Master en Computación

Programación Concurrente

Bloque II: Programación concurrente en POSIX

Tema 2. Sistema Operativo MaRTE OS

2.1 Sistemas operativos de tiempo real

En el pasado, muchos sistemas de tiempo real no necesitaban sistema operativo

Hoy en día, muchas aplicaciones requieren servicios de sistema operativo tales como:

- ***programación concurrente*, redes de comunicación, sistema de ficheros, etc.**

El comportamiento temporal de un programa depende fuertemente del comportamiento del sistema operativo

Durante el curso utilizaremos los sistemas operativos

- **MaRTE OS**
- **Linux**

Características de los RTOS

Los S.O. de Tiempo Real (RTOS) tienen características especiales

- **ya que su comportamiento temporal afecta al de la aplicación**

S.O. de Tiempo Compartido vs. RTOS:

	S.O. de Tiempo Compartido	RTOS
Capacidad	Potencia de cálculo	Planificabilidad
Respuesta Temporal	Rápida respuesta en promedio	Respuesta de peor caso garantizada
Sobrecarga	Reparto Equitativo	Estabilidad

Según el estándar POSIX la *definición de tiempo real en sistemas operativos* es:

- **"La habilidad del sistema operativo para proporcionar el nivel de servicio requerido con un tiempo de respuesta acotado."**

Limitaciones para TR de los SOs convencionales

- **No tienen políticas y protocolos de tiempo real**
- **Tiempos de respuesta no acotados para algunos servicios**
  - **planificador O(n)**
  - **memoria dinámica**
  - **...**
- **"*Swapping*"**
- **Latencia a interrupciones no acotada**
  - **un driver "mal escrito" puede deshabilitar interrupciones por un intervalo de tiempo arbitrariamente grande**
  - **IRQs compartidas**
- **Baja resolución del temporizador (p.e. 10ms)**
- **Uso de "*spinlocks*"**

2.2 Uso de Linux para Tiempo Real

Linux es un sistema operativo ampliamente utilizado (PCs, supercomputadores, servidores, ...)

- **este auge condujo a su utilización también en TR**
- **originalmente con muchas limitaciones**

Mejoras en el núcleo 2.6 para eliminar las limitaciones para TR:

- **mayor grado de implementación de la interfaz POSIX de TR**
- **mayor grado de expulsión en las llamadas al sistema**
- **planificador O(1)**
- **posibilidad de deshabilitar la memoria virtual**
- **...**

Parches aún no incluidos en las distribuciones estándares

- **"Complete Preemption Patch"**

Uso de Linux para Tiempo Real (cont.)

Cualquier distribución incluye el kernel 2.6: Ubuntu, Fedora, openSUSE, Debian, etc.

Distribuciones comerciales para tiempo real

- **Red Hat Enterprise MRG Realtime**
- **SUSE Linux Enterprise Real Time Extension**

El núcleo 2.6 con el "Complete Preemtion Patch"

- **permite utilizar Linux en aplicaciones de tiempo real laxo**

Aún así sigue teniendo importantes limitaciones:

- **servicios con respuesta no acotada (memoria dinámica)**
- **drivers "mal escritos" para TR**

El esfuerzo de adaptación de Linux a TR parece que va a continuar

- **¿alcanzará la conformidad con el perfil de *Sistema de Tiempo Real Multi-Propósito* en los próximos años?**

2.3 Linux con doble núcleo

Arquitecturas basadas en doble núcleo

- **micronúcleo que se incrusta en el núcleo de Linux**
- **asume el control de las interrupciones hardware**
- **Wind River Real-Time Core for Linux, RTLinux Free**

Arquitecturas de doble núcleo basadas en virtualización

- **La virtualización da preferencia al micronúcleo de tiempo real**
- **RTAI, Xenomai (utilizan Adeos)**

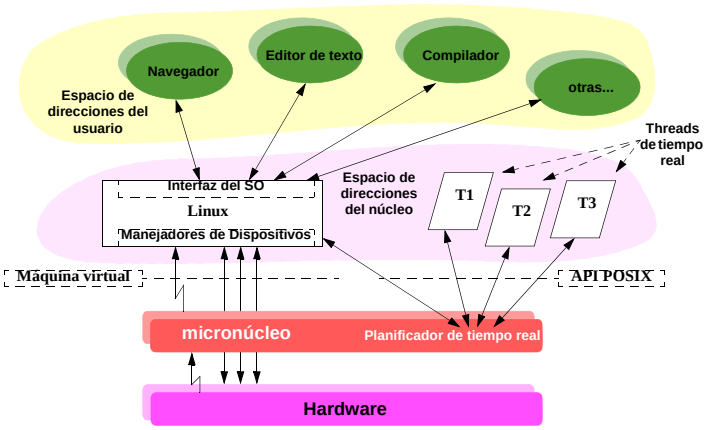
En ambos casos los threads con requisitos de tiempo real ejecutan bajo el control del micronúcleo

- **con preferencia sobre cualquier aplicación de Linux**

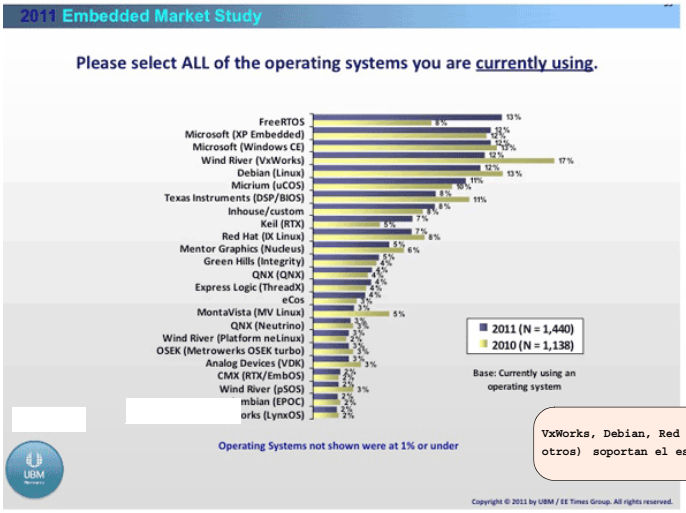
Apropiado para aplicaciones con requisitos de TR estricto

- **únicamente para los threads controlados por el micronúcleo**

Arquitectura de doble núcleo



2.4 Mercado de los RTOSs



VxWorks, Debian, Red Hat Linux y QNX (entre otros) soportan el estándar POSIX

2.5 MaRTE OS

Características principales:

- **Conforme con el perfil mínimo**
  - **Concurrencia a nivel de threads**

- **Todos los servicios tienen tiempos de respuesta acotados**
  - **también acotada la latencia de atención a interrupciones**
- **Único espacio de direcciones compartido por el núcleo y la aplicación**
- **Núcleo monolítico**
- **Escrito en Ada (pequeñas partes en C y ensamblador)**
- **Permite ejecutar aplicaciones concurrentes C, C++ y Ada**
- **Portable a diferentes plataformas (x86, MC68332 y Linux)**
- **Desarrollado en el grupo CTR. Código libre (GPL): *http://marte.unican.es***

Entorno de desarrollo

En un RTOS debemos diferenciar los siguientes aspectos:

- **núcleo (o *kernel*) del sistema operativo**
- **librerías y *drivers***
- **entorno de desarrollo**

El *entorno de desarrollo* es un conjunto de herramientas para:

- **editar**
- **compilar y enlazar**
- **depurar**
- **realizar medidas temporales**
- **diseñar y analizar la aplicación**
- **cargar la aplicación (sistemas empotrados)**
- **...**

Entorno de desarrollo cruzado

Los sistemas operativos de tiempo real se usan principalmente en aplicaciones empotradas

- **utilizan un entorno de desarrollo cruzado**

Equipo de desarrollo

- **ejecuta el entorno de desarrollo**

Equipo empotrado

- **ejecuta la aplicación**

Entorno de desarrollo cruzado (Arquitectura x86)

Entorno cruzado:

- ***Host*: PC con Linux**
- ***Target*: PC desnudo**

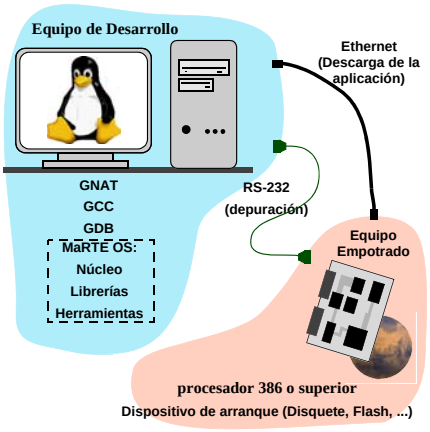
Creación de aplicaciones:

- **Basada en GCC y GNAT**
- **mgcc y mgnatmake**

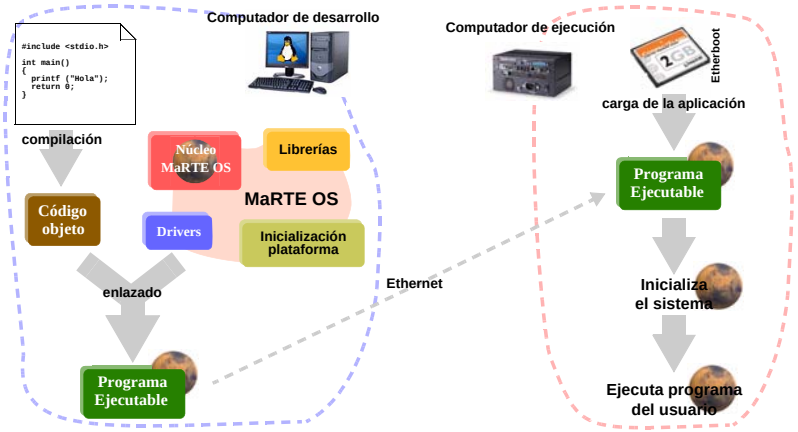
```
$ mgcc -g fich.o prog.c
$ mgnatmake prog.adb
```

Carga de la aplicación:

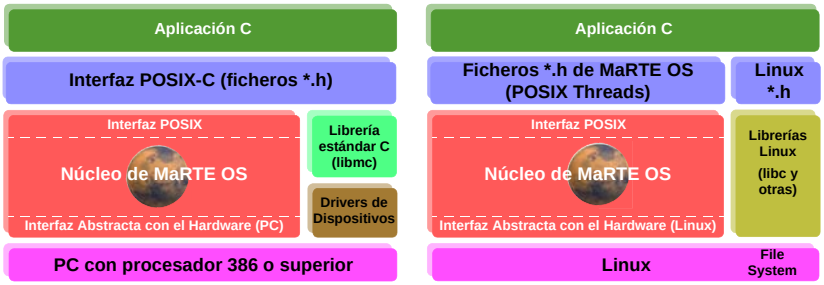
- **Ethernet**
- **Dispositivo de arranque**



Entorno de desarrollo cruzado (Arquitectura x86) (cont.)



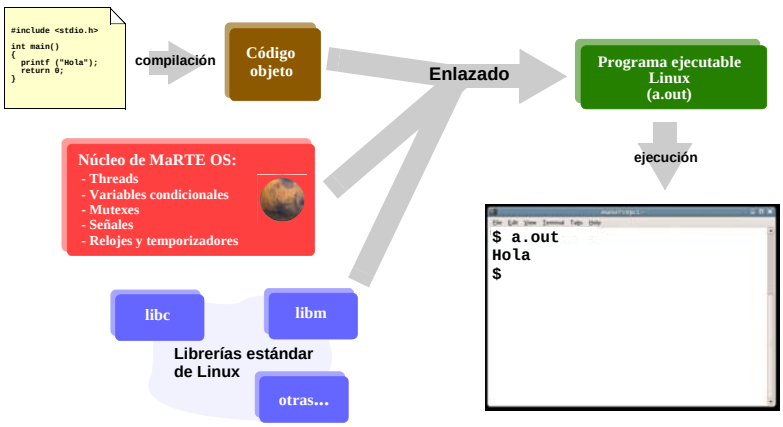
Arquitecturas "x86" y "Linux Lib"



Aplicación C ejecutando sobre MaRTE OS para PC

Aplicación C ejecutando sobre MaRTE OS para Linux

Generación de aplicaciones (Arquitectura "Linux_lib")



Limitaciones de la arquitectura "Linux Lib"

No se puede lograr comportamiento de tiempo real estricto

- **Aplicaciones controladas por el planificador de Linux**
- **Afectadas por "*swapping*", otras actividades del núcleo de Linux, etc.**

Medida del tiempo de ejecución de las tareas

- **No se tienen en cuenta los cambios de contexto con otros procesos Linux**

Operaciones de E/S bloqueantes

- **Cuando un thread se bloquea se bloquea toda la aplicación**

Descarga e Instalación de MaRTE OS

Descarga de `http://marte.unican.es` (sección "`Downloads`")

Instalación (seguir las instrucciones del fichero "`INSTALL`"):

1. **Instalar la versión apropiada del compilador GNAT**
2. **Instalar MaRTE OS**

No son necesarios permisos de administrador

© M. Aldea, M. González
oct.-13