

Sistemas Informáticos de Tiempo Real

Programa

1. Conceptos básicos en los Sistemas de Tiempo Real

- 1.1. Introducción
- 1.2. Propiedades importantes
- 1.3. Clasificación
- 1.4. Justificación
- 1.5. Necesidades de los sistemas de tiempo real
 - 1.5.1. Especificación, análisis y verificación
 - 1.5.2. Programación en tiempo real
 - 1.5.3. Sistemas operativos de tiempo real
 - 1.5.4. Arquitecturas de tiempo real
 - 1.5.5. Comunicaciones de tiempo real
- 1.6. Aspectos particulares
 - 1.6.1. Sistemas empotrados
 - 1.6.2. Tolerancia a fallos
- 1.7. Ejemplos

2. Desarrollo de Sistemas de Tiempo Real

- 2.1. Introducción
- 2.2. Especificaciones
- 2.3. Diseño
- 2.4. Codificación.
 - 2.4.1. Lenguajes de programación
 - 2.4.2. Criterios generales en el diseño de lenguajes
- 2.5. Verificación
 - 2.5.1. Localización de errores
 - 2.5.2. Simuladores
 - 2.5.3. Herramientas
- 2.6. Prototipos

3. Programación de los Sistemas de Tiempo Real

- 3.1. Introducción
- 3.2. Programación concurrente
 - 3.2.1. Construcciones para la programación concurrente
 - 3.2.2. Procesos y objetos
 - 3.2.3. Representación de los procesos concurrentes
- 3.3. Comunicación y sincronización
 - 3.3.1. Basada en memoria compartida
 - 3.3.2. Basada en paso de mensajes
- 3.4. Fiabilidad y tolerancia a fallos
 - 3.4.1. Conceptos
 - 3.4.2. Técnicas de programación
- 3.5. Manejo de excepciones
 - 3.5.1. Representación de las excepciones
 - 3.5.2. Dominio de un manejador de excepción

- 3.5.3. Propagación de las excepciones
- 3.5.4. Modelos de reanudación y terminación
- 3.6. Consideraciones prácticas
 - 3.6.1. La noción de tiempo
 - 3.6.2. Acceso a un reloj
 - 3.6.3. Retardos y timeouts

4. Sistemas Operativos de Tiempo Real

- 4.1. Introducción
- 4.2. Requerimientos
- 4.3. Métricas
- 4.4. Ejemplo. Linux para tiempo real

5. Planificación en Tiempo real

- 5.1. Introducción
- 5.2. Conceptos
 - 5.2.1. Tipos de tareas
 - 5.2.2. Parámetros
 - 5.2.3. Tiempo real crítico y opcional
- 5.3. Modelo de referencia de los sistema de Tiempo Real
 - 5.3.1. Tiempo de ejecución
 - 5.3.2. Modelo de tareas periódicas
 - 5.3.3. Dependencias entre tareas
 - 5.3.4. Parámetros funcionales
 - 5.3.5. Planificadores
 - 5.3.6. Restricciones del modelo simple (resumen)
- 5.4. Políticas de planificación
 - 5.4.1. Planificadores cíclicos
 - 5.4.2. Planificación con prioridades
- 5.5. Planificadores cíclicos
 - 5.5.1. Supuestos
 - 5.5.2. Planificaciones estáticas
 - 5.5.3. Planificadores cíclicos con tramas
 - 5.5.4. Algoritmo para la construcción de planificadores estáticos
 - 5.5.5. Propiedades e inconvenientes
- 5.6. Planificadores con prioridades estáticas
 - 5.6.1. Planificador Rate Monotonic (RM)
 - 5.6.2. Análisis del tiempo de respuesta
 - 5.6.3. Planificador Dead-Line Monotonic (DM)
 - 5.6.4. Tiempo de ejecución en el peor de los casos
 - 5.6.5. Mejoras del modelo simple
- 5.7. Planificadores con prioridades dinámicas
 - 5.7.1. Planificador óptimo
 - 5.7.2. Test de planificabilidad
 - 5.7.3. Gestión de recursos
 - 5.7.4. Servicio aperiódico

6. Arquitecturas hardware para tiempo real

- 6.1. Introducción
- 6.2. Entrada/salida.

- 6.2.1. Control por supervisión del estado
- 6.2.2. Control por interrupción
- 6.3. Programación a bajo nivel
 - 6.3.1. Requerimientos del lenguaje
 - 6.3.2. Ada
 - 6.3.3. C, C++
 - 6.3.4. Planificación y drivers de periféricos
- 6.4. Sincronización por interrupciones
 - 6.4.1. Polling e interrupciones vectorizadas
 - 6.4.2. Controladores de interrupción
- 6.5. Transferencia de datos. DMA
 - 6.5.1. Sincronización en la utilización del BUS
 - 6.5.2. Retardos introducidos por el DMA
- 6.6. Servicios de tiempo
 - 6.6.1. Timers
 - 6.6.2. Circuitos de Watch-dog
- 6.7. Buses
 - 6.7.1. Ejemplo. Bus VME
- 6.8. Redes
 - 6.8.1. Ejemplo. Red CAN