

Máster en Computación

TESIS DE MÁSTER

INSTRUMENTACIÓN DE CÓDIGO
PARA EL CÁLCULO DE
TIEMPOS DE EJECUCIÓN Y RESPUESTA
EN SISTEMAS DE TIEMPO REAL

Autor:

Álvaro García Cuesta

Director:

J. Javier Gutiérrez García

Grupo de Computadores y Tiempo Real



Indice

- **Introducción**
- Objetivos
- RapiTime
- Instrumentación a nivel de Aplicación y Sistema Operativo
- Casos de uso
- Integración en MAST
- Conclusiones y líneas futuras

Conceptos

- **Sistema de Tiempo Real**
 - resultado correcto
 - **cumplir requisitos temporales**
- **Análisis de planificabilidad**
 - cálculo de tiempo de respuesta de peor caso
 - usan valores del *WCET* (**tiempo de ejecución de peor caso**) de los elementos software
- **Instrumentación**
 - determinación de puntos del código para medida de tiempos

Métodos de cálculo de WCETs

- **basados en análisis estático**
 - identifican los **posibles caminos a través del código**
 - necesitan un **modelo preciso del hardware**
- **basados en medidas**
 - utilizan **medidas** producidas al **ejecutar el programa** en el **hardware real**

Arquitectura del Software de Tiempo Real

- **Consideramos dos capas diferenciadas:**
 - **Aplicación o Aplicaciones:**
 - lenguajes de programación
 - código disponible para el programador
 - **Sistema Operativo:**
 - estructura más compleja
 - código de más difícil acceso, **no siempre disponible**

Indice

- Introducción
- **Objetivos**
- RapiTime
- Instrumentación a nivel de Aplicación y Sistema Operativo
- Casos de uso
- Integración en MAST
- Conclusiones y líneas futuras

Objetivo

- Integrar una herramienta de cálculo de tiempos en una plataforma de tiempo real, desarrollar metodologías de aplicación y uso y estudiar la integración con herramientas de análisis de planificabilidad

Herramientas y plataforma de tiempo real

- ***MAST*** (análisis de planificabilidad)
 - **código abierto disponible**
 - **conocimiento** sobre su funcionalidad
- ***RapiTime*** (cálculo de WCETs)
 - herramienta propietaria disponible
 - **ausencia de alternativas** de código abierto fiables
- ***MaRTE OS*** (S.O. de tiempo real)
 - **libre** distribución
 - fácil acceso a su estructura y conocimiento profundo

Objetivos específicos

- Estudiar las capacidades y funcionamiento de la herramienta *RapiTime*
- Desarrollar una metodología para la instrumentación de código de aplicación ADA y C.
 - Integrar *RapiTime* en *MaRTE OS*
- Desarrollar una metodología para la instrumentación del propio código fuente de *MaRTE OS*
- Estudiar la viabilidad de la integración de los datos generados con *RapiTime* en el conjunto de herramientas *MAST*

Indice

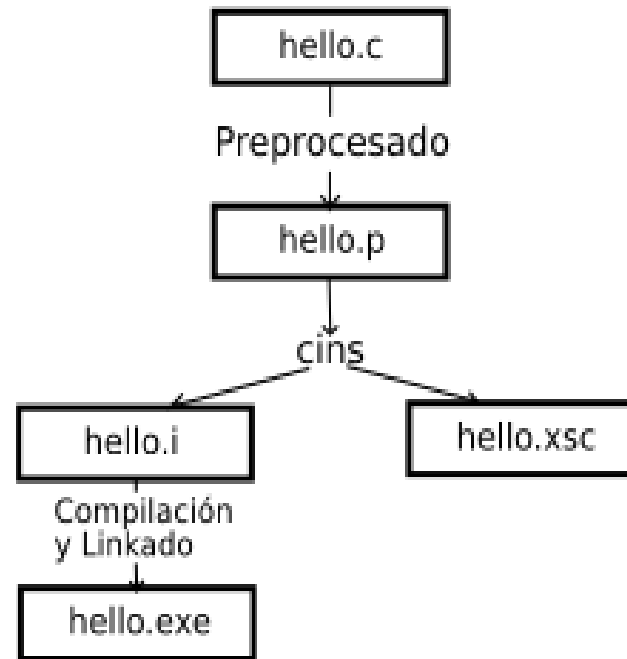
- Introducción
- Objetivos
- **RapiTime**
- Instrumentación a nivel de Aplicación y Sistema Operativo
- Casos de uso
- Integración en MAST
- Conclusiones y líneas futuras

RapiTime

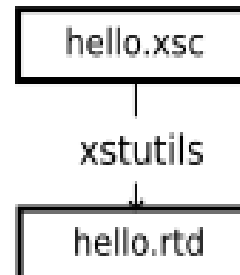
- Herramienta comercial de Rapita Systems
- Cálculo de *WCETs* a partir de:
 - representación en forma de **árbol**
 - **medidas exhaustivas** sobre el sistema real
- Resultados de interés:
 - *WCET* calculado
 - Máximo tiempo de ejecución medido
 - Mínimo tiempo de ejecución medido
 - Tiempo de ejecución medio medido

Proceso de medida con RapiTime (5 pasos)

1- Construcción



2- Análisis estructural

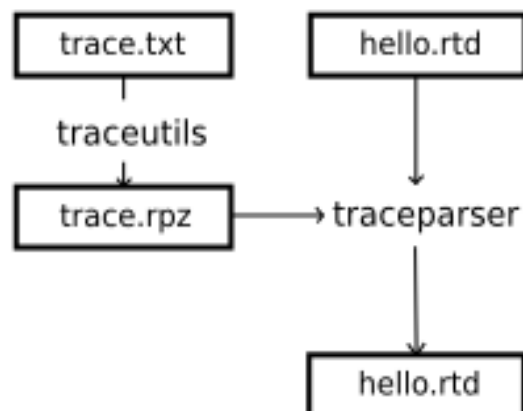


Proceso de medida con RapiTime (5 pasos)

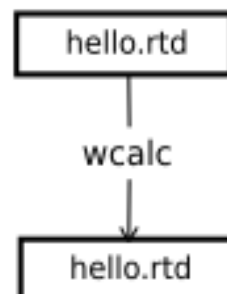
3- Testeo y generación de trazas



4- Procesado de las trazas



5- Cálculo de WCETs



Indice

- Introducción
- Objetivos
- RapiTime
- **Instrumentación a nivel de
Aplicación y Sistema Operativo**
- Casos de uso
- Integración en MAST
- Conclusiones y líneas futuras

Instrumentación a nivel de Aplicación

- **Implementación de filtros**
 - *scope.flt, marte.flt y little_endian.flt*
- **Implementación de librería de funciones (C y ADA):**
 - RPT_Ipoint
 - RPT_Context_Switch_To
 - RPT_Intrpt_Entry y RPT_Intrpt_Exit
 - RPT_Output_Trace

Instrumentación a nivel de Aplicación (cont)

– **RPT_Ipoint**

- relacionar **posición** con un **instante temporal**
- estructura: **ID + Timestamp**
- *TSC* (Time Stamp Counter), a partir del *Pentium*

– **RPT_Context_Switch_To**

- colocar en el punto donde se realiza el cambio de contexto

Instrumentación a nivel de Aplicación (cont)

– **RPT_Intrpt_Entry** y **RPT_Intrpt_Exit**

- al inicio y final del manejador
- señalar tiempo que no pertenece a la tarea

– **RPT_Output_Trace**

- enviar array de **trazas** para salvarlo en fichero

Instrumentación a nivel de S.O.

- Análisis del código para buscar los puntos de instrumentación
 - código complejo
 - limitación de la herramienta
- Adaptar código a las **reglas de estilo** de *MaRTE OS*
- Uso de **pragmas** (directivas de compilación) para especificar **áreas concretas a instrumentar**

Indice

- Introducción
- Objetivos
- RapiTime
- Instrumentación a nivel de Aplicación y Sistema Operativo
- **Casos de uso**
- Integración en MAST
- Conclusiones y líneas futuras

Plataforma de trabajo



Casos de Uso

- **Objetivos:**
 - evaluar código a **nivel de aplicación (C)**
 - evaluar código a **nivel de *S.O.* (ADA)**
- **Programa ejemplo seleccionado:**
 - **C**
 - **3 threads**
 - periodos: 1, 2 y 3 respectivamente
 - mismo cuerpo (lazo de sumas)
 - ***clock_nanosleep* para suspenderse**

Casos de Uso (cont)

- **Aplicación:** tiempos de **peor caso** y tiempos **máximos medidos**:

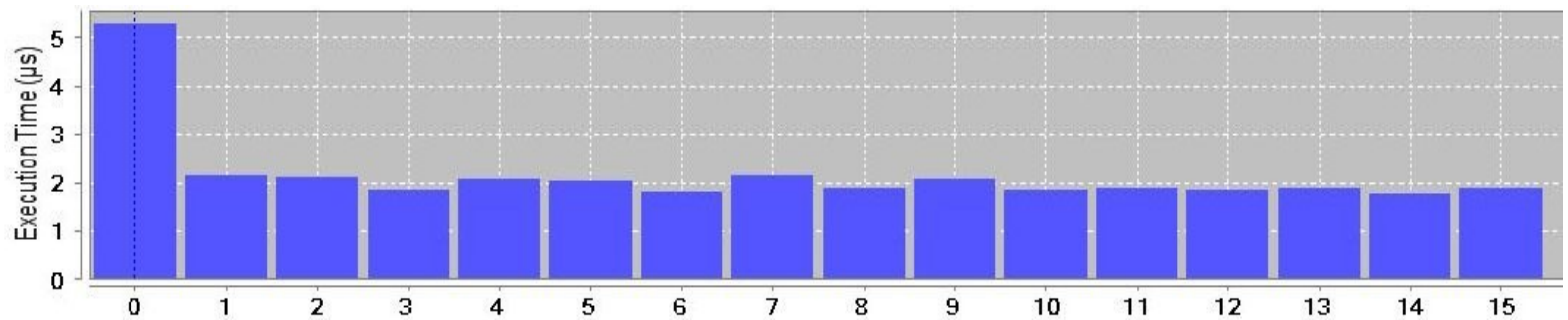
	<i>W-OverET</i> (μ s)	<i>Max-OverET</i> (μ s)
<i>Thread 1</i>	2,768	2,388
<i>Thread 2</i>	2,054	1,845
<i>Thread 3</i>	2,054	1,845

- Primera invocación del thread 1 dura más -> posiblemente **efecto de la *cache***
- **Sistema Operativo:** tiempos de peor caso y máximo, mínimo y medio medidos para **Clock_Nanosleep**:

<i>W-OverET</i> (μ s)	<i>Max-OverET</i> (μ s)	<i>Min-OverET</i> (μ s)	<i>A-OverET</i> (μ s)
5,280	5,280	1,755	2,142

Ejemplo de información gráfica

- Gráfica de tiempos por invocación:



- Gráfica *1-Probabilidad acumulada*:



Indice

- Introducción
- Objetivos
- RapiTime
- Instrumentación a nivel de Aplicación y Sistema Operativo
- Casos de uso
- **Integración en MAST**
- Conclusiones y líneas futuras

Integración en MAST

- Modelo *MAST*
 - parámetros temporales
- Identificación de los parámetros temporales en *MAST*
 - **Hecho** (operations, context switch)
 - **Pendiente** (timers, drivers)
 - **A estudiar** (ISR_switch)
- Tomar los datos de salida de *RapiTime* (SQLite) e integrarlos en *MAST* (ASCII, XML), usando lenguajes como PHP o C

Indice

- Introducción
- Objetivos
- RapiTime
- Instrumentación a nivel de Aplicación y Sistema Operativo
- Casos de uso
- Integración en MAST
- **Conclusiones y líneas futuras**

Conclusiones

- Estudio del **funcionamiento e integración** de *RapiTime* con plataforma de tiempo real *MaRTE OS* para **ADA y C**
- **Creación de una librería de instrumentación** adaptada al sistema usado
- Desarrollo de una **metodología para la instrumentación** de código a **nivel de aplicación**

Conclusiones (cont)

- Desarrollo de una **metodología para la instrumentación** de código a **nivel de sistema operativo**
- **Estudio de la integración** de las medidas de *WCET* con las herramientas de análisis *MAST*
- Se han apreciado **limitaciones** en la **herramienta de cálculo de *WCETs***:
 - unidad básica la función, no permitiéndonos obtener el *WCET* entre dos puntos cualesquiera del código

Líneas Futuras

- Completar las **medidas exhaustivas de todos los servicios del sistema operativo** *MaRTE OS*
- **Desarrollar las herramientas** que integren automáticamente estos resultados de tiempos de peor caso con *MAST*