

Instalación de RTAI para Sistemas Operativos en Tiempo Real

by Oswaldo Cuencas.

Instalación de RTAI para Sistemas Operativos en Tiempo Real

Conceptos Basicos

¿Que es kernel?

Planificación de Procesos de Tiempo Real

¿Qué es un proceso de tiempo real?

¿Qué es un Sistema Operativo de Tiempo Real?

Clasificación

Podemos clasificar los procesos de tiempo real de diferentes maneras:

Definicion:

Instalación de RTAI en Virtual box

Requerimientos basicos para la instalación

Caracteristicas del PC sobre el cual se virtualizo ubuntu_rtai

Instalacion de Virtual box v5 y Ubuntu 14.04 con kernel 3.19

Descargar imagen ISO

Descargar e instala Virtual box para su sistema operativos.

Configuracion de caracteristicas para la maquina virtual en Vbox v5.

Pasos para instalación de RTAI en Vbox y sistemas linux-debian/ubuntu

Paso 1: Descargar rtai 4.1 e intalar generic Kernel linux 3.10

Paso 2: Se generan enlaces simbolicos

Paso 3: Actualizar las lista de repositorios

Paso 4: Instalacion de librerias y paquetes para la compilacion del nucleo rtai

Paso 5: Compilar kernel generico 3.10

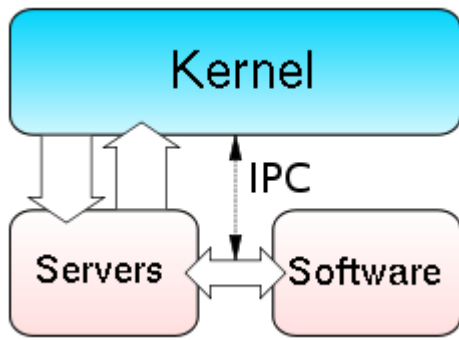
Paso 6: Instalacion del nucleo RTAI en el kernel generico 3.10

Reflexión

Conceptos Basicos

¿Que es kernel?

Kernel es un software que constituye una parte fundamental del sistema operativo. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá usar un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como multiplexado. Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de abstracciones del hardware. Esto permite esconder la complejidad, y proporcionar una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al programador.



Planificación de Procesos de Tiempo Real

¿Qué es un proceso de tiempo real?

Un proceso de tiempo real es aquel cuya actividad tiene un plazo de finalización.

¿Qué es un Sistema Operativo de Tiempo Real?

Un sistema operativo de tiempo real dispone de un planificador de procesos que tiene mecanismos para hacer lo máximo posible para garantizar que sus procesos de tiempo real cumplan los plazos de finalización que tienen establecidos.

Clasificación

Podemos clasificar los procesos de tiempo real de diferentes maneras:

Según el plazo de tiempo:

Estricto (*hard-realtime*): se debe realizar en un plazo de tiempo determinado. Si no lo hace, deja de tener sentido. Si el plazo de tiempo para realizarla es superado el proceso se aborta. Ejemplo: Industriales (sensores, activadores, etc.)

Flexibles (*soft-realtime*): es deseable que se cumpla el plazo de tiempo. Ejemplo: Videoconferencia. (Mientras más rápido vaya mejor será la comunicación, pero si no, nos adaptamos a ella).

Según la periodicidad:

Aperiódicas: se deben a sucesos externos que deben ser atendidos. El sistema operativo no sabe, a priori, cuándo van a llegar ni el tiempo que va a durar. Ejemplo: Notificaciones de errores, Sensores de emergencia en un coche, Ventiladores para refrigeración.

Periódicas: se realizan cada cierto tiempo (actividad repetitiva). A diferencia del anterior el sistema operativo conoce a priori cuándo van a llegar y su tiempo de duración. Ejemplo: leer la temperatura de un sensor.

Definición:

Rtai es un proyecto o iniciativa *Open Source* cuyo objetivo principal es presentar una herramienta especializada para la planificación de tareas en tiempo real. RTAI añade un pequeño núcleo Linux de tiempo real bajo el núcleo estándar de linux y trata al núcleo linux como una tarea de menor prioridad. RTAI además proporciona una amplia selección de mecanismos de comunicación entre procesos y otros servicios de tiempo real.

RTAI tiene una arquitectura similar a RTLinux. Al igual que RTLinux, RTAI trata el núcleo estándar de Linux como una tarea de tiempo real con la menor prioridad, lo que hace posible que se ejecute cuando no haya ninguna tarea con mayor prioridad ejecutándose. Las operaciones básicas de las tareas de tiempo real son implementadas como módulos del núcleo al igual que RTLinux. RTAI maneja las interrupciones de periféricos y son atendidas por el núcleo linux después de las posibles acciones de tiempo real que hayan podido ser lanzadas por efecto de la interrupción.

A continuación se presenta esta herramienta como una alternativa para el procesamiento de tareas en tiempo real, creando una guía o manual de instalación que nos ayude a conceptualizar que es y como usar rtai.

Instalación de RTAI en Virtual box

Requerimientos basicos para la instalación

Notas:

*Es posible instalar rtai en cualquier versión de ubuntu o debian **ya que para ejecutar el nucleo rtai es necesario instalar un nuevo kernel compatible con las características de rtai y sobre el instalar el nucleo de tiempo real**, por ello es muy importantante que se conozca la documentación de cada versión o release de rtai ya que tienen características muy específicas de compatibilidad tanto en kernel como en chipset.*

Aqui solo se presenta una de las muchas alternativas de instalacion.

Características del PC sobre el cual se virtualizo ubuntu_rtai

```
Caracteristicas de CPU por nucleo:
Intel Corporation 2nd Generation Core Processor Family
number_cores:      4
model_name:        Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz
vendor_id:         GenuineIntel
cpu_family:        6
cpu MHz:           799.890 MHz
cache size:        3072 KB
address_sizes:     36 bits physical, 48 bits virtual
```

```
Caracteristicas de RAM
RAM                8GB
```

```
Caracteristicas de Red
driver:            Qualcomm Ather AR8151 v2.0 Gigabit eth0
driver:            Intel Corp Cent Wireless-N 1030 wlan0
```

```
Caracteristicas disco duro
marca:             Kingston
sata:              Gen3 signaling speed (6.0Gb/s)
sdd:               60GB
velocidad real(W): 42.4 MB/s
capacidad usada:   80%
```

```
Kernel linux
distro:            Kali GNU/Linux Rolling 2017.1
kernel_version:    4.12.0-kali1-amd64
```

Puede ser una PC con menor capacidad en RAM, pero no lo sugiero a menos que sea una instalacion sobre hardware real.

Instalacion de Virtual box v5 y Ubuntu 14.04 con kernel 3.19

Descargar imagen ISO

Elegir la version de ubuntu 14.04.3 <http://old-releases.ubuntu.com/releases/14.04.3/>

Se sugiere esta versión ya que tiene compatibilidad con el kernel 3.10 que instalaremos despues, esto para facilitar la misma en la maquina virtual, pero pueden probar con la release de linux que necesiten ya que rtai tiene soporte para kernel 4.x.x. en su version rtai 5.0.1.

Descargar e instala Virtual box para su sistema operativos.

descarga <https://www.virtualbox.org/wiki/Downloads>

Configuracion de caracteristicas para la maquina virtual en Vbox v5.

100 GB Disco duro dinamico (aprovisionamiento) 2 GB RAM CPU 2 Cores Red Tipo: Puente Deshabilitar modo I/O APIC Dar de alta un disco IDE con la direccion de la imagen iso de ubuntu descargada

Instalar la imagen iso en el disco duro virtual como cualquier otro sistema operativo linux virtualizado, en el caso de que no se tenga los datos de como hacerlo de los siguientes links. [Guia para Windows](#) [Guia para Linux](#)

Pasos para instalación de RTAI en Vbox y sistemas linux-debian/ubuntu

Traducción y Adaptación para Vbox de los documentos [gaoyifan](#) y [João Monteiro](#) por Oswaldo Cuencas.

Paso 1: Descargar rtai 4.1 e intalar generic Kernel linux 3.10

Nota: **TODO ES COMO SUPER USUARIO, SI SIGUES ESTE MANUAL COPIA Y PEGA EN TU TERMINAL**

Nos ubicamos en la direccion `/usr/src` y descargamos de la página oficial de rtai el nucleo de tiempo real en su version 4.1, si desea instalar la version mas reciente de kernel consulte: <https://www.rtai.org/?Homepage&id=38>, ademas descargaremos e instalaremos una version generica del kernel linux , para esta caso en especifico la versión 3.10 x64 sobre la cual se va a instalar el nucleo rtai, *este paso dependera de la version del nucleo rtai que se quiera instalar*, para esta caso en particular se deben teclear los siguientes comandos:

```
sudo su
cd /usr/src
curl -L https://www.rtai.org/userfiles/downloads/RTAI/rtai-4.1.tar.bz2 | tar xj
curl -L https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.10.32.tar.xz | tar xJ
curl -L http://kernel.ubuntu.com/~kernel-ppa/mainline/v3.10.32-saucy/linux-image-3.10.32-031032-generic_3.10.32-031032.201402221635_amd64.deb -o linux-image-3.10.32-generic-amd64.deb
dpkg-deb -x linux-image-3.10.32-generic-amd64.deb linux-image-3.10.32-generic-amd64
```

Paso 2: Se generan enlaces simbolicos

```
ln -s linux-3.10.32 linux
ln -s rtai-4.1 rtai
```

Paso 3: Actualizar las lista de repositorios

```
apt-get update
```

Paso 4: Instalacion de librerias y paquetes para la compilacion del nucleo rtai

General: Librerias c para drivers

```
apt-get install cvs subversion build-essential git-core g++-multilib gcc-multilib bc
```

Paquetes Rtai, librerias para crear el nuvo nucleo rtai:

```
apt-get install libtool automake libncurses5-dev kernel-package
```

Paquetes scilab 5.5.0 [acerca de scislab](#)

```
apt-get install default-jre docbook-xsl fop javahelp2 libavalon-framework-java libbatik-  
java libfftw3-3 libflexdock-java libgfortran3 libhdf5-7 libjeuclid-core-java  
libjgoodies-looks-java libjgraphx-java libj hdf5-java libjlatexmath-fop-java  
libjlatexmath-java libjogl-java libjrosetta-java liblapack3gf libncurses5 libpcre3  
libpvm3 libquadmath0 libsaxon-java libskinlf-java libstdc++6 libtinfo5 libxml2 tcl8.5  
tk8.5 zlib1g libgcc1 libc6 libblas-dev libblas3gf libatlas3gf-base gfortran liblapack-  
dev
```

Paquetes qrtailab

```
apt-get install libqt4-dev libqwt5-qt4-dev
```

Paso 5: Compilar kernel generico 3.10

Copia del archivo `conf` del kernel original de ubuntu 3.10 a la carpeta `linux`

```
cp /usr/src/linux-image-3.10.32-generic-amd64/boot/config-3.10.32-031032-generic  
/usr/src/linux/.config
```

patch el kernel

```
cd /usr/src/linux  
patch -p1 < /usr/src/rtai/base/arch/x86/patches/hal-linux-3.10.32-x86-5.patch
```

Configurar el archivo `config` para ajustar las características del kernel generico 3.10

```
make menuconfig
```

Nota: algunas ternimanles linux tiene problemas para mostrar la matriz de configuracion de lo siguiente [link](#) con informacion al respecto.

Realiza las siguientes configuraciones sobre el archivo `config` ya que son las recomendadas, aun así puedes cambiar algunas opciones según las necesidades finales del sistema, revisa todas para verificar que podrías necesitar:

```
Processor type and features  
-> Processor family = Select yours  
-> Maximum number of CPUs (NR_CPUS) = Set your number (it's generally "4")  
-> SMT (Hyperthreading) scheduler support = DISABLE IT  
Power Management and ACPI options  
CPU idle PM support = DISABLE IT
```

Construye-compila el nucleo generico:

Este proceso puede tardar hasta 3 horas dependiendo el tipo de maquina

```
make -j `getconf _NPROCESSORS_ONLN` deb-pkg LOCALVERSION=-rtai
```

Instala el nuevo kernel en la PC virtual el cual se usara para instalar el nucleo rtai despues.

```
cd ..  
dpkg -i linux-image-3.10.32-rtai_3.10.32-rtai-1_amd64.deb  
dpkg -i linux-headers-3.10.32-rtai_3.10.32-rtai-1_amd64.deb
```

La contrucion he instalacion de este kernel se lleva aproximadamente 20GB del disco.

Modificar el archivo grub, en la linea GRUB_CMDLINE_LINUX_DEFAULT as:

```
gksudo nano /etc/default/grub  
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash lapic=notscdeadline"
```

Cargar las nuevas configuracion al grub

```
update-grub
```

Ahora reinicia y preciona `shift + bloq mayus` para iniciar el menu del grub, elige despues `*opciones avanzadas para ubuntu*`, y seleccionar el nuevo kernel instalado rtai_3.10 .

Paso 6: Instalacion del nucleo RTAI en el kernel generico 3.10

Ya iniciada la sesion con el nuevo kernel revisaremos que este esta activo con el comando `uname`

```
sudo su  
uname -r
```

Esta debe ser tu salida:

```
3.10.32-rtai
```

Configurando el archivo `config` de rtai para instalar las características finales del sistema.

```
cd /usr/src/rtai  
make menuconfig
```

Actualizamos la cantidad de CPU reales de los que dispone la maquina virtual en el archivo de configuracion.

Ahora se contruye el nucleo rtai

```
make -j `getconf _NPROCESSORS_ONLN`
```

Se instala el nucleo rtai

```
make install
```

Se agregan las siguientes linea de variables al el archivo bashrc, ~/.bashrc

```
nano ~/.bashrc
export PATH=/usr/realtime/bin:$PATH
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/realtime/lib
```

Hacemos disponibles las nuevas variables de entorno para bash

```
source ~/.bashrc
```

Crea el siguiente archivo `/etc/ld.so.conf.d/rtai.conf` con el siguiente contenido:

```
nano /etc/ld.so.conf.d/rtai.conf
/usr/realtime/lib
```

ejecuta `ldconfig` para tomarlo en cuenta

```
ldconfig
```

Ahora, corre la prueba de latencia:

```
cd /usr/realtime/testsuite/kern/latency
./run
```

Al terminar la configuracion y las pruebas tendras rtai y tu kernel anterior funcionando en la misma PC asi que se puedes elegir cual usar, dependiendo la circunstancia o las pruebas a realizar. Así finaliza la creación del ambiente de pruebas.

Reflexión

En esta prueba de concepto observamos que es posible generar un ambiente de pruebas con procesado en tiempo real dentro de un entorno virtualizado, además de esta manera es práctico poder probar con distintos sistemas operativos. Otra de las ventajas es la posibilidad de trabajar en entornos como ESXi, esto ayuda a generar nuevas estrategias para el consumo de aplicaciones y el aprovechamiento de todo el potencial de un sistema o servidor, ya sea que se encuentre en nube o físicamente dentro de una compañía.

Aun así debe ser cauteloso ya que el uso del núcleo rtai tiene sus limitaciones en cuanto a los chipsets soportados, revise muy bien el soporte del chipset en <https://www.rtai.org/> antes de generar una solución definitiva.