ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA

Principios de Manejo de Señales con Herramientas Open Source

Grace Maricela Bermeo Quezada (1), Carlos Eduardo Velásquez Rubio (2), María Antonieta Álvarez (3)
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador
gbermeo@espol.edu.ec (1), cevelasq@espol.edu.ec (2)
Escuela Superior Politécnica del Litoral (ESPOL) (3), Ingeniera en Electrónica y Telecomunicaciones (3),
aalvare@fiec.espol.edu.ec (3)

Resumen

Este proyecto introduce la herramienta de código abierto para la adaptación de sistemas operativos comunes hacia sistemas operativos de tiempo real, RTAI; la herramienta para el manejo de matrices, Scilab y su módulo para la creación de esquemáticos de control, Scicos. En el desarrollo del mismo se detalla el proceso de instalación de las herramientas RTAI, Scilab, Comedi, la cual es la librería de controladores para tarjetas de adquisición de datos, en este caso la tarjeta utilizada es la PCI-6024E de National Instruments, y la adaptación del núcleo de Linux para poder obtener retardos mínimos en los tiempo de atención de llamadas de interrupciones. Las instalaciones de estas herramientas fueron realizadas en los sistemas operativos: Ubuntu, Fedora y Centos, con el fin de determinar cuál de estos sistemas operativos es el más eficiente para el manejo de tareas de tiempo real. También se muestra el proceso de pruebas de control de una planta de control de nivel de fluido y el sistema de visualización y control de señales de forma remota con las herramientas JRtaiLab y RTAI-XML.

Palabras Clave: Tiempo real, núcleo.

Abstract

This project introduces the open source tool for the adaptation of the common operating systems to real time operating systems, RTAI; the tool for the management of matrixes, Scilab, and its module for the creation of schematics of control, Scicos. During the development of this project, it is shown the process of installation of RTAI, Scilab, Comedi, which is the library of the drivers for the data acquisition cards, in this case the card used is PCI-6024E of National Instruments, and the adaptation of the kernel of Linux for obtain the smallest delays for calls of interruptions. The installations of these tools were made on the operating systems: Ubuntu, Fedora and Centos to determine which of these operating systems is more efficient for handling real time task. It also shows the process of testing the control of a fluid level control plant, and the system of remote visualization and control of signals with the tools JRtaiLab and RTAI-XML.

Keywords: Real Time, kernel.

1. Introducción

El predominio de soluciones propietario para sistemas de control de maquinarias y de plantas industriales, significan una voluminosa inversión para su implementación y utilización; por lo tanto, este proyecto constituye un inicio del proceso de migración de aplicaciones licenciadas hacia herramientas Open Source, es decir de código libre; también está enfocado en reducir la complejidad que involucra la instalación, configuración y manejo de soluciones Open Source vinculadas al proyecto.

El proyecto tiene como objetivo examinar la eficiencia de la herramienta de código libre RTAI (Interfaz de Aplicaciones de Tiempo Real), diseñado para Debian para el control de aplicaciones de tiempo real, probar la viabilidad de su instalación en los sistemas operativos Ubuntu, Fedora y Centos.

Luego de realizado el proceso de instalación en cada uno de los sistemas operativos, se calificará el desempeño de las herramientas en cada sistema operativo cuantificando la capacidad de respuesta gráfica de los sistemas operativos durante la ejecución de las aplicaciones de tiempo real, los cambios del tiempo de estabilización de la planta y los tiempos de respuesta de los sistemas operativos adaptados mediante las herramientas de medición provistas por la herramienta RTAI. También se comprobará las capacidades y limites de funcionamiento del sistema de control remoto de aplicaciones de tiempo real, por criterio de máximo rango de frecuencia de trabajo, el cual está conformado por el servidor RTAI-XML y la herramienta cliente JRtaiLab.

2. Herramientas utilizadas para la instalación de RTAI

Definición de Sistema de Tiempo Real.

Un sistema de tiempo real (STR) es definido como todo sistema capaz de garantizar que los procesos o tareas que se encuentran bajo su control sean ejecutados o atendidos dentro de intervalos mínimos de tiempo relativamente invariables [1].

Todos los sistemas tienden a tener baja latencia, pero un sistema de tiempo real requiere que la latencia y el jitter sean tiempos predeterminados y constantes sin importar la cantidad de carga en el procesamiento. La cualidad de tiempo real se la puede clasificar de tres formas [2]:

Tiempo Real Estricto: Todas las acciones deben ocurrir dentro de un plazo especificado (Hard Time).

Tiempo Real Flexible: Se pueden perder plazos de vez en cuando, el valor de la respuesta decrece con el tiempo.

Tiempo Real Firme: Se pueden perder plazos ocasionalmente, el valor de una respuesta tardía no tiene valor.

Adaptación de un Sistema para Funcionamiento en Tiempo Real.

El método aplicado consiste en colocar una capa interfaz entre el hardware y el kernel estándar. Esta capa controla la ejecución de las tareas de tiempo real y ejecuta el kernel estándar como una tarea en background, es decir, el kernel estándar sólo se ejecuta cuando no hay tareas de tiempo real pendientes. Otra de las finalidades de esta adaptación es disminuir la carga de procesamiento, mediante la reducción de la carga de algunos módulos del kernel que para fines de operación de un sistema de tiempo real son esencialmente innecesarios.

Fundamentos de Funcionamiento de Sistema en Tiempo Real

Algunas de las funcionalidades de un sistema operativo general son la gestión de procesos, gestión de memoria, interacción con el hardware, servidor de ficheros, servidor de comunicaciones. Todo sistema operativo de propósito general trabaja de tal forma que sus tareas manejan prioridades que pueden cambiar a lo largo del tiempo, y que pueden ser interrumpidas por algún proceso y ser postergadas hasta que lo establezca su nueva prioridad. Esto se debe a que estos sistemas operativos están diseñados primordialmente para atender de forma inmediata cualquiera petición del usuario

Por otra parte, un sistema de tiempo real tiene como funcionalidad principal el servicio de aplicaciones para una respuesta en un intervalo de tiempo predeterminado y se centra en atender de forma primordial dos tipos de interrupciones, las interrupciones del procesador y las interrupciones de los periféricos. Esta característica está orientada a garantizar que ninguna tarea diseñada para tiempo real, sea postergada por alguna interrupción producida por alguna otra tarea.

El funcionamiento básico de las herramientas utilizadas consiste en colocar una capa, la cual es el nuevo administrador central de las gestiones del sistema operativo. Esta capa, discrimina las tareas en: tareas comunes y en tareas de tiempo real. Esta categorización es realizada disminuyendo la prioridad del kernel y de todo proceso dependiente de él .Las tareas de tiempo real tienen la más alta prioridad y no pueden ser postergadas por tareas comunes. Comúnmente el manejo de interrupciones es trabajo del kernel, pero, ya que el kernel tiene una prioridad baja y las tareas de tiempo real están vinculadas con el tiempo y el hardware, las interrupciones también pasan a ser manejadas por esta nueva capa. Caso contrario las interrupciones solo podrían ser atendidas luego de la finalización de una tarea de tiempo real.

El parche colocado dentro del proceso de adaptación del sistema, permite crear aplicaciones que corran bajo órdenes de usuario, es decir dentro del espacio de usuario el cual es manejado por el kernel, y que a pesar de esto, se ejecuten como aplicaciones de tiempo real.

Interfaz de Aplicación de Tiempo Real-RTAI

Interfaz de Aplicación de Tiempo Real o RTAI, por sus siglas en ingles REAL TIME APPLICATION INTERFACE, es la implementación de un sistema operativo de tiempo real estricto para GNU/Linux de forma que añade un pequeño kernel de tiempo real bajo el kernel estándar de GNU/Linux y trata al kernel de éste como una tarea de menor prioridad.

RTAI proporciona una opción llamada LXRT para facilitar el desarrollo de aplicaciones de tiempo real en el espacio de memoria del usuario. Las tareas básicas de tiempo real son implementadas como módulos del kernel. RTAI se ocupa de manejar de forma inmediata las interrupciones de periféricos antes de que lo haga el kernel, luego de ejecutar las posibles acciones de tiempo real que hayan podido ser lanzadas por efecto de la interrupción, pasan a ser atendidas por el kernel como cualquier otro proceso dentro de un sistema general.

RTAI utiliza el método Micro-Kernel de modificación de kernel, Este kernel añadido en realidad es una interfaz entre el hardware y el kernel estándar, lo que se llama tradicionalmente HAL (capa de abstracción de hardware o Hardware Abstraction Layer). La Figura 2.1 muestra la arquitectura de un microkernel.

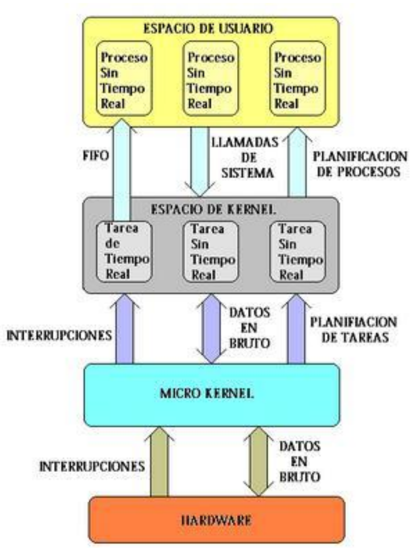

Figura 2.1 Arquitectura Micro kernel

Este micro-kernel capta las interrupciones de hardware y se asegura que las tareas de tiempo real se ejecuten con la mayor prioridad posible para minimizar la latencia. Este mismo sistema es utilizado por RTLinux, otro sistema de tiempo real.

RTAI está orientado a módulos. El uso de módulos dinámicos de kernel, permite escribir porciones de kernel como objetos separados que pueden ser cargados y suprimidos mientras el sistema este ejecutándose. Para usar RTAI es necesario cargar los módulos que implementan cualquier capacidad de RTAI que se pueda necesitar.

La Figura 2.2 muestra con mayor detalle la arquitectura básica de RTAI, que es muy similar a la de RTLinux.

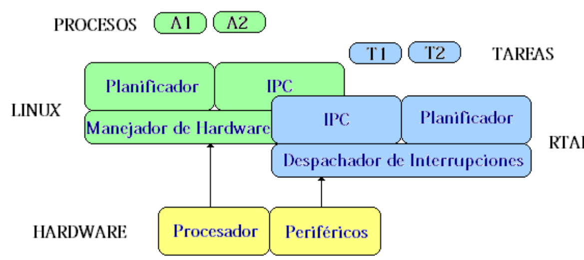

Figura 2.2 Arquitectura de RTAI

En la figura 2.3 se muestra una representación gráfica de las herramientas principales que han sido instaladas en cada sistema operativo.

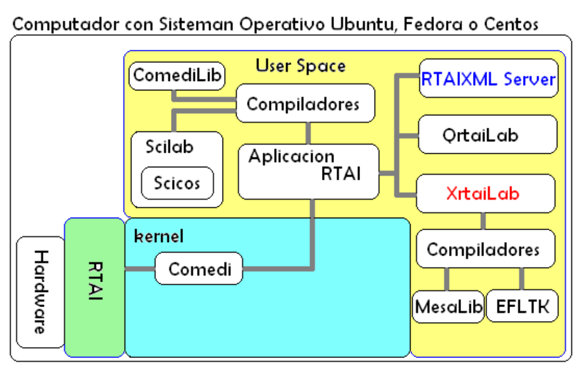

Figura 2.3. Esquema de Herramientas para RTAI.

Software Instalado en Ubuntu
Ubuntu

Antes de comenzar con el proceso de instalación de las herramientas principales, se debe instalar paquetes, es decir programas, librerías o códigos pequeños que son requeridos por los programas principales. En software libre estos paquetes son utilizados en la elaboración y ejecución de varios programas, es decir, pueden ser utilizados por varias aplicaciones sin que sea exclusivo de alguna de ellas, de esta forma se reduce espacio lógico y procesamiento. Por lo tanto los programas que son construidos en base a estos paquetes no podrán funcionar sin que estos estén instalados dentro del sistema operativo, por este motivo se los conoce como dependencias.

Las dependencias, pueden ser utilizadas durante el proceso de instalación, como lo es en este caso las dependencias de uso general, como también pueden ser para el uso de herramientas específicas como son las dependencias de RTAI.

Entre los tipos de dependencias que se requieren en el proceso de instalación están las dependencias generales [14], utilizadas por los proceso de configuración e instalación de herramientas, las cuales son: cvs, svn, build-essential, checkinstall.

También son necesarias dependencias para el proceso de levantamiento las cuales son: libncurses5-dev, kernel-package y fakeroot.

Luego se encuentran las dependencias exclusivas para cada herramienta las cuales se detallan a continuación.

Dependencias de RTAI
libxmu-dev, libxi-dev, doxygen, autoconf, automake, libtool
Dependencias de COMEDILIB
bison, flex, python-dev
Dependencias de COMEDI-CALIBRATE
libboost-program-options-dev, libgsl0-dev,
Dependencias de SCILAB-4.1.2
g77, gfortran, sablotron, tcl8.4, tk8.4, tcl8.5-dev, tk8.5-dev, xaw3dg-dev, libpvm3, libgtkhtml2-dev, libzvt-dev, libvte-dev
Dependencias de SCILAB-5
swig, pvm-dev, gettext,
Dependencias de MESALIB
X11proto-xext-dev, xlibs-static-dev, libxext-dev, libxt-dev, libglu1-mesa-dev
Dependencias de QRTAILAB
libqt4-dev, libqwt5-qt4-dev

Software Instalado en Fedora y Centos

Las dependencias instaladas en Ubuntu no son diferentes en funcionalidad respecto a las dependencias instaladas en Fedora y en Centos. Esencialmente los cambios que existen entre las dependencias de estos sistemas, en ciertos casos son exclusivamente solo cambios en los nombres, como también se da el caso en que un paquete deba ser reemplazado por dos o más paquetes o dos o más paquetes puedan ser reemplazados con un solo paquete.

Muchos de las dependencias que son instaladas individualmente en el proceso de instalación de las herramientas en Ubuntu, sin instalados como un grupo de paquetes cuando se realiza la instalación de Development Tools, en el proceso de instalación en Fedora y en Centos.

A continuación se muestra el listado de dependencias instaladas en Fedora y en Centos.

Dependencias GENERALES, KERNEL Y RTAI
cvs subversion Development Tools checkinstall intltool ncurses ncurses-devel libXi-devel libXmu-devel
Dependencias de COMEDILIB
python-devel python flex
Dependencias de COMEDICALIBRATE
gsl-devel boost
Dependencias de SCILAB-4.1.2
sablotron pvm compat-gcc-34-g77 tcl-devel tk-devel gtkhtml2-devel Xaw3d-devel
Dependencias de MESALIB
xorg-x11-proto-devel libXext-devel libXt-devel
Dependencias de QRTAILAB
qt-devel

3. Análisis del desempeño de RTAI para el manejo de señales eléctricas en las distribuciones: Ubuntu, Fedora y Centos.

Se realizan la comparación del desempeño de las herramientas por cada sistema operativo mediante pruebas utilizan los siguientes criterios.

Criterios Lógicos: Se ha identificado como criterios lógicos de análisis, a las herramientas de medición de velocidad de respuesta y procesamiento que son incluidas durante la instalación del paquete de RTAI. Estos conceptos comparativos son, la latencia que es el tiempo desde que se produce la interrupción hasta que se ejecuta la primera instrucción de la rutina de tratamiento, y el jitter que es la variación de la latencia.

Criterios Gráficos: Los criterios gráficos de evaluación del desempeño de las herramientas son cualidades con las que el usuario interactúa directamente durante el uso de una aplicación de tiempo real, ya que comprenden la definición en la gráfica de señales y la capacidad máxima de procesamiento gráfico que presenta cada sistema operativo durante el uso de las herramientas instaladas.

Criterios de Desempeño: Se ha definido como criterios de desempeño al tiempo en que le toma responder y estabilizarse a todo el sistema en conjunto, es decir la planta de control de nivel de fluidos y el computador que posee la tarjeta de adquisición de datos. Se ha realizado pruebas con cada uno de los sistemas operativos, utilizando una misma aplicación de tiempo real para cada prueba con un mismo tiempo de muestreo. A continuación se muestran las gráficas de las medidas cuyos resultados fueron los más concluyentes dentro del proceso de pruebas de criterios lógicos. Todos los valores están en el orden de los nanosegundos. En la figura 3.1 se muestra que Ubuntu y Fedora presentan los retardos máximos más cortos.

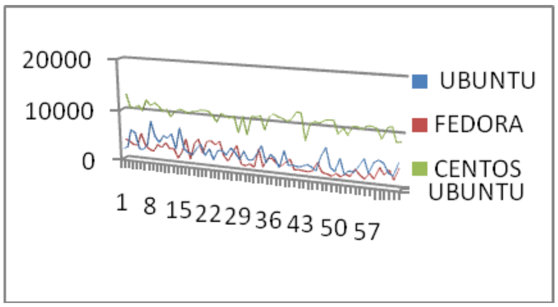

Figura 3.1. Latencia Máxima con herramienta latencia del kernel.

Gráficas cualitativamente semejantes se presentaron con los valores medidos con la herramienta de latencia de usuario. También se obtuvo un muestreo de latencias mínimas, en los cuales se puede observar que Ubuntu y Centos presentan las latencias mínimas más cortas.

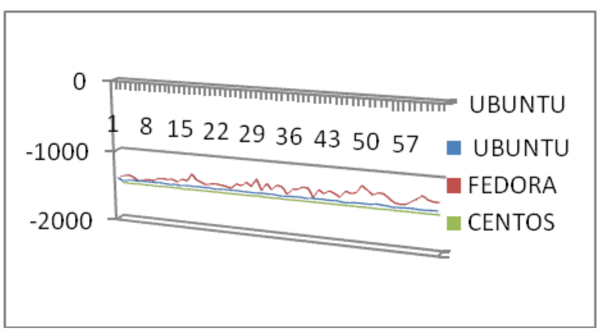

Figura 3.2. Latencia Mínima con herramienta latencia del kernel

Dado que Ubuntu converge en ambas estadísticas, se determina que Ubuntu es el sistema operativo más eficiente en términos lógicos.

A continuación se muestran las estadísticas de los resultados obtenidos para las pruebas de criterio gráfico que consistieron en generar una onda sinusoidal y secuencialmente se disminuyo el periodo de muestro, y se evaluó la calidad del manejo de las imágenes de los sistemas operativos. Se creó la siguiente lista de valores para poder categorizar la eficiencia frente a cada experimento. Excelente: 6, Muy bueno: 5, Bueno y buena con problemas de maximización de ventana: 4, Graficación Lenta o Distorsionada: 3, con retrasos o segmentación: 2, sin visualización: 1, se inhibe o se cierra xrtailab: 0.

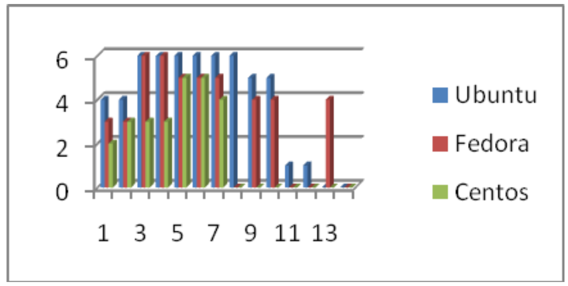

Figura 3.3. Desempeño Gráfico de los Sistemas Operativos.

Los resultados muestran que el sistema operativo Ubuntu presento un mejor desempeño frente a frecuencias más altas de trabajo, seguido por el sistema operativo Fedora.

Las pruebas de desempeño, han sido realizadas junto a la planta de control de fluidos y con una aplicación de tiempo real elaborado en base a un esquemático con el lazo de control de la planta. Estas pruebas muestran las pequeñas variaciones de tiempo que puede tener todo el sistema por efectos de uso de un sistema operativo específico. Para las pruebas, todos los valores y parámetros permanecieron. La figura 3.4 muestra los resultados de las pruebas para el criterio de desempeño.

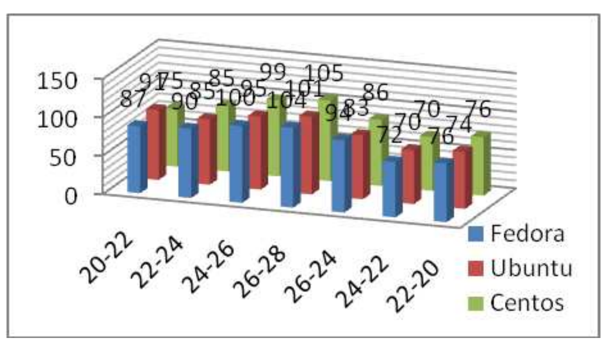

Figura 3.4. Desempeño de la Planta

Las pruebas consistieron en la alteración de la altura del fluido dentro de la planta y la lectura del tiempo que le toma estabilizarse. Por los resultados obtenidos se puede concluir que estadísticamente los tiempos de estabilización de Ubuntu son más cortos que los de los otros sistemas operativos

4. Visualización remota de Señales Eléctricas en Tiempo Real

RTAI-XML

RTAI-XML es un componente de tipo servidor del proyecto RTAI, implementa un servicio basado en diseño y desarrollo del control de aplicaciones en tiempo real. Su funcionamiento se basa en un puerto de red de un servidor en espera de llamadas entrantes donde el proceso de tiempo real o la tarjeta, están en ejecución o están listos para estarlo.

El sistema también cuenta con un cliente, el cual se comunica con el servidor por medio de TCP/IP, como se muestra en la figura 4.1

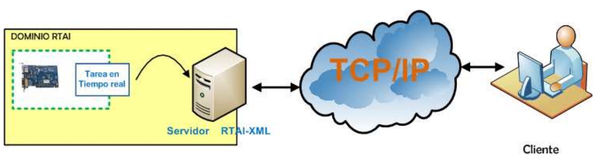

Figura 4.1. Estructura de RTAI-XML.

RPC –PROTOCOL

El RPC (Remote Procedure Call) Llamada a Procedimiento Remoto, es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos, es decir trabaja con un alto nivel de abstracción, ignorando los mecanismos de comunicación subyacentes y las características de su implementación.

XML-RPC

XML-RPC es un protocolo para la realización de llamadas a procedimiento remoto a través de TCP/IP, utiliza XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. La figura 4.4 nos ilustra de forma gráfica lo mencionado anteriormente.

En la figura 4.2 se puede apreciar la estructura dentro de la comunicación utilizada por RTAI-XML.

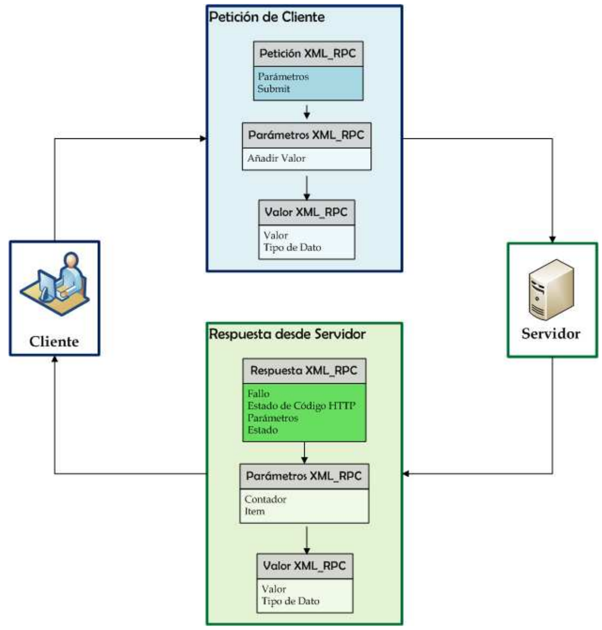

Figura 4.2. Procedimiento RPC entre Cliente y Servidor.

JRtaiLab

Es una pequeña aplicación escrita en lenguaje de programación java, también es conocida como applet; implementa un cliente genérico para ser atendido a través de internet mediante un servidor web que en este caso es RTAI-XML.

Luego de realizar la instalación del servidor RTAI-XML en el computador donde se encuentrar instalada la tarjeta de adquisición de datos, es necesario crear la aplicación de tiempo real, crear el script de ejecución de la aplicación para RTAI-XML, y levantar el servidor RTAI-XML.

La figura 4.3 muestra la imagen del script de configuración para las aplicaciones para RTAI-XML, entre los valores más importantes a configurar, están, el nombre de la aplicación a ejecutar, la carpeta donde se encuentra esta aplicación, y el tiempo que se desea que se ejecute la aplicación una vez que haya sido llamada por el cliente JRtaiLab.

```
#HOWTO in brief
#     >> The filename MUST be composed at maximum by 6 chars.
#Uncomment the following line if before executing this target you
# need to start another one defined by the variable depend.
#(This can be used recursively).
#Name of the  target.

#depend=

#[optional] Introduce a delay before target execution. Useful to execute depending targets.
#delay=0.0

#Name of the target executable
target=

#Absolute path
dir=

#[optional] Real time priority
priority=8

#Final time
time=10

#[optional] Starting option. If true the target starts in wait.
#wait=true

#[optional] Verbose output
verbose=true

#[optional] Log file
#log=true

#[optional] Name of the real time Scope
rts=RTS

#[optional] Name of the real time Digital (Led)
rte=RTE
```

Figura 4.3. Script de Ejecución para RTAI-XML

Cada aplicación de tiempo real que se desee ejecutar eventualmente, deberá tener su propio script de ejecución. Para determinar la eficiencia del servidor en cada sistema operativo se realizaron pruebas para la visualización y control de señales para el control de una planta de control de fluidos, sin alterar los parámetros entre cada sistema operativo.

La figura 4.4 muestra la ventana de visualización de señales del lado del cliente, cuando las señales del sistema están estabilizándose, y por lo tanto la altura del fluido de la planta, se estabiliza.

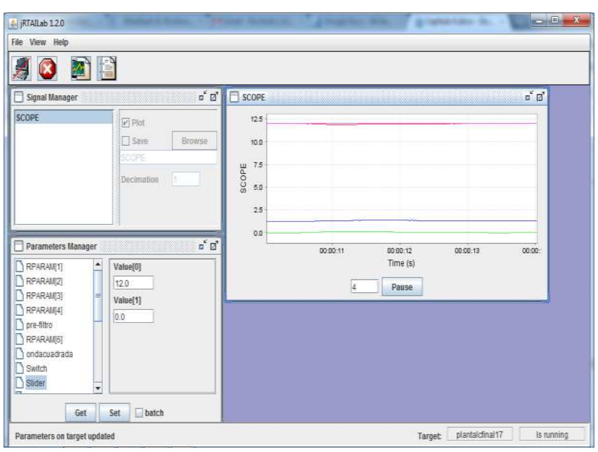

Figura 4.4 Visualización de una señal.

Se realizaron pruebas sobre la eficiencia de cada sistema operativo, y durante los experimentos Ubuntu, fue el único sistema operativo que permitió la alteración de los valores de amplitud de las señales para el control de la planta.

5. Conclusiones

- Se concluye que es viable la instalación de las herramientas en los sistemas operativos Ubuntu, Fedora y Centos y que dado que la herramienta RTAI trabaja a nivel de kernel, un error durante el proceso de una tarea podría dejar inoperable el computador en el que se trabaja.
- Se concluye que en Ubuntu y Fedora son más cortos que los máximos retardos que puede presentar Centos; también se obtuvo medidas con las que se demuestra que los retardos más cortos que se pueden conseguir, son los retardos en los sistemas operativos de Ubuntu y de Centos. Ubuntu es el sistema que estadísticamente puede presentar retrasos cortos en la atención de una llamada o interrupción que Fedora y Centos.
- Se concluye que Ubuntu es capaz de trabajar con frecuencias de muestreo superiores a las frecuencias de muestreo máximas con las que Fedora y Centos pueden trabajar las cuales son alrededor de 10 K hercios. Ubuntu muestra un buen desempeño gráfico hasta alcanzar periodos de muestreo alrededor de los 0.0005 segundos, mientras que los sistemas operativos Fedora y Centos se inhiben al trabajar con periodos de este orden. Fedora muestra eficiencias de trabajo muy cercanas a las de Ubuntu, Centos respondió de forma relativamente eficiente solo en el 50 por ciento de las pruebas, en las que el período mínimo de muestreo alcanzado es del orden de los 0.0001.
- Se concluye que el sistema operativo Ubuntu es más eficiente en cuanto a la visualización remota ya que en las pruebas realizadas con la planta, se ha podido visualizar eficientemente las gráficas de las señales en Jrtailab, estas señales son transmitidas desde los tres sistemas: Ubuntu, Fedora y Centos, pero solo el Sistema Ubuntu permite realizar cambios en los parámetros,ya que al tratar de realizar esto en los Sistemas Fedora y Centos, cambios controlados en el nivel del agua del tanque; el motor de planta se apaga y no responde ante ningún y los sistemas del cliente y del servidor se inhiben.

6. Recomendaciones

- Se recomienda verificar dentro de los archivos fuente de RTAI, que exista el correspondiente parche para la versión del kernel que se va a levantar y además verificar que exista el soporte de la tarjeta de adquisición de datos a utilizar dentro de la librería de Comedi.

- Se recomienda realizar todo el proceso de instalación sin alteración en su orden ya que en caso de ser alterado tanto las herramientas instaladas no cumplirán sus funciones correctamente
- Se recomienda la continuidad de las investigaciones en este proyecto para sentar las bases de migración a sistemas de código abierto en el área de control de sistemas industriales.

7. Referencias

[1] WIKIPEDIA, *Tiempo Real,* http://es.wikipedia.org/wiki/Tiempo_real. Última actualización: 18 Agosto 2010.

[2] LÓPEZ ZAMARRÓN D., *Análisis de Sistemas Operativos de Tiempo Real Libres,* Universidad Politécnica de Madrid, http://gayuba1.datsi.fi.upm.es/~dlopez/cache/doc/sotr.pdf, 2004.

[3] YARMOUK UNIVERSITY, *Requerimientos de Tareas de Tiempo Real*, http://faculty.yu.edu.jo/halzoubi/DownloadHandler.ashx?pg=24fc7f00-0bf6-43ed-8bb0-8faac491ae13&section=4dc59155-70a0-4123-b10d-4ef4f53839bf&file=L4.pdf.,2009.

[4] NATIONAL INSTRUMENTS, *PCI-6023E/6024E/6025E User Manual* , http://www.ni.com/pdf/manuals/322072a.pdf, October 1998

[5] LÓPEZ ZAMARRÓN D., *Diseño de un Rótulo Luminoso con fines Docentes,* Trabajo Fin de Carrera, Segovia, Universidad Politécnica de Madrid, Junio del 2005.

[6] BOVET, D. J. - CESATI M., *Understanding The Linux Kernel* (22-25), First Ed., San Diego: O'Reilly, 2000.

[7] WIKIPEDIA, *Núcleo Linux*, http://es.wikipedia.org/wiki/Núcleo_Linux. Última actualización: 20 de Diciembre del 2010

[8] GRUPO DE DESARROLLO DE MESA 3D, *The Mesa 3D Graphics Library*, http://www.mesa3d.org/, Octubre 2010

[9] EDE TEAM, *EQUINOX DESKTOP ENVIRONMENT*, http://equinox-project.org/, 2010

[10] DAVID SCHLEEF - FRANK MORI HEES - IAN ABBOTT, *Comedi*, http://www.comedi.org/doc/, 2009

[11] DAVID SCHLEEF - FRANK MORI HEES - IAN ABBOTT,*Comedi- Gerarquía de Dispositivos*, http://www.comedi.org/doc/index.html#COMEDIDEVICES, 2009

[12] WIKIPEDIA, *Scilab*, http://es.wikipedia.org/wiki/Scilab. Última Actualización: 6 de Diciembre del 2010

[13] THOMAS NETTER - INRIA, *Scicos: Block diagram modeler/simulator*, http://www-rocq.inria.fr/scicos/, Última Actualización: 2009

[14] HOLGER NAHRSTAEDT - ANDREAS VIKLUND, *QRTAILAB - Installation RTAI*, http://qrtailab.sourceforge.net/rtai_installation.html, 2008-2009

[15] MAASSE J., *Scicos as an alternative for Simulink, Migrating from Simulink to Scicos with respect to real time programs,* Eindhoven, Technische Universiteit Eindhoven, Mayo del 2006

[16] WIKILEARNING, *Compilación del kernel paso a paso,* http://www.wikilearning.com/tutorial/compilacion_del_kernel_paso_a_paso-el_proceso_de_configuracion/876-3, Consultado en Octubre 2010.

[17] BUCHER R., MANNORI S., NETTER T., *RTAI-Lab tutorial: Scilab, Comedi, and real-time control*, http://www.dti.supsi.ch/~bucher/pdf/RTAI-Lab-tutorial.pdf, Marzo del 2009

[18] LÓPEZ ZAMARRÓN D., *Análisis de Sistemas Operativos de Tiempo Real Libres,* http://gayuba1.datsi.fi.upm.es/~dlopez/cache/doc/sotr.pdf, 2004.

[19] PÁGINA DEL PORTAL DE INGENIERÍA QUÍMICA, *Tutorial de Scilab,* http://www.ingenieriaquimica.org/system/files/TutorialDeScilab.doc, 2010

[20] HOLGER NAHRSTAEDT - ANDREAS VIKLUND, *QRTAILAB - Performance*, http://qrtailab.sourceforge.net/performance.html, 2008-2009

[21] GRUPO RTAI, *RTAI-XML*, http://www.rtaixml.net/, 2010

[22] BASSO M., VASSALI M, RTAI-*XML: A Web Services Approach to Real-Time Control Systems,* Universit_a degli Studi di Firenze, Julio del 2009.

[23] TEXTOS CIENTÍFICOS, *RPC Llamada a procedimiento remoto*, http://www.textoscientificos.com/redes/tcp-ip/servicios-capa-transporte/rpc, Consultado en Octubre 2010.