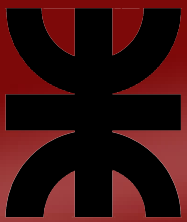




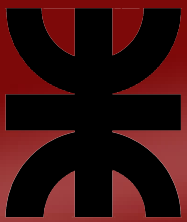
FreeOSEK

Sistema Operativo de Tiempo Real
de la Computadora
Industrial Abierta
Argentina



OSEK-OS Estático (1)

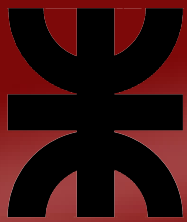
- OSEK-OS es un sistema operativo estático, tanto
 - tareas, sus prioridades, etc;
 - cantidad de memoria que utilizanson definidos antes de compilar el código en un proceso que se llama **generación**.
- No es posible
 - **crear** una tarea de forma dinámica,
 - **cambiar** la prioridad a una tareas.



OSEK-OS Estático (2)

- No es posible que una tarea no sea cargada porque no hay más memoria.
- Las tareas tienen una prioridad asignada de **antemano**, por ende una tarea tendrá siempre esa misma prioridad.

Esto es importante en sistemas de control críticos con requerimientos **donde los fallos no son aceptables** o tienen un altos costos.



OSEK: OIL

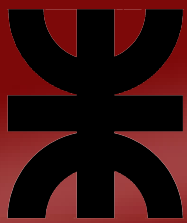
OSEK-VDX definió otro estándar llamado *OSEK Implementation Language* (AKA **OIL**)

Es un lenguaje textual donde se indica las características del OS, Tareas, Prioridades, etc



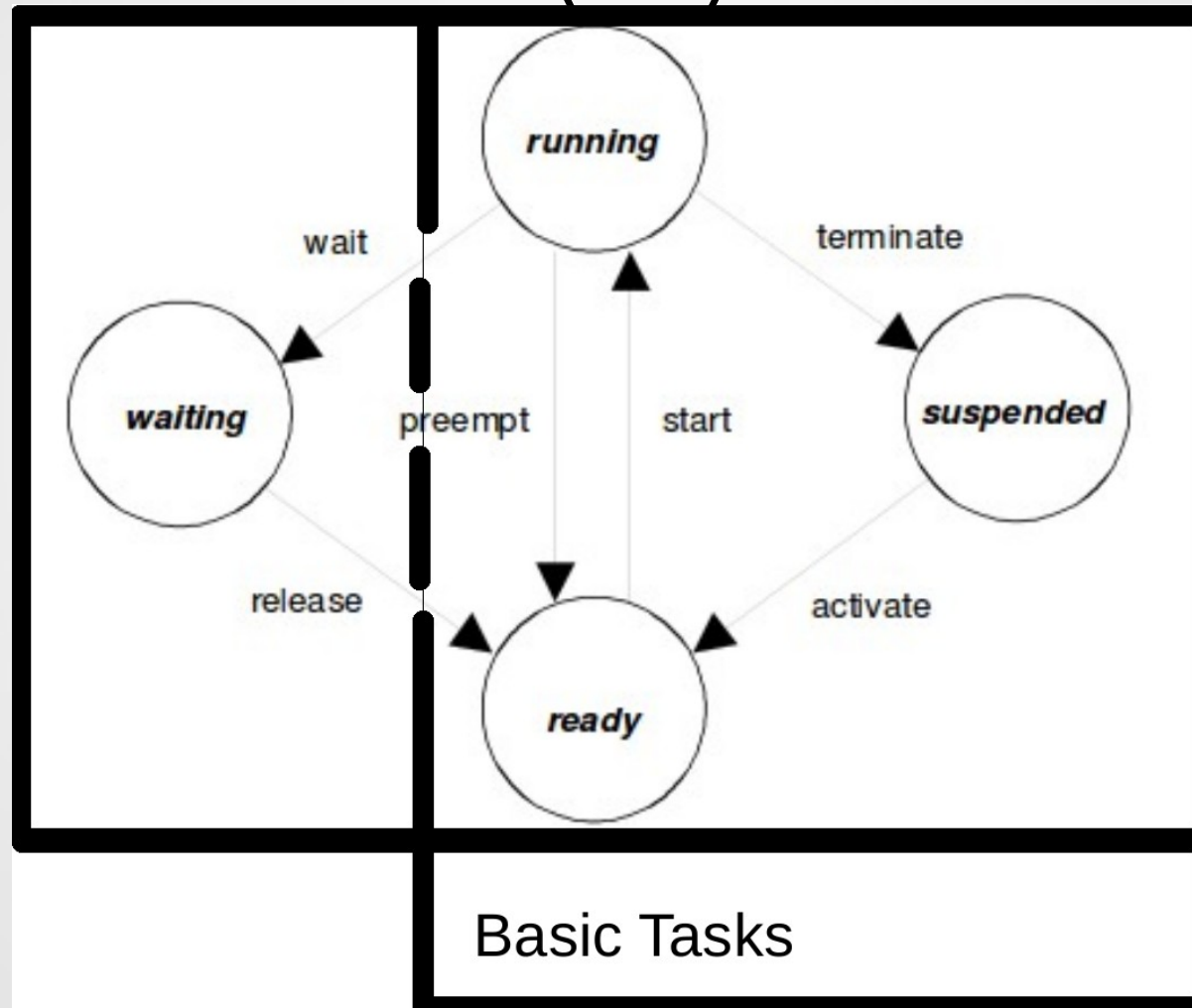
OSEK Tareas

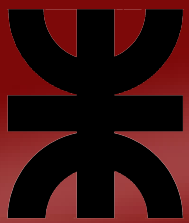
- En general en los RTOS las tareas realizan sus actividades y terminan.
- Para esperas se usan eventos



OSEK Tareas - Estados

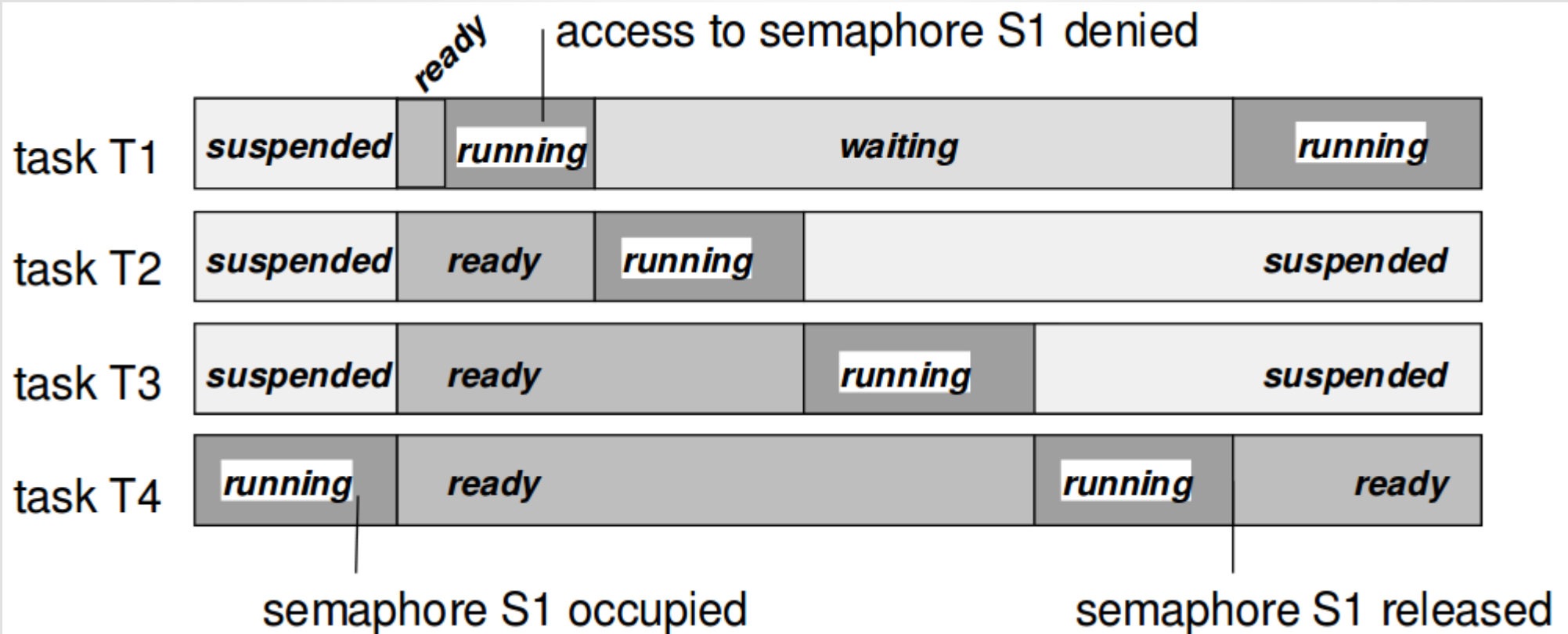
- Cada tarea en OSEK-VDX se encuentra siempre en uno de 4 (o 3) estados:

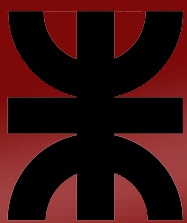




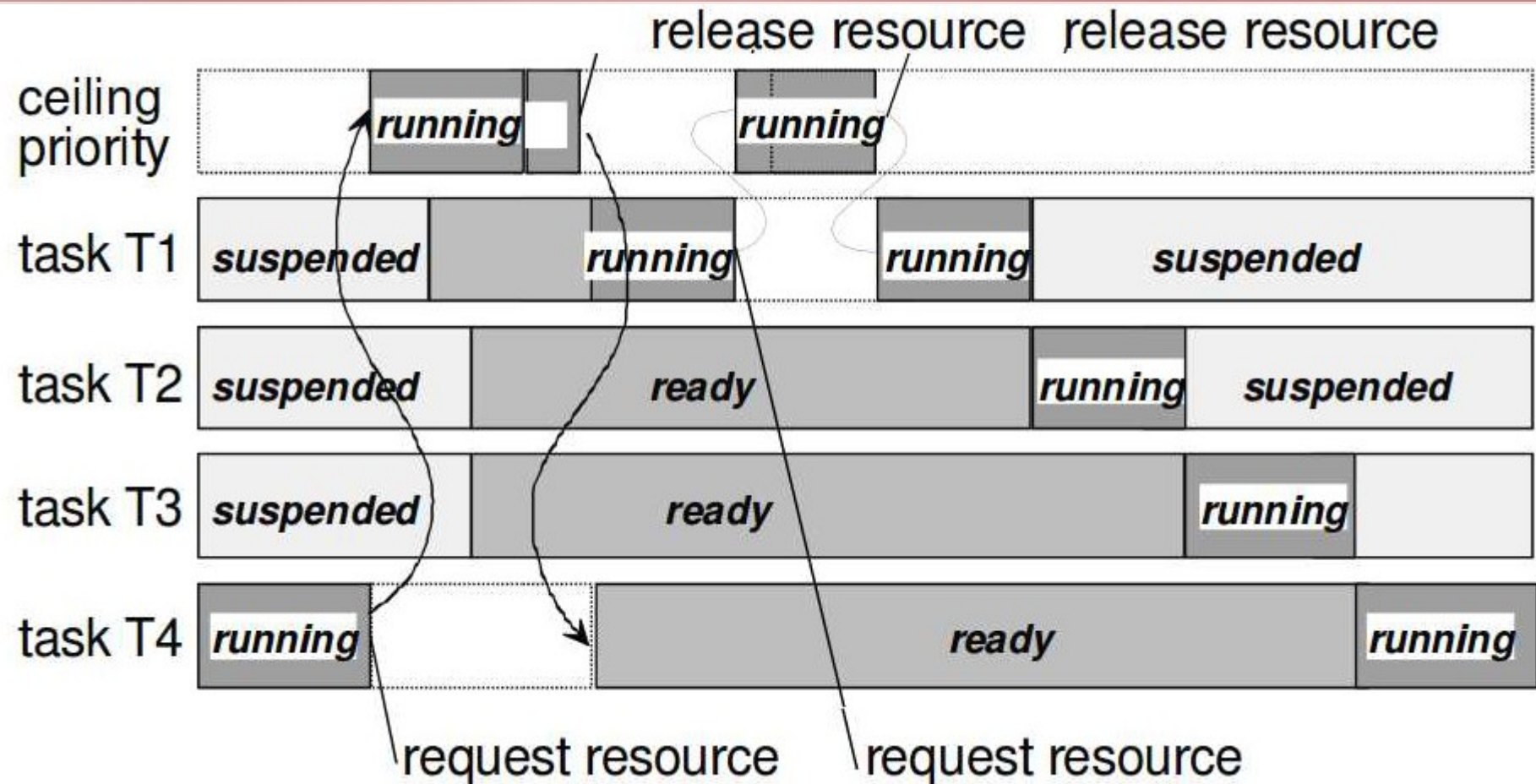
Recursos – Inversión de prioridades

- Con semáforos:

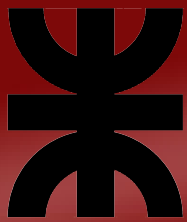




OSEK – OSEK Priority Ceiling Protocol



T4 corre con prioridad baja, cuando pide acceso al recurso el SO le otorga **mientras lo use** una prioridad similar a la de T1



OSEK – Recursos (1)

- Para la utilización de recursos los mismos deben ser declarados en OIL e indicado que recursos serán utilizados por cada tarea
- se puede acceder a los recursos mediante:
 - `GetResource()` ,
 - `ReleaseResource()`



OSEK – Recursos (2)

```
TASK(TaskA) {  
    SCHEDULE = NON;  
    ACTIVATION = 1;  
    PRIORITY = 5;  
    STACK = 128;  
    TYPE = BASIC;  
    RESOURCE = Res1;  
    RESOURCE = Res2;  
}  
TASK(TaskB) {  
    SCHEDULE = NON;  
    ACTIVATION = 1;  
    PRIORITY = 7;  
    STACK = 128;  
    TYPE = BASIC;  
    RESOURCE = Res1;  
}  
TASK(TaskC) {  
    SCHEDULE = NON;  
    ACTIVATION = 1;  
    PRIORITY = 3;  
    STACK = 128;  
    TYPE = BASIC;  
    RESOURCE = Res2;  
}  
  
TASK(TaskA) {  
    /* some code */  
    GetResource(Res1);  
    /* perform actions using resource 1 */  
    /* during this section TaskA has priority 7 */  
    ReleaseResource(Res1);  
    TerminateTask();  
}  
TASK(TaskC) {  
    /* some code */  
    GetResource(Res2);  
    /* perform actions using resource 2 */  
    /* during this section TaskA has priority 5 */  
    ReleaseResource(Res2);  
    TerminateTask();  
}
```



OSEK – Alarmas (1)

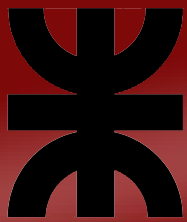
- Utilizadas para realizar una acción luego de determinado tiempo. Acciones como:
 - Activar una tarea
 - Setear un evento de una tarea



OSEK – Alarmas (2)

```
ALARM ActivateTaskC {
    COUNTER = SoftwareCounter;
    ACTION = ACTIVATETASK {
        TASK = TaskC;
    }
    AUTOSTART = FALSE;
}
ALARM SetEvent1TaskA {
    COUNTER = SoftwareCounter;
    ACTION = SETVENT {
        TASK = TaskA;
        EVENT = Event1;
    }
    AUTOSTART = FALSE;
}
```

```
TASK(TaskB) {
    /* some code */
    SetRelAlarm(ActivateTaskC, 100, 100);
    SetRelAlarm(SetEvent1TaskA, 150,
200);
    TerminateTask();
}
TASK(TaskC) {
    static int counter = 0;
    if (counter ++ > 10) {
        CancelAlarm(ActivateTaskC);
    }
    /* do something */
    TerminateTask();
}
```



Biblografía

Bibliografía

Cerdeiro, Mariano.

Introducción a OSEK-OS. El Sistema operativo del CIAA-Firmware. 2015.

ISBN 978-987-45523-6-5

Cerdeiro, Mariano . Breve introducción a OSEK-VDX. *Un sistema operativo de tiempo real estandarizado.* 2014.

OSEK/VDX steering committee.

OSEK/VDX Operating System Specification 2.2.3. 2005

- OSEK/VDX steering committee.

OIL: OSEK Implementation Language Version 2.5. 2004.