Requerimientos de implementación para Sistema Operativo

Maestría en Sistemas Embebidos
Implementación de Sistemas Operativos I

Docente
Mg. Ing. Gonzalo E. Sanchez

FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Tabla de contenido

Revisión	Cambios realizados	Fecha
1.0	Creación del documento	30/03/2020
1.1	Actualización de parámetros de estructura tarea	07/04/2020
1.2	Actualización de tabla de contenidos	11/05/2020
1.3	Requerimiento 4.b modificado	13/06/2020

Requerimientos para implementación de Sistema Operativo

Recomendaciones

Para la aprobación de la asignatura Implementación de Sistemas Operativos (I) es mandatoria la implementación de la totalidad de las funcionalidades y características las cuales se listan en la sección siguiente. Las características que el estudiante desee implementar más allá de los requerimientos básicos están bajo exclusiva responsabilidad del mismo, y no se contempla la sustitución de una funcionalidad básica listada en los requerimientos por otra que el estudiante considere oportuna.

Luego de evaluarse las características y funcionalidades básicas listadas en los requerimientos, serán evaluadas las funcionalidades y características extra que el estudiante haya implementado, impactando en su nota de aprobación final.

Listado de requerimientos

Los requerimientos aquí listados están pensados para facilitar la implementación del Sistema Operativo de cada estudiante. Suponen una base mínima que debe cumplirse, y no son bajo ningún concepto restricciones. Si el estudiante desea sobrepasar estos requerimientos mínimos con funcionalidades adicionales, tiene la libertad de hacerlo. Bajo ningún concepto se tomará por aprobado el estudiante que no cumpliere con los requerimientos minimos.

1. El sistema operativo (de aquí en más nombrado como OS) será del tipo estático, entendiéndose por este término que la totalidad de memoria asociada a cada tarea y al kernel será definida en tiempo de compilación.
2. La cantidad de tareas que soportara el OS será ocho (8).
3. El OS debe administrar las IRQ del hardware.
4. El kernel debe poseer una estructura de control la cual contenga como mínimo los siguientes campos:
    a. Último error ocurrido
    b. Estado de sistema operativo, por ejemplo: Reset, corriendo normal, interrupción, etc.
    NOTA: Los estados mencionados no son obligatorios sino solamente un ejemplo.
    c. Bandera que indique la necesidad de ejecutar un scheduling al salir de una IRQ.
    d. Puntero a la tarea en ejecución.

   e. Puntero a la siguiente tarea a ejecutar.

5. Cada tarea tendrá asociada una estructura de control que, como mínimo, tendrá los siguientes campos:
   a. Stack (array).
   b. Stack Pointer.
   c. Punto de entrada (usualmente llamado *entryPoint*).
   d. Estado de ejecución.
   e. Prioridad.
   f. Número de ID.
   g. Ticks bloqueada.
6. Los estados de ejecución de una tarea serán los siguientes:
   a. Corriendo (Running).
   b. Lista para ejecución (Ready).
   c. Bloqueada (Blocked).
   d. Suspendida (Suspended) - *Opcional*
7. El tamaño de stack para cada tarea será de 256 bytes.
8. La implementación de prioridades será de 4 niveles, donde el nivel cero (0) será el de más alta prioridad y tres (3) será el nivel de menor prioridad.
9. La política de scheduling entre tareas de la misma prioridad será del tipo Round-Robin.
10. El tick del sistema será de 1 [ms].
11. El OS debe tener *hooks* definidos como funciones **WEAK** para la ejecución de código en las siguientes condiciones:
    a. Tick del sistema (*tickHook*).
    b. Ejecución de código en segundo plano (*taskIdle*).
    c. Error y retorno de una de las tareas (*returnHook*).
    d. Error del OS (*errorHook*).
12. El OS debe poseer una API que contenga como mínimo las siguientes funciones:

a. Función de retardos (delay).

b. Semáforos binarios.

c. Colas (*queue*).

d. Secciones críticas

e. Forzado de Scheduling (*cpu yield*).