

# Charla Real Time Linux

---

**David Martinez Diaz -- Charla DSD**

Marcello Cinque - Master Program on Software Development

## Indice:

- [Charla Real Time Linux](#)
  - [Indice:](#)
  - [¿Que significa Real Time?](#)
  - [Ejemplos de Real Time](#)
  - [¿Por qué Linux Real-Time?](#)
  - [¿Por qué no es vanilla Linux kernel adecuado para Real-Time?](#)
  - [Latencia](#)
    - [Tipos de latencia:](#)
    - [Como es el esquema de latencia de tareas en linux](#)
    - [Como arreglamos el problema](#)
  - [Kernels Duales](#)
    - [Kernels duales: RTAI](#)
  - [PREEMPT\\_RT](#)
    - [Diferencias entre PREEMPT y PREEMPT\\_RT](#)

---

## ¿Que significa Real Time?

Real Time es algo que ocurre justo en este mismo tiempo, es un sistema del computador para que el usuario pueda interactuar con este.

---

## Ejemplos de Real Time

Un ejemplo de esto es el mercado de automoviles, donde los microcpus de los coches estan interconectados a la red, los sensores y cualquier otro tipo de aparato generalmente.

No hablamos del streaming, o llamada de voz, cuando hablamos de real time, queremos hacer especial hincapié en los programas de vuelos y los computadores de los aviones donde deben conocer todos los datos necesarios para poder controlar el sistema de las alas, motores en el mismo momento.

---

## ¿Por qué Linux Real-Time?

Debido a que tiene un gran numero de ventajas, por ejemplo, tiene una comunidad gigante de programadores y desarrolladores, una gran disponibilidad de librerias, drivers, entre otras características.

## ¿Por qué no es vanilla Linux kernel adecuado para Real-Time?

Esto se debe principalmente a que Linux es multi-hebra pero los ISRs y los drives utilizan los spinlocks para la sincronizacion, o puede causar interrupciones.

En lugar de estar durmiendo todo el rato preguntando si puede ejecutar sin tener que consumir los ciclos de reloj.

Basicamente cuando el recurso este libre, rapidamente es asignado dicho recurso de manera directa a otro proceso y asi continuamente. Tiene que ser utilizado por un recurso al inicio, sin embargo, aqui surge los problemas de exclusion mutua en tiempo real. Aunque estas soluciones son caracteristicas de mejores rendimientos y velocidad.

---

## Latencia

La medida es la diferencia entre el deasdo y el actual tiempo, esquema, es decir, las tareas de J esperan tener un tiempo teorico de T, sin embargo, obtenemos R, por lo tanto la latencia es la difrencia entre T-R.

### **Tipos de latencia:**

- La primera latencia es cuando se tarda la interrupcion de la generacion.
- La segunda es la no esperada.
- La tercera es la latencia esquematizada, cogido de la cuenta de los tests.

### Como es el esquema de latencia de tareas en linux

Tenemos una primera tarea con "alta prioridad", siendo en este caso task 2. Pero si hay una interrupcion mientras que el task 1 se ejecuta, tiene que forzosamente completarse antes de asignar cualquier otra tarea. Por defecto, el kernel is no siempre predecible, como resultado, el tiempo de activacion no puede ser predecido.

### Como arreglamos el problema

Necesitamos hacer que el kernel de linux sea predecible en favor de las tareas en tiempos reales.

Tenemos diferentes soluciones en linux:

- Configuracion del servidor
  - Configuarion del escritorio
  - Configuracion de baja latencia
  - Parche "PREEMPT\_RT"
  - Parches "Co-kernel"
- 

## Kernels Duales

Son ejecuciones de nivel bajo, directamente manejados por interrupciones de handlers. Son interrupciones no real time. No puede desactivar las interrupciones y las real time task no puede utilizar los kernel.

Kernels duales: RTAI

Es una interfaz de aplicacion a tiempo real, que utiliza un parche para linux que le permite al sistema operativo realizar dichas operaciones.

---

## PREEMPT\_RT

En vez de añadir otro kernel, el kernel de linux mismo es modificado para eliminar la latencia de las seleccion no predecibles.

### Diferencias entre PREEMPT y PREEMPT\_RT

En la configuracion de baja latencia, los "spinlock" son permitidos, pero evitan el tiempo limite, las secciones criticas no pueden ser predecidas.