

## Prácticas Sistemas Electrónicos

---

### Prácticas

---

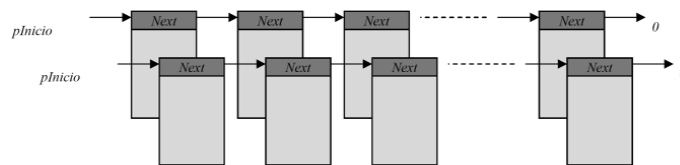
- **Práctica 1.-**

#### **Realización de un emulador de microKernel sobre Plataforma PC**

```
#####
Implementación de un Emulador
de MicroKernel sobre Plataforma PC
S i s t e m a s   E l e c t r o n i c o s
Nombre Apellido1 Apellido2
Curso 
#####
Menú de comandos
-----
? - Presenta el menú de opciones
i - Inicializa el sistema emulado
c - Crea una tarea
e - Elimina una tarea
r - Retarda una tarea
n - Tarea espera por un evento
v - Genera evento para una tarea
m - Envía mensaje a tarea
l - Tarea espera mensaje
s - Muestra las listas de BCPs
b - Cambia prioridad de una tarea
t - Ejecuta TICK del sistema Emulado
```

---

- El uKernel consiste en un núcleo de un sistema operativo en tiempo real, por lo que debe ser expropiativo basado en prioridades, es decir, siempre debe ejecutar el proceso de mayor prioridad.
- Se gestionarán las siguiente listas enlazadas:
  - bloques libres
  - procesos preparados
  - procesos retardados
  - procesos esperando



## Estructuras y Proceso nulo

### ESTRUCTURA del BCP

Puntero a la pila de la tarea  
 Siguiente BCP en Retardados  
 Anterior BCP en Retardados  
 Siguiente BCP en Preparados  
 Siguiente en lista de todos los BCP

```
Retardo    typedef struct bcp {
Prioridad
Estado
Mensaje
Nombre
    unsigned char *pila;
    struct bcp *prox;
    struct bcp *ant;
    struct bcp *hebra;
    struct bcp *todos;
    unsigned int retardo;
    unsigned char prioridad;
    unsigned char estado;
    void *mensaje;
    char nombre[10];
} BCP;
```

### ESTRUCTURA del BUZÓN

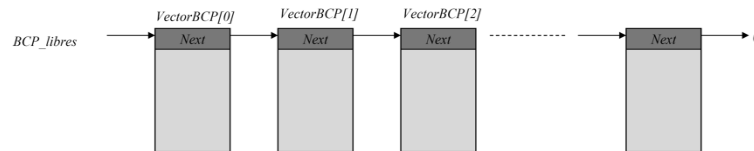
Puntero al primer mensaje  
 Puntero al último mensaje  
 Puntero al primer proceso  
 Puntero al último proceso  
 Puntero a la lista de buzones libres

### ESTRUCTURA del MENSAJE

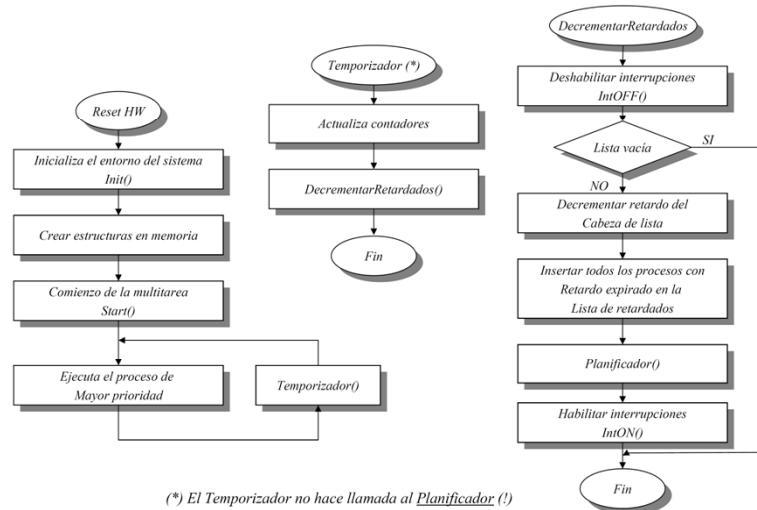
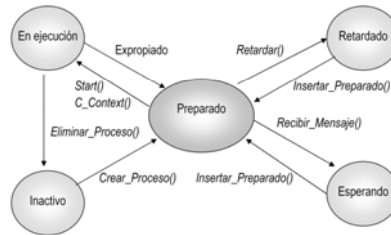
Puntero a la lista de Mensajes  
 Puntero al cuerpo del mensaje

```
Void Proceso_Nulo(void) {
    while (TRUE) {
        ; /* Nada */
    }
}
```

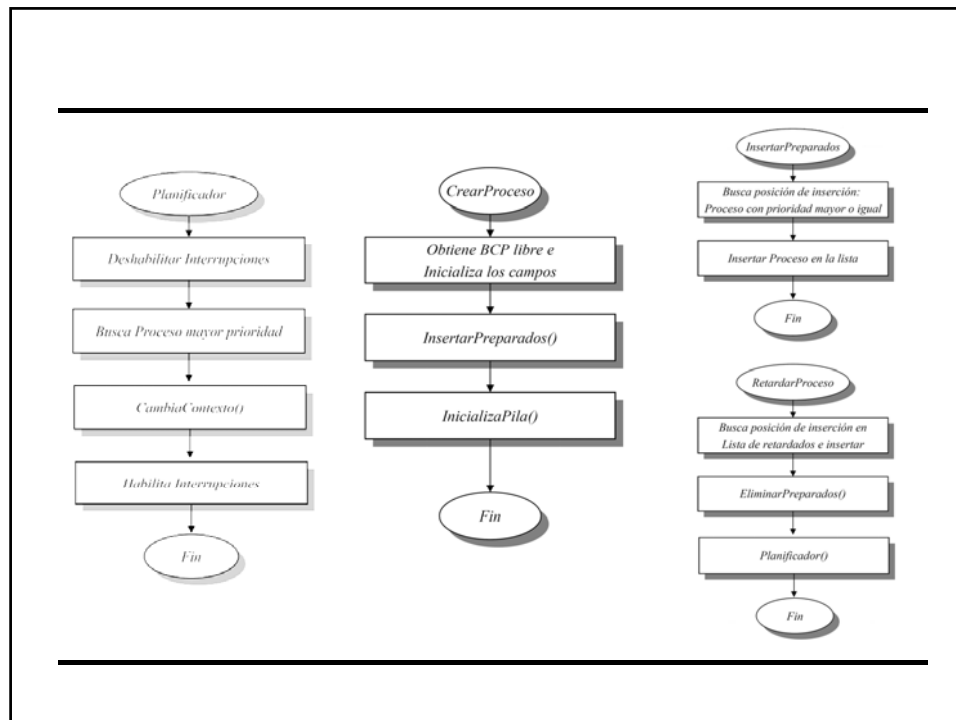
- Lista de BCP libres



- Estados de los procesos



(\*) El Temporizador no hace llamada al Planificador (!)



## Inicialización del sistema

Función	Init()
Tipo	Servicio Público
Descripción	Crea el entorno para la ejecución multitarea del sistema
Parámetros	Ninguno
Fichero	uKernel.c
<u>FUNCIONALIDAD</u>	
<ul style="list-style-type: none"> <li>• Crea e inicializa la lista de BCP libres</li> <li>• Crea las estructuras de buzones y mensajes</li> <li>• Crea el proceso nulo.</li> <li>• Crea el mecanismo de soporte a interrupciones</li> </ul>	

Función	Start()
Tipo	Servicio Público
Descripción	Transfiere el control de la CPU al proceso de mayor prioridad
Parámetros	Ninguno
Fichero	uKernel.c
<u>FUNCIONALIDAD</u>	
<ul style="list-style-type: none"> <li>• Ejecuta la tarea de mayor prioridad (hasta siguiente evento)</li> </ul>	

## Crear proceso

<i>Función</i>	<i>CrearProceso()</i>
<i>Tipo</i>	<i>Servicio público</i>
<i>Descripción</i>	<i>Crea Proceso</i>
<i>Parámetros</i>	1.- Puntero a dirección del código del proceso 2.- Nombre del proceso 3.- Puntero a la dirección de la pila 4.- Prioridad del proceso
<i>Fichero</i>	<i>uKTask.c</i>
<u>FUNCIONALIDAD</u> <ul style="list-style-type: none"> <li>• Obtener nuevo BCP</li> <li>• Deshabilitar interrupciones</li> <li>• Rellenar BCP con nuevos datos</li> <li>• Inicializa la pila del proceso</li> <li>• Insertar BCP en lista de preparados (lista ordenada por prioridad)</li> <li>• Habilitar interrupciones</li> </ul>	

## Inicializar pila BCP

<i>Función</i>	<i>InicializaPila()</i>
<i>Tipo</i>	<i>Servicio privado. Función dependiente del HW</i>
<i>Descripción</i>	<i>Inicializa la pila del Proceso (BCP)</i>
<i>Parámetros</i>	1.- Puntero a dirección del código del proceso 2.- Puntero a Pila 3.- Puntero a BCP
<i>Fichero</i>	<i>uKTask.c</i>
<u>FUNCIONALIDAD</u> <ul style="list-style-type: none"> <li>• Coloca la dirección de comienzo del código del proceso en la posición correspondiente al contador de programa en la pila del nuevo proceso</li> </ul>	
<u>COMENTARIOS</u> <ul style="list-style-type: none"> <li>• La pila de cada proceso se inicializa sólo una vez y en el momento de su creación</li> </ul>	

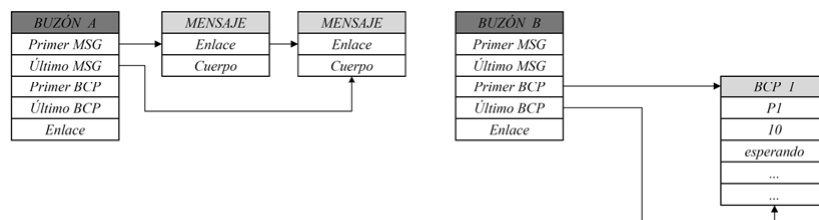
## Eliminar o retardar proceso

Función	<i>EliminarProceso()</i>
Tipo	<i>Servicio Público</i>
Descripción	<i>Elimina el BCP del proceso invocante</i>
Parámetros	<i>Ninguno</i>
Fichero	<i>uKTask.c</i>
<b>FUNCIONALIDAD</b> <ul style="list-style-type: none"> <li>• <i>Desactiva Interrupciones</i></li> <li>• <i>Elimina el BCP de la lista de preparados: EliminarPreparados()</i></li> <li>• <i>Sitúa el BCP en la lista de Libres</i></li> <li>• <i>Invoca al Planificador()</i></li> </ul>	

Función	<i>RetardarProceso()</i>
Tipo	<i>Servicio Público</i>
Descripción	<i>Retarda o duerme el proceso invocante durante un número de ticks</i>
Parámetros	<i>1.- Número de Tics</i>
Fichero	<i>uKTask.c</i>
<b>COMENTARIOS</b> <ul style="list-style-type: none"> <li>• <i>Permite que un proceso de mayor prioridad ceda voluntariamente el control de la CPU al siguiente proceso de menor prioridad</i></li> </ul>	

## Comunicación y sincronización entre procesos

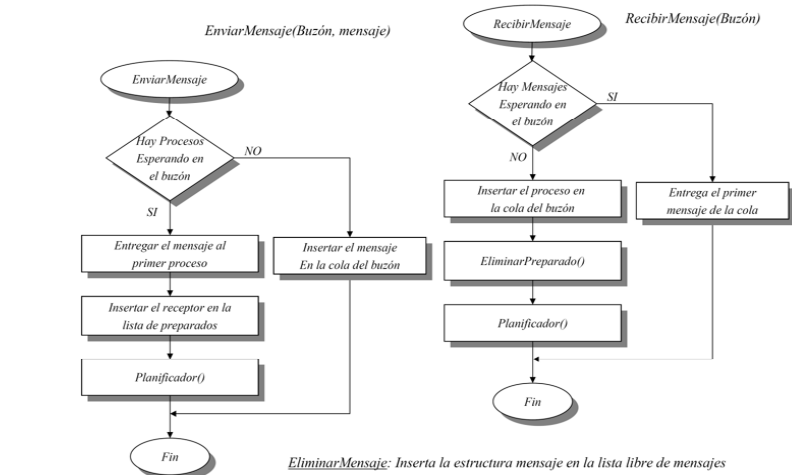
- Se implementará un mecanismo de comunicación y sincronización mediante buzones y mensajes. Por lo que será necesario gestionar estos recursos
  - *Listas encadenadas formando colas de tipo FIFO*
  - *Cada buzón está formado por una estructura capaz de encabezar una cola de mensajes que esperan ser entregados*
  - *Por otro lado, se tiene una lista encadenada de procesos que esperan mensajes.*



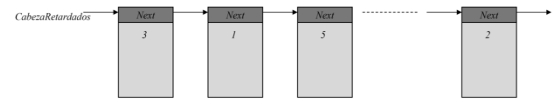
## Crear buzón y crear mensaje

Función	CrearBuzon()
Tipo	Servicio Público
Descripción	Crea un buzón
Parámetros	Ninguno
Fichero	uKMsg.c
<u>FUNCIONALIDAD</u>	
<ul style="list-style-type: none"> <li>Obtiene un Buzón libre de la lista</li> <li>Inicializa los campos a NULL</li> </ul>	
<u>COMENTARIOS</u>	
<ul style="list-style-type: none"> <li>Normalmente se crea un buzón para cada evento que se quiere atender (p.e. Interrupciones)</li> </ul>	

Función	CrearMensaje()
Tipo	Servicio Público
Descripción	Crea un mensaje
Parámetros	Ninguno
Fichero	uKMsg.c
<u>FUNCIONALIDAD</u>	
<ul style="list-style-type: none"> <li>Obtiene un mensaje libre de la lista</li> <li>Inicializa los campos</li> </ul>	

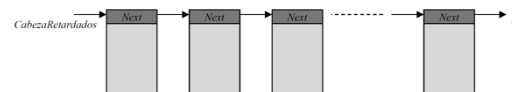


- Lista de BCP retardados

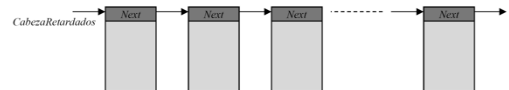


NOTA: Retardos relativos (!)

- Lista de buzones libres



- Lista de mensajes libres



## Estructura del código

