

Acondicionamiento del Kernel de Linux para ejecución de aplicaciones con altas exigencias de respuesta temporal - Análisis de desempeño de Tiempo Real

Diego E. Quimbert Montes
dquimbert@gmail.com

Jorge A. Nava Amador
jorgeantonio.navaamador@gmail.com

ABSTRACT

El poco conocimiento de los sistemas en tiempo real en Bolivia, y el constante avance de trabajos de investigación en este campo, particularmente en el caso de sistemas de tiempo real estricto o robusto, obligan a abordar un aspecto que es necesario para los ingenieros electrónicos que se especializan en la Ingeniería de Control, especialmente de aquellos que trabajan en el desarrollo de sistemas con altas exigencias de respuesta temporal para la supervisión y control de procesos.

Los ingenieros de control en Bolivia, en su mayoría, desarrollan sistemas que solo consideran requisitos funcionales y dejan de lado los requisitos temporales, esto debido a la poca orientación que existe en este campo.

En este artículo se describe de manera sistematizada y clara, como se debe acondicionar un sistema operativo Open Source, como lo es la distribución Ubuntu de Linux, para poder desarrollar o manejar aplicaciones que trabajan en tiempo real o que exigen plataformas de tiempo real.

Keywords – *Ubuntu, Linux, Open Source, Real Time, Sistemas de Control en Tiempo Real, Exigencias Temporales.*

I. INTRODUCCIÓN

El diseño de controladores para un proceso o sistema que hoy en día se manejan tienen exigencias temporales y funcionales que cumplir estrictamente, es por ello que obligan a la ingeniería a emplear mecanismos de procesamiento de datos, adquisición de señales y/o envío de señales en altas velocidades o a velocidades que no deben sobrepasar un tiempo fijado, requiriendo así dispo-

ner de plataformas que operen en Tiempo Real, ya sea en computadores personales o estaciones de trabajo y/o también en sistemas embebidos como los Arduino, Intel, Raspberry Pi, y otros, empleando técnicas de Tiempo Real estricto o no estricto, incluso en sistemas de Hardware Reconfigurable como los FPGA's.

El constante avance de trabajos de investigación en este campo, particularmente en el caso de sistemas de tiempo real estricto, obligan a abordar un aspecto que es necesario para los ingenieros electrónicos que se especializan en la Ingeniería de Control, especialmente de aquellos que trabajan en el desarrollo de sistemas con altas exigencias de respuesta temporal para la supervisión y control de procesos.

Los ingenieros de control desarrollan sistemas que solo consideran requisitos funcionales y dejan de lado los requisitos temporales, esto debido a la poca orientación que existe en este campo.

Esta es la motivación que ha impulsado a que la carrera de Ingeniería Electrónica incorpore dentro de sus líneas de investigación, la de “Desempeño de Sistemas de Tiempo Real”, aprobada por el Honorable Consejo de Carrera durante el primer semestre de la gestión 2014.

Esta Línea de Investigación tiene el propósito de impulsar el trabajo en Sistemas de Tiempo Real, para complementar la formación del pregrado, así como vincularnos con universidades extranjeras. Entre sus objetivos se tienen:

- Formular una metodología para el modelado y análisis de los requisitos, configuración y despliegue de sistemas distribuidos heterogéneos que permitan la gestión de estos sistemas.
- Desarrollar una metodología de modelado y análisis que basada en la descripción del sistema (plataforma, aplicaciones, carga de trabajo y requisitos temporales) permita garantizar el cumplimiento de requisitos combinados de tiempo real estricto, de distribuciones especificadas de las latencias en la ejecución de transacciones y métricas de productividad y rendimiento.

- Desarrollo de técnicas de extracción de los modelos de comportamiento en base a la instrumentación estándar disponible en la plataforma y de los modelos de despliegue existentes. Se consideran sistemas no críticos (firm real-time) en los que la observación de la ejecución del sistema sobre el entorno real proporciona información estadística significativa para la configuración del sistema.
- Desarrollo de técnicas de verificación del comportamiento del sistema y del cumplimiento de sus requisitos en base a la monitorización de su ejecución sobre el entorno real, de su modelo de despliegue y comportamiento.

Como producto de los primeros pasos que se dieron, es que se pueden apreciar los resultados y planteamientos que se exponen en este artículo, donde presentamos los aspectos necesarios para regenerar el Kernel de Linux con funcionalidades que permiten incorporar planificadores o ejecutivos de tiempo real, ejecutando políticas que son conocidas en el diseño y análisis de Real Time Systems.

II. DETALLES DEL SISTEMA PARA EL ACONDICIONAMIENTO

Se espera de los sistemas operativos en tiempo real futuros, garanticen los plazos y las relaciones de precedencia entre las tareas que se sostienen bajo observación de las medidas de tolerancia a fallos en la base de una planificación integrada de los recursos[1].

Los plazos comunes de varias tareas cooperantes han de cumplirse en los sistemas distribuidos que tienen la sobrecarga de la transmisión en cuenta.

Frecuentemente las medidas temporales de control, se deben tomar durante la ejecución del programa para garantizar el comportamiento del sistema a tiempo o para iniciar la degradación agraciada de rendimiento. La llegada de tareas listas para su ejecución y los recursos que soliciten ya no puede ser considerada como un proceso aleatorio.

En áreas de la previsibilidad, un procedimiento más determinista debe ser aplicado, que utilicen la información sobre las futuras tareas, instantes cuando se entra en el estado de lista que está disponible en los sistemas en tiempo real. De este modo, futuros conflictos de recursos pueden ser detectados y posiblemente se resuelvan en una etapa muy temprana. Se espera de un sistema operativo en tiempo real, que puede predecir en cualquier punto en el tiempo, si todas las tareas activas se reunirán con sus plazos.

Éste artículo está basado en Ubuntu 14.04 LTS (Long Term Support) con el kernel por defecto 3.13.0, en el cual se instalará el Kernel 3.4 con el parche en Tiempo Real 3.4 rt.

III. METODOLOGÍA PROPUESTA PARA ACONDICIONAMIENTO

En primera instancia es necesario tener el Sistema Operativo de Linux, Ubuntu 14.04 LTS el cual se puede conseguir de la página web, <http://www.ubuntu.com/download/ubuntu/download>

Es aconsejable tener solamente el Ubuntu como Sistema Operativo de la Computadora en la cual vayamos a trabajar, ya que pueden existir problemas en la creación del kernel, actualización del Sistema Operativo o en el gestor de arranque el cual se denomina GRUB, de todas maneras si tuviéramos el último problema (el Gestor de Arranque) se plantea una solución en el último punto.

Si desea saber la versión de su Kernel de Ubuntu, abra el terminal (Ctrl+Alt+T) e introduzca el siguiente comando:

```
uname -a
```

Los pasos para poder comenzar con la creación del Kernel en Tiempo Real son:

1. Configuramos la conexión a internet para poder actualizar los repositorios con el Synaptic y también las actualizaciones del sistema mediante el Terminal.
2. A partir de las versiones 12.04LTS ya no viene por defecto el Gestor de paquetes Synaptic, para lo cual entramos al Centro de Software de Ubuntu y en el buscador escribimos Synaptic Gestor, y procedemos a descargar el Gestor de Paquetes Synaptic y luego lo instalamos.



Figura 1: Gestor de Paquetes Synaptic instalado.

3. Abrimos el Gestor de Paquetes Synaptic y abrimos Configuración, seleccionamos la pestaña Repositorios y luego la pestaña Otro Software y habilitamos los paquetes provenientes del Canonical (ver Figura 2), al cerrar aparecerá un cuadro de dialogo en el cual tenemos que seleccionar recargar el Synaptic y luego lo cerramos.



Figura 2: Activar software de Socios de Canonical

- Si no maneja Ubuntu 14.04 y utiliza la versión LTS 12.04 Precise Pongolin, puede utilizar los Real Time PPA de Alessio Bogani, que ha empaquetado amablemente el kernel de vainilla con el parche aplicado RT y se lo mantiene en sintonía con los números de versión en Precise.

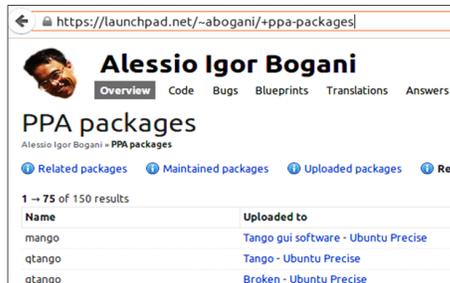


Figura 3: Real Time PPA de Alessio Igor Bogani

IV. CREACIÓN E INSTALCIÓN DE KERNEL EN TIEMPO REAL

- Instale los siguientes paquetes que son requeridos introduciendo lo siguiente en la consola de Ubuntu (Ctrl+Alt+T):

```
sudo apt-get install kernel-package
fakeroot build-essential libncurses5-dev
```

- Luego se procede a buscar el Núcleo Vanilla y su respectivo parche RT en las siguientes direcciones:

<https://www.kernel.org/pub/linux/kernel/v3.x/>
<https://www.kernel.org/pub/linux/kernel/projects/rt/>

En nuestro caso escogemos el parche RT lanzado oficialmente el 17-Jul-2014 15:21 la cual es la versión 3.4

- Para todo ello entonces escribimos los siguientes comandos en la consola:

```
mkdir -p ~/tmp/linux-rt
cd ~/tmp/linux-rt
wget http://www.kernel.org/pub/linux/kernel/v3.x/linux-3.4.tar.bz2
wget https://www.kernel.org/pub/linux/kernel/projects/rt/3.4/older/patch-3.4-rt7.patch.bz2
tar xjvf linux-3.4.tar.bz2
cd linux-3.4
patch -p1 < <(bunzip2 -c ../patch-3.4-rt7.patch.bz2)
```

- Luego Configuramos el Kernel utilizando el siguiente comando:

```
cp /boot/config-$(uname -r) .config &&
make oldconfig
```

Nota: Escoger la opción Full Preemption (opción 5)

- Después procedemos a la creación del Núcleo con el siguiente comando:

```
sed -rie 's/echo "\+/#echo "\+/' scripts
/setlocalversion
make-kpkg clean
CONCURRENCY_LEVEL=$(getconf
_NPROCESSORS_ONLN) fakeroot make-kpkg --
initrd --revision=0 kernel_image
kernel_headers
```

- Luego se instala el Kernel creado con el siguiente comando:

```
sudo dpkg -i ../linux-{headers,image
}-3.4.0-rt7_0_*.deb
```

Eso es todo, Reiniciamos el sistema y debería aparecer el GRUB dándonos la opción de escoger el kernel entre el genérico y el 3.4 rt7 en el arranque.

Podrían ocurrir 2 eventos:

- Que el Grub de Linux permita escoger el Kernel
- Que se inicie el Sistema Operativo directamente con el kernel genérico comprobando una vez iniciado en consola con `uname -a`.

Las soluciones propuestas se dan a conocer en el siguiente punto.

V. SOLUCIÓN Y FINALIZACIÓN DEL ACONDICIONAMIENTO

1. Que el Grub de Linux permita escoger el Kernel

Esto indica que todo está bien y podemos escoger el Kernel con RT o Genérico.

1. Que se inicie el Sistema Operativo directamente con el kernel genérico comprobando, una vez iniciado el OS, en consola con `uname -a`.

Sí éste es el caso una de las soluciones es reinstalar el Grub o instalar el Burg en nuestro Linux, para ello se tiene que realizar los siguientes puntos:

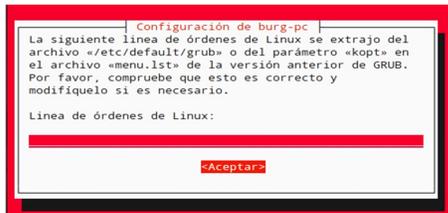
Primero añadimos el repositorio, para ello abrimos una terminal y escribimos:

```
sudo add-apt-repository ppa:n-muench/burg
```

Actualizamos el sistema e instalamos BURG escribiendo:

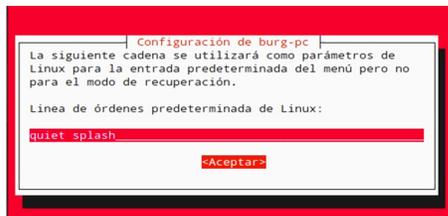
```
sudo apt-get update
sudo apt-get install burg
```

Durante la instalación se presentaran varias pantallas de configuración con diferentes mensajes, cuando aparezcan estas ventanas nos movemos con el teclado, apretamos enter para aceptar y continuamos.



En esta escogemos el tema por defecto, apretamos enter para continuar, ya que después podremos cambiarlo.

Al final nos aparecerá una pantalla para seleccionar donde instalaremos el BURG. Lo usual es que lo hagamos en `dev/sda/` que siempre es el disco duro primario, si tenemos más de un disco duro y no sabemos en cual están instalados nuestros sistemas operativos, entonces lo mejor es seleccionarlos todos.



Para seleccionar el disco navegamos con las flechas del teclado o el tabulador y apretamos la barra de espacio para marcar.

Sabemos que el disco esta seleccionado cuando veamos un asterisco marcado. (Importante seleccionar un disco correctamente o de lo contrario no se instalara el cargador y es lo mismo que si no hiciésemos nada)

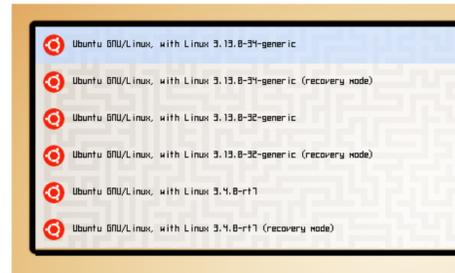


Pulsamos “enter”.

Si no hay ningún error procedemos a simular el burg con el siguiente comando:

```
sudo burg -emu
```

Veremos una terminal como ésta, donde el BURG encuentra todos los sistemas operativos que tengamos instalados y/o versiones de GNU/Linux (Kernels).



REFERENCIAS

- [1] “Journal of Computing and Information Technology”. En: *CIT* 8. Vol. 3. 2000 (vid. pág. 2).