

Introducción

- Los SSOO de TR son necesarios cuando el lenguaje de programación no ofrece los servicios necesarios para el tiempo real
- No todos los SSOO son válidos para el TR. Se necesitan unos requerimientos
- Existen métricas especializadas para los SSOO de TR que permiten comparar diferentes SSOO
- Linux para Tiempo Real

Requerimientos

- Garantizar la correcta ejecución de todas las tareas críticas
- Administrar adecuadamente el uso de los recursos compartidos
- Buen tiempo de respuesta a las tareas que no tengan plazo de terminación
- Recuperación ante fallos software y hardware
- Soportar los cambios de modo
- Buen tiempo de respuesta a las interrupciones
- Eficiencia en los cambios de contexto

Garantizas la ejecución

- Cumplir los plazos de ejecución de las tareas críticas, aún en el caso de sobrecarga
- Se necesita un criterio para saber si una aplicación cumplirá los plazos
- Predecir en tiempo de ejecución el cumplimiento de los plazos es un problema no resuelto
- Lo usual es utilizar políticas de planificación bien estudiadas que disponen de este criterio

Administración de recursos

- El compartir recursos precisa de una sincronización
- Se debe evitar la inversión de prioridad y los interbloqueos
- Los efectos de la sincronización se deben tener en cuenta en la política de planificación para seguir asegurando los plazos

Tareas sin plazo

- Es normal que en los STR coexistan tareas críticas y tareas sin plazo
- No siempre es adecuado dejarlas para el último lugar
- Se puede adelantar su ejecución siempre que se respeten los plazos de las tareas críticas
- El SO debe ser capaz de manejar de forma adecuada este tipo de tareas

Recuperación ante fallos

- Necesario en algunos sistemas de control para aumentar la fiabilidad
- Dificultad para recuperar los fallos y mantener las restricciones temporales, pero necesario en los STR estricto
- Facilidades
 - Funcionamiento distribuido
 - Procesos replicados
 - Grupos de puertos
 - Puntos de recuperación y vuelta atrás

Cambios de modo

- Los sistemas pueden tener varios modos de funcionamiento donde actúan tareas diferentes
- El cambio de modo puede suponer la mezcla de las tareas del modo anterior y del nuevo, con la sobrecarga que supone
- El sistema debe facilitar el cambio ordenado de un conjunto de tareas asegurando que no se incumple ninguna restricción temporal

Respuesta a las interrupciones

- Importantes en los STR pues se usan para detectar eventos en el entorno
- Factores que influyen en el tiempo de respuesta
 - Niveles de interrupción
 - Duración del tratamiento de las interrupciones
 - Tiempo de latencia de interrupción
- Los dos últimos factores son software y dependen del SO

Cambios de contexto

- Se produce cuando cambia la tarea en ejecución y su duración afectará al tiempo de respuesta
- En los STR debe ser lo mas corto posible, sobre todo si se realiza desde interrupción
- Para simplificarlo se utilizan procesos ligeros (hilos) que comparten los recursos del proceso
- Muchos SO para aplicaciones empotradas soportan un único proceso y múltiples hilos

Métricas

- Nos permiten comparar la bondad de distintos SO para el tiempo real
 - Métricas Rhealstone
 - Tiempo de latencia en la activación de procesos
 - Medida tridimensional
 - Real/Stone Benchmark

Métrica Rhealstone

- Tiempo de cambio de tarea
- Tiempo de prioridad
- Tiempo de latencia de interrupción
- Tiempo de transición en un semáforo
- Tiempo de ruptura
- Velocidad de flujo de datos en entrada/salida

Tiempo de latencia en la activación

- Tiempo de respuesta a la interrupción
 - Retardo hardware en atender la interrupción
 - Finalización de la instrucción actual
 - Tiempo de latencia de la interrupción
- Rutina de tratamiento de la interrupción
 - Pre-procesado
 - Rutina de servicio de la interrupción
 - Post-procesado
- Decisión de cambio de contexto
- Cambio de contexto

Medida tridimensional

- Tres medidas importantes en el TR
 - Velocidad de cálculos medido en MIPS1
 - Capacidad de tratar interrupciones en MIPS2
 - Rendimiento de Entrada/Salida en MIPS3
- Rendimiento equivalente para TR

$$MIPS_e = \sqrt[3]{MIPS1 \cdot (MIPS2 \cdot 100) \cdot MIPS3}$$

Real/Stone Benchmark

- Es un test artificial que simula un entorno de tiempo real
- Se miden las siguientes características
 - Sensibilidad del sistema
 - Priorizabilidad del sistema
 - Rendimiento del sistema para las E/S
- El resultado se expresa en unas gráficas variando el comportamiento según el número de procesos

Linux y Tiempo Real

- Inconvenientes más importantes
 - Desactivación de interrupciones durante un tiempo excesivo
 - La resolución del timer es de 10 ms y las activaciones se encolan a la salida de los servicios
 - La librería de pthreads no soporta las extensiones de tiempo real
- Extensiones de tiempo real en Linux
 - UTIME
 - KURT
 - RT Linux

KURT

- Utiliza UTIME para ofrecer un reloj con mayor granularidad
- Útil para aplicaciones que pueden soportar alguna demora en los plazos
- Modos de funcionamiento
 - **Dedicado**: Sólo los procesos de tiempo real
 - **Normal**: Todos los procesos son estándar de Linux. Modo en que se arranca
 - **Mixto**: Los procesos que no son de TR se ejecutan en background
- En los procesos de TR se conocen sus instantes de activación y sus plazos
- El paso a un modo de TR lo realiza un proceso ejecutivo después de calcular todos los instantes de activación de cada proceso
- Siguen existiendo fuentes de impredecibilidad

RT Linux

- Tiempo real estricto
- La aplicación se divide en dos partes
 - La de TR que se ejecuta en el núcleo de TR
 - La normal no de TR que corre en Linux
- Las dos partes se comunican por canales RT-FIFO
 - En linux se ven como periféricos
 - En el núcleo de TR las L/E son no bloqueantes y atómicas
- Las interrupciones hardware se emulan por software en el núcleo de TR
- La parte de tiempo real se escribe como módulos cargables
- Las tareas en cada módulo pueden tener su propio planificador:
 - Rate Monotonic
 - Earliest Deadline First