

Time Logger: Una Herramienta Software de Soporte a la Docencia de Sistemas Operativos de Tiempo Real

Silvia Terrasa*, Patricia Balbastre*, Marián Roselló#, Ismael Ripoll*,
Alfons Crespo*.

*Dpt. de Informática de Sistemas y Computadores
Universitat Politècnica de Valencia
460220 Valencia
{sterrasa, patricia, iripoll, alfons}@disca.upv.es.

#ESA - ESTEC
European Space Agency
Noordwijk-The Netherlands
Maria.Rosello@esa.int

Resumen

En el presente artículo se describe una herramienta software de soporte a la docencia de asignaturas relacionadas con sistemas de tiempo real. Con ella se pretende que el alumno sea capaz por sí mismo de encontrar las diferencias entre los sistemas operativos de propósito general y los que están diseñados para soportar tareas o procesos de tiempo real. El manejo de esta herramienta es sencillo y flexible y además, proporciona un entorno visual para la representación de resultados.

1.- Introducción

En la enseñanza de sistemas de tiempo real, una de las ideas que el alumno debe tener clara cuando acaba de cursar la asignatura es que, aunque se utilice un lenguaje de programación tiempo real como por ejemplo Ada'95, si el sistema operativo que hay por debajo no es de tiempo real el sistema global no se puede considerar de tiempo real. Esta idea, aunque a simple vista pueda parecer trivial, sin la posibilidad de comprobarlo en un sistema operativo real queda bastante abstracta, y el problema que se nos plantea, es que los alumnos pueden llegar a no entender las implicaciones que tiene la elección del sistema operativo en el desarrollo de sistemas de este tipo. Con la intención de solucionar este problema, se ha introducido la posibilidad de que el alumno utilice la herramienta que presentamos en este artículo.

La herramienta en cuestión, es una simple librería de funciones escrita en ANSI C que ofrece la posibilidad de medir tiempos mediante la generación de eventos y que permite trabajar tanto en un sistema operativo de propósito general (Linux), como en un sistema operativo de tiempo real (RT-Linux). Una vez generados todos los eventos de medición la información se vuelca en un fichero que es interpretado por una aplicación gráfica que muestra los resultados en cronogramas.

El resto del artículo está esquematizado de la siguiente manera, en la sección 2 se verá la motivación y el planteamiento del problema, en la sección 3 la herramienta software, también se describirá someramente el sistema operativo de tiempo real RT-Linux, imprescindible para entender el funcionamiento de la misma. Finalmente en la sección 4 concluiremos y presentaremos una serie de trabajos futuros.

2.- Planteamiento del Problema.

Para entender la problemática que surge al ver la importancia de los sistemas operativos en el desarrollo de sistemas de tiempo real sólo hay que ver en que titulaciones se imparte la asignatura. En particular, la asignatura de Sistemas de Tiempo Real (*STR*) se imparte en cuatro titulaciones distintas: Ingeniero Industrial, Ingeniero en Automática i Electrónica Industrial, Ingeniero en Informática, Ingeniero Técnico en Informática (en las dos especialidades). En todas

ellas se imparte como asignatura optativa de último curso excepto en el caso de Ingeniero en Automática i Electrónica que es una asignatura obligatoria [8].

Lógicamente, se supone que para alumnos que cursen las titulaciones de ingeniero (técnico o superior) en informática, el concepto de sistema operativo debe de estar claro cuando llegan al último curso, sin embargo para ingenieros industriales el tema no está tan claro. Por tanto si ni siquiera comprenden bien las tareas que desempeña un sistema operativo, difícilmente podrán ver las diferencias existentes entre los sistemas operativos de tiempo real y los sistemas operativos de propósito general [6].

Debido a la diversidad de las titulaciones, en un principio se decidió incorporar el uso de la herramienta sólo en aquellas que no tenían nociones sobre los sistemas operativos, pero al final se incorporó a todas las titulaciones, ya que a veces la redundancia de conocimientos es beneficiosa y permite que cosas que no se acaban de entender en un principio, queden completamente claras cuando se repasan una segunda vez.

Una vez vista la motivación pasamos a describir la herramienta en sí.

3.- Herramienta Software.

La herramienta de medición de tiempos que se presenta en este artículo surgió como trabajo de investigación y en principio para atender a una necesidad de medir tiempos tanto en aplicaciones de propósito general (sobre Linux), como en tareas de tiempo real (ejecutadas sobre RT-Linux). A pesar de que en un principio sólo se pensó para utilizarla en cuestiones relacionadas con la investigación, pronto se adoptó como herramienta de soporte para las asignaturas de sistemas de tiempo real, ya que precisamente ese es el ámbito del trabajo de investigación dónde se enmarca la herramienta.

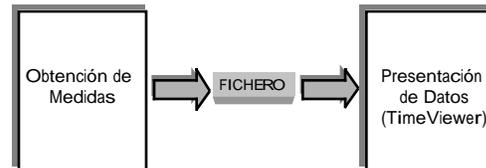


Figura ¡Error!Argumento de modificador desconocido.
.- Diagrama de bloques

Cómo se puede ver en la figura 1, la herramienta consta de dos módulos: un módulo de medición y otro de visualización de resultados, ambos implementados en el lenguaje de programación C.

Antes de continuar con la descripción de la herramienta se va a describir algunos detalles del sistema operativo de tiempo real RT-Linux, ya que el conocimiento de algunas de sus características es necesario para poder entender el funcionamiento global de la herramienta.

3.1.- Sistema operativo RT-Linux

RT-Linux [1] es un sistema operativo de tiempo real estricto. Dispone de un planificador expulsivo, basado en prioridades fijas, y su idea de funcionamiento (figura 2) es la de crear una capa entre el hardware y el sistema operativo Linux de manera que desde esta capa se capturan todas las interrupciones y se accede a los puertos (dispositivos hardware). Con esto Linux pasa a ser una tarea más del sistema operativo de tiempo real, concretamente pasa a ser la tarea menos prioritaria del sistema.

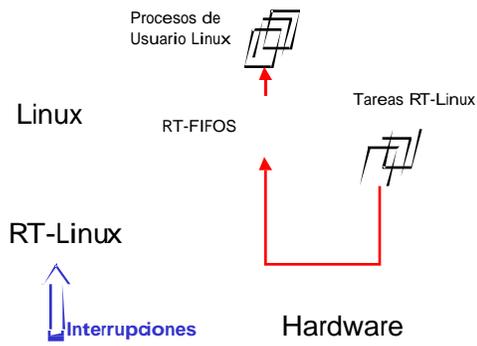


Figura 3. Esquema básico de RT-Linux

Mediante este esquema es posible que tareas de tiempo real convivan con procesos de usuario corrientes, los cuales se ejecutan desde Linux. La comunicación entre los dos niveles es posible a través de unas "tuberías" llamadas *rt_fifo*s, las cuales se gestionan como cualquier dispositivo en Linux.

Las ventajas que aporta este sistema operativo frente a otros existentes, son por una parte que se trata de código libre que el alumno puede obtener de la web [3], además su instalación es muy sencilla, sólo hay que aplicar un parche al núcleo de Linux y compilarlo, una vez realizadas estas tareas, para activar su funcionamiento sólo tenemos que cargar dinámicamente el módulo del planificador de tiempo real. Finalmente para programar las tareas de tiempo real se sigue el esquema de la programación de módulos Linux, con esto sólo hay que cargar su módulo para ponerlas en funcionamiento.

Una vez aclarados los conceptos básicos del sistema operativo RT-linux, pasamos a describir cada uno de los módulos de la herramienta

3.2.- Módulo de mediciones.-

El módulo de mediciones es una biblioteca que ofrece una serie de funciones (figura 3) para:

- Iniciar la sesión de medida
- Crear eventos
- Crear estados

- Generar Eventos,
- Volcar información a un fichero.

Esta biblioteca dispone de dos interfaces, una para poder medir tiempos en los procesos de usuario y otra para medir tiempos en tareas de RT-Linux. La figura 3 muestra la interfaz para los programas de usuario.

La forma de utilizar este módulo es sencilla, primero se inicia la sesión y se crean los eventos que se van a utilizar, y si se desea, se pueden definir estados (que no son más que los periodos entre dos determinados eventos), finalmente, se colocan en el código la generación de los eventos. Por ejemplo, si queremos medir el tiempo que tarda una operación de E/S sólo tenemos que generar un evento (*start*) justo antes de la operación de E/S y otro evento distinto (*finish*) inmediatamente después. El intervalo de tiempo existente entre los eventos *start* y *finish* será el tiempo que se ha estado haciendo la operación de E/S¹. Una vez acabada la sesión de medición se realiza el volcado de la información a un fichero con determinado formato. Este fichero será la entrada del siguiente módulo de la Herramienta (TimeViewer).

```
int InitSession(char *file);
int CreateEvent(int id, char *name, int pid);
int CreateState(int event1, int event2,
                char *state, int pid);
int GenerateEvent(int event, int pid);
int FlushBuffer(void);
```

Figura 4. Funciones del módulo de medición

Respecto a la interfaz de la biblioteca para tareas de RT-Linux, el planteamiento es el mismo, es decir, las funcionalidades son las mismas pero adaptadas al entorno. Como se puede ver en la figura 4, ahora la biblioteca está a dos niveles, por una parte hay un módulo RT-Linux que ofrece las funciones a las tareas de tiempo real, pero debido a que en RT-Linux no existe sistema de ficheros la información de los eventos debe "subirse" a la capa de Linux para poder guardarla en disco.

¹ Se supone que la sobrecarga de la generación de los eventos es despreciable, ya que se utiliza el registro TSC del procesador para capturar la hora del sistema.

Esquema de Funcionamiento para RT-Linux

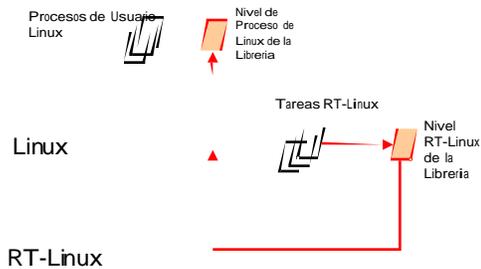


Figura ;Error!Argumento de modificador desconocido.

tarea a nivel de Linux y que pasa cuando lo hacen

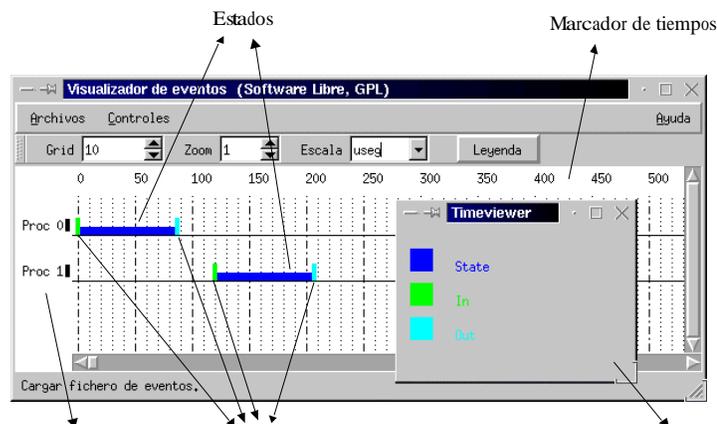


Figura ;Error!Argumento de modificador desconocido. **Modulo TimeViewer**

3.3.- Módulo TimeViewer.

Una vez generado el fichero del módulo de mediciones, éste ya puede ser leído por el módulo TimeViewer que representará la información (eventos y estados) en forma de cronograma.

Como se puede ver en la figura 5 este módulo a parte de visualizar información de la traza del medidor permite realizar otras operaciones como:

- Cambiar de escala,
- Hacer zoom,
- Etiquetar eventos y estados,
- Etc.

Con este módulo los alumnos pueden ver exactamente que pasa cuando implementan una

a nivel de RT-Linux, pudiendo observar la sobrecarga del operativo en el caso de Linux (paginación, acceso a dispositivos, sobrecarga del código del núcleo, etc.)

El entorno gráfico se ha diseñado gracias a las bibliotecas gráficas QT [7], las cuales pueden generar entornos de ventanas para Linux pero tambien existe la posibilidad de adquirir un compilador para entornos Windows'9x

Para finalizar la sección se plantea un ejemplo de uso simple de la herramienta.

3.4.- Ejemplo de uso de la Herramienta.

El ejemplo presenta un proceso padre que crea un total de 10 hijos) Cada uno de ellos tiene un bucle donde realizan una operación de E/S (sacar por pantalla la frase “Hello Word”).

A parte de los eventos (*Start* y *Final*) se crea un estado (“*Entrada/Salida*”) que determinará el periodo entre cada par de eventos (*start*, *final*). Finalmente cuando el proceso padre detecta que todos sus hijos han finalizado el mismo realiza el volcado de la información.

```

/* Programa de prueba */
#include "my_lib.h"
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>

#define MAX_FILL 10

int main()
{
    long ranf;
    int i, pid=MAX_FILL;

    /*Inicio de la sesion de medida */
    InitSession("pasima21\0");

    /*Generación de los procesos */
    for(i=0; i<MAX_FILL; i++)
    {
        if(fork()==0)/*proceso hijo */
        {
            pid =i;
            break;
        }
        else /* proceso padre */
            continue;
    }
    if (pid==MAX_FILL)/* proceso padre */
    {
        /* Antes de hacer un FlushBuffer()
        espera a que mueran todos sus hijos */
        for(i=0;i<MAX_FILL;i++) wait();
        FlushBuffer();
        exit(0);
    }
    else /*Proceso hijo*/
    {
        /* Creación de los eventos */
        CreateEvent(1, "Start\0", pid);
        CreateEvent(2, "Final\0",pid);

        /* Creación del estado */

```

```

CreateState(1,2,"Entrada/Salida\0",pid)
;
    for(i=0; i<5; i++)
    {
        /* Medimos el tiempo que cuesta
        hacer una operación de E/S */
        GenerateEvent(1,pid);
        printf("Hello World\n");
        GenerateEvent(2,pid);
        /* Entre medida y medida
        esperamos 3 segundos */
        sleep(3);
    }
    exit(0) ;
}
}

```

Figura ¡Error!Argumento de modificador desconocido.
Código ejemplo

4. Conclusiones.

En el presente plantea la problemática con la que se encuentran los profesores de las asignaturas de sistemas de tiempo real a la hora de explicar conceptos sobre sistemas operativos de tiempo real a alumnos sin conocimientos previos sobre sistemas operativos.

Esta problemática se ve en parte resuelta gracias a la herramienta software que se ha descrito. Mediante esta herramienta un alumno puede ver las diferencias existentes en la ejecución de una tarea en un sistema operativo de propósito general (Linux) y la respuesta de esa misma tarea ejecutada por un sistema operativo de tiempo real (RT-Linux). Esta comparación le ofrece una prueba válida de que los conceptos que se han desarrollado en clase sobre la conveniencia de utilizar sistemas operativos de tiempo real como soporte a los sistemas de tiempo real.

Por último señalar que como trabajo futuro se está planteando ampliar las funcionalidades de la herramienta para que sea mas robusta y completa.

Referencias

- [1] M. Barabanov. “A Linux-base Real Time

- Operating System*” PTh. 1997.
- [2] Real Time Linux Homepage :
<http://www.rtlinux.org>
 - [3] Tutorial RT-Linux de Ismael Ripoll en castellano:
<http://bernia.disca.upv.es/~iripoll/rt-linux/rtlinux-tutorial>
 - [4] “*Kernel HOWTO, The linux Kernel HOWTO*”
<http://www.linux.org/docs/ldp/howto/Kernel-HOWTO.html>
 - [5] Linux Kernel Homepage
<ftp://ftp.kernel.org/pub/>
 - [6] A. Crespo, J. Vila, F. Blanes, I. Ripoll. “Real-Time Education in a Control Engineering Curriculum”. *Real-Time Systems Education*. IEEE Computer Society Press.
 - [7] Qt on-line reference documentation
<http://doc.trolltech.com/>
 - [8] S. Terrasa, P.Balbastre, A. Crespo. “La importancia del uso de procesos físicos reales en la enseñanza universitaria en la ingeniería”. JENUI’2000, pp.304-310