInit(); //Inicializa reloj

CoInitOS(void); //Inicializa variables SO

CreateSystemObjects(); //Inicializa Sem, flags, colas…

CreateUserTasks(); //Creación de las tareas

CoStartOS(void); //Start planificador (Tareas creadas antes de lanzarlo)

Bucle while infinito //Nunca debería llegar aquí

-Tareas:

Implementan típicamente un bucle infinito que ejecuta código y luego se bloquea (esperando un evento)

Bloqueos debidos a:

Semáforos, banderas, secciones críticas, o recepción de mensaje

Hay apropiación si existe tarea de mayor prioridad

Múltiples tareas con igual prioridad rotan en su ejecución

Crear/Destruir una tarea implica añadir/retirar una TCB

CoIdleTask(void) //tarea con mínima prioridad + bucle infinito //hook.c

-código:

OS\_STK pilaTareas[nºTareas][64];

CoCreateTask(Tarea,arg,prioridad,&pilaTareas[0][63],64); //Crear tarea

CoTimeDelay(Horas,minutos,segundos,milisegundos); //Espera

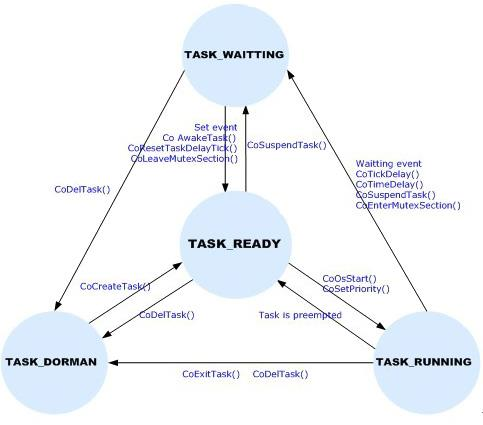
CoExitTask(); //Finaliza tarea

CoDelTask(); //Elimina tarea

CoSuspendTask(); //Suspende tarea

CoAwakeTask(); //Despierta tarea

ESTADOS DE UNA TAREA



-Timers Software:

Función que se ejecuta cada X tiempo //Periódicas o One-Shot

No puede llamar a funciones que la bloqueen

Se suelen usar para postear banderas, semáforos, colas…

Los timers One-Shot suelen usarse como WatchDog Timer //Ir refrescando

-código:

OS\_TCID miTimerID;

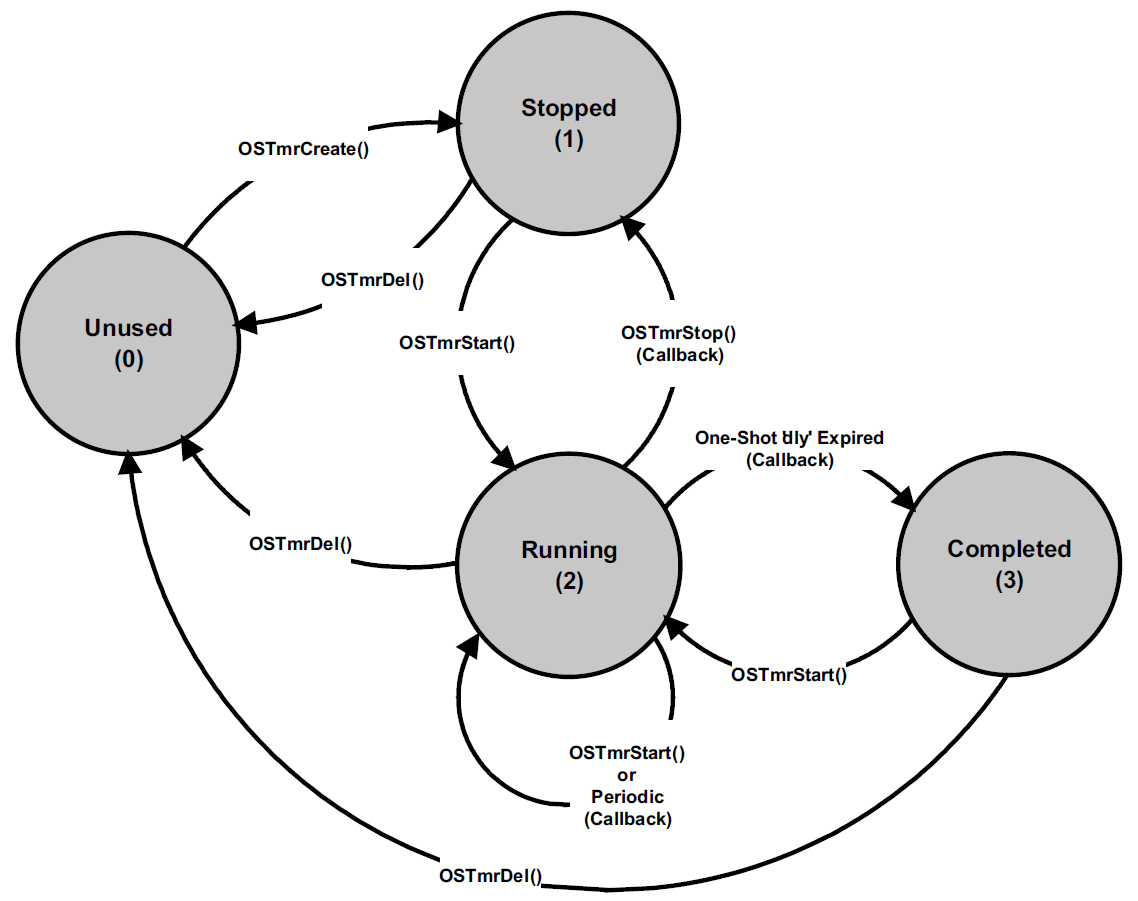
miTimerID = CoCreateTmr(TMR\_TYPE\_X,tmrStart,tmrReload,Tarea);

CoStartTmr(miTimerID); //Crea

CoStopTmr(miTimerID); //Para

CoDelTmr(miTimerID); //Elimina

DIAGRAMA DE ESTADO DE LOS TIMER SOFTWARE



-Secciones críticas:

Sección que se ejecuta sin que otra tarea o interrupción interfiera

Inhabilita interrupciones y paraliza el planificador

Prohibido llamar a funciones bloqueantes //Cuelgue

-código:

CoSchedLock(); //Bloquea planificador

CoSchedUnlock(); //Desbloquea planificador

-Banderas:

Se usan para comunicar eventos en el sistema entre tareas

Consumir una bandera implica que vuelve a 0 después de ser esperada

status -> E\_INVALID\_ID //E\_TIMEOUT //E\_OK

error -> E\_INVALID\_PARAMETER //E\_TIMEOUT //E\_FLAG\_NOT\_READY //E\_OK

-código:

OS\_FlagID banderaID;

StatusType error;

banderaID = CoCreateFlag(autoReset, initialState); //(0->manual/1->se consume,0/1)

CoSetFlag(banderaID); //Activar

CoClearFlag(banderaID); //Desactivar

status = CoWaitForSingleFlag(banderaID,TimeOut); //TimeOut = 0 -> espera indefinida

flagID = CoWaitForMultipleFlags(flag1 | flag2, OPT\_WAIT\_X, TimeOut, &error);

//X -> ALL – ANY // And – Or // Todas – Alguna

-Semáforos:

Controlan el acceso a recursos compartidos

-código:

OS\_EventID semafo;

Semafo = CoCreateSem(ContadorIni, ContadorMax, EVENT\_SORT\_TYPE\_X);

//Tareas //X -> FIFO – PRIO //Se encolan en una fifo – se ordenan según prioridad

status = CoPendSem(semafo, TimeOut); //Espera a que se libere el semáforo

CoPostSem(semafo); //Libera el semáforo

-Mensajes:

-----------------