Generación e Instalación de un Núcleo de Tiempo Real en Ubuntu

Fernando G. Tinetti

Reporte técnico TR-RT-02-2021
III-LIDI, Facultad de Informática, UNLP
CIC, Prov. de Buenos Aires
Argentina
fernando@info.unlp.edu.ar
Marzo 2021

Resumen

Se explica cómo generar e instalar un núcleo (*kernel*) de tiempo real en Ubuntu, desde una instalación estándar hasta la puesta en ejecución del binario del núcleo. Si bien se podría directamente proveer el binario a instalar, se explican algunos detalles de las relaciones entre los núcleos utilizados en las distribuciones de Linux y los núcleos disponibles en código fuente. Lamentablemente no hay muchos detalles de cómo las distribuciones (por ejemplo, Debian y sus derivados como Ubuntu o Mint) alteran o agregan software a los núcleos y módulos de Linux "básico" (por decirlo de alguna manera). En cualquier caso, en esta guía se muestra el proceso de obtención del código fuente del núcleo, las opciones de compilación para la administración de CPU con apropiatividad (*preemption*) para las tareas de tiempo real, y la instalación del binario generado a partir de la compilación del código fuente con tales opciones.

1.- Introducción

Aunque durante algún tiempo se mantuvieron distribuciones de Linux con opciones de tiempo real, entre las que estuvo Ubuntu en particular, actualmente no hay una opción "directa" (en el sentido de contar con código binario disponible) de instalar un sistema operativo Linux con apropiatividad para tareas de tiempo real. Sin embargo, existe una comunidad activa de desarrollo y de utilización de este tipo de sistemas. Como mínimo, se dispone de código fuente para el "patch" del núcleo de Linux y de usuarios por ejemplo en el área de robótica. De hecho, se va a utilizar como referencia inicial el documento/guía desarrollado/a en el contexto del área de robótica:
https://docs.ros.org/en/foxy/Tutorials/Building-Realtime-rt_preempt-kernel-for-ROS-2.html

2.- Contexto de Instalación

Este reporte técnico se puede considerar como una continuación de
"Instalación de Ubuntu como Segundo Sistema Operativo a partir de Windows", Fernando G. Tinetti, Reporte técnico TR-RT-01-2021, III-LIDI/CIC Prov. de Bs. As., Fac. de Informática, UNLP, Marzo 2021.

Como paso previo a lo más específico de la construcción del núcleo de tiempo real, se instalarán las herramientas necesarias para la compilación e instalación del mismo, con (es una única línea de comandos en bash):

sudo apt-get install libncurses-dev flex bison openssl libssl-dev dkms libelf-dev libudev-dev libpci-dev libiberty-dev autoconf fakeroot

El contexto de trabajo en el sistema operativo es, entonces, Ubuntu 20.04.1 instalado, actualizado y con las herramientas de generación del kernel con RT_PATCH, donde el espacio en disco total y utilizado es tal como se muestra a continuación:

df -h

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda6	77G	8,3G	65G	12%	/

3.- Binarios Instalados e Identificación de Versiones

En este punto, se deben identificar las versiones de
- Código fuente del núcleo de Linux a utilizar
- Código fuente para el "patch" de tiempo real para el código fuente del núcleo

Estas versiones deben necesariamente coincidir y, además, ser de alguna manera "consistentes" con los binarios instalados, porque sin lugar a dudas todas las herramientas y aplicaciones del sistema operativo instalado van a ser las que finalmente van a requerir servicios del sistema operativo finalmente instalado. Más específicamente, deben considerarse:
- La versión del núcleo de Ubuntu en ejecución
- La versión de fuente del núcleo a usar para incluir PREEMPT_RT (preferentemente la misma que la de Ubuntu en ejecución)
- La versión del patch para incluir PREEMT _RT (idéntica a la versión de fuente del núcleo a usar para incluir PREEMPT_RT)

Tal como se explica en la sección anterior, la versión de Ubuntu instalada y en ejecución es Ubuntu 20.04.1 con todas las actualizaciones. La versión de Ubuntu y la instalación de sus actualizaciones no provee a priori ninguna información de la versión del núcleo instalada y en producción. Para verificar el núcleo instalado y usado en particular se debe recurrir al contenido del directorio /boot, que en este caso es (se eliminó la información de "owner" y "group" del listado, que en todos los casos es root):

ls -l /boot

```
total 131622
-rw-r--r--   237773 jul  9  2020 config-5.4.0-42-generic
-rw-r--r--   248237 feb 10 15:51 config-5.8.0-44-generic
drwx------     2048 dic 31  1969 efi
drwxr-xr-x     4096 mar  3 20:06 grub
lrwxrwxrwx       27 mar  3 19:31 initrd.img -> initrd.img-5.8.0-44-generic
-rw-r--r-- 49992738 mar  3 20:01 initrd.img-5.4.0-42-generic
-rw-r--r-- 52063884 mar  3 20:09 initrd.img-5.8.0-44-generic
lrwxrwxrwx       27 mar  3 19:07 initrd.img.old -> initrd.img-5.4.0-42-generic
-rw-r--r--   182704 ago 18  2020 memtest86+.bin
-rw-r--r--   184380 ago 18  2020 memtest86+.elf
-rw-r--r--   184884 ago 18  2020 memtest86+_multiboot.bin
-rw-------  4738430 jul  9  2020 System.map-5.4.0-42-generic
-rw-------  5519559 feb 10 15:51 System.map-5.8.0-44-generic
lrwxrwxrwx       24 mar  3 19:31 vmlinuz -> vmlinuz-5.8.0-44-generic
-rw-r--r-- 11662080 mar  3 15:41 vmlinuz-5.4.0-42-generic
-rw-------  9735840 feb 10 17:51 vmlinuz-5.8.0-44-generic
lrwxrwxrwx       24 mar  3 19:31 vmlinuz.old -> vmlinuz-5.4.0-42-generic
```

Tal como se puede ver, la versión del núcleo “en funcionamiento” es 5.8.0-44-generic (es a la que apuntan los enlaces simbólicos initrd.img y vmlinuz, por lo tanto, se debe verificar en
http://cdn.kernel.org/pub/linux/kernel/projects/rt/
la versión del patch que le corresponde. En el momento en que se escribe esta guía, el contenido de ese sitio es el que se muestra a continuación, por lo que se debe entonces verificar el contenido del directorio “5.8/”. Lamentablemente no existe el directorio “5.8/” y no está claro exactamente por qué. Las versiones más “cercanas” son 5.6/ y 5.9/. Se elige avanzar por la versión 5.9 asumiendo que provee los mismos servicios que la versión 5.8 utilizada por el sistema en ejecución.

Index of /pub/linux/kernel/projects/rt/

../		
2.6.22/	08-Aug-2013 18:24	-
2.6.23/	08-Aug-2013 18:26	-
2.6.24/	08-Aug-2013 18:27	-
2.6.25/	08-Aug-2013 18:27	-
2.6.26/	08-Aug-2013 18:28	-
2.6.29/	08-Aug-2013 18:28	-
2.6.31/	04-Nov-2014 14:19	-
2.6.33/	08-Aug-2013 18:29	-
3.0/	19-Nov-2013 22:02	-
3.10/	23-Nov-2017 05:44	-
3.12/	08-Jun-2017 13:40	-
3.14/	13-Feb-2017 22:26	-
3.18/	23-May-2019 16:23	-
3.2/	23-Nov-2017 05:53	-
3.4/	16-Nov-2016 19:26	-
3.6/	19-Nov-2013 22:01	-
3.8/	04-Nov-2014 13:35	-
4.0/	13-Jul-2015 21:06	-
4.1/	29-Nov-2017 22:12	-
4.11/	17-Oct-2017 13:42	-
4.13/	17-Nov-2017 17:03	-
4.14/	22-Jan-2021 19:35	-
4.16/	03-Aug-2018 07:39	-
4.18/	29-Oct-2018 11:51	-
4.19/	08-Feb-2021 16:11	-
4.4/	05-Feb-2021 18:01	-
4.6/	30-Sep-2016 21:37	-
4.8/	23-Dec-2016 15:26	-
4.9/	04-Feb-2021 01:30	-
5.0/	10-Jul-2019 15:17	-
5.10/	19-Feb-2021 18:30	-
5.11/	02-Mar-2021 19:49	-
5.2/	16-Dec-2019 17:11	-
5.4/	02-Feb-2021 16:39	-
5.6/	20-Aug-2020 14:23	-
5.9/	28-Oct-2020 20:05	-

Los patches disponibles para la versión 5.9 se muestran en la siguiente página, y con esta información se llega a que se construirá el núcleo para tiempo real a partir del par (núcleo - patch):
linux-5.9.1.tar.gz - patch-5.9.1-rt20.patch.xz

Index of /pub/linux/kernel/projects/rt/5.9/

../		
incr/	28-Oct-2020 20:05	-
older/	28-Oct-2020 20:05	-
patch-5.9.1-rt20.patch.gz	28-Oct-2020 20:05	192K
patch-5.9.1-rt20.patch.sign	28-Oct-2020 20:05	438
patch-5.9.1-rt20.patch.xz	28-Oct-2020 20:05	159K
patches-5.9.1-rt20.tar.gz	28-Oct-2020 20:05	326K
patches-5.9.1-rt20.tar.sign	28-Oct-2020 20:05	438
patches-5.9.1-rt20.tar.xz	28-Oct-2020 20:05	229K
sha256sums.asc	27-Jan-2021 16:21	1253

4.- Generación e Instalación del Núcleo de Tiempo Real

A partir de la identificación de las versiones realizadas, se puede continuar directamente con

```
mkdir rtkernel; cd rtkernel
wget https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.gz
tar -xzf linux-5.9.1.tar.gz
wget http://cdn.kernel.org/pub/linux/kernel/projects/rt/5.9/patch-5.9.1-rt20.patch.gz
gunzip patch-5.9.1-rt20.patch.gz
cd linux-5.9.1/
patch -p1 < ../patch-5.9.1-rt20.patch
cp /boot/config-5.8.0-44-generic .config
yes '' | make oldconfig
make menuconfig
```

En la configuración se debe elegir lo siguiente:

```
=====================================================
# Enable CONFIG_PREEMPT_RT
 -> General Setup
  -> Preemption Model (Fully Preemptible Kernel (Real-Time))
   (X) Fully Preemptible Kernel (Real-Time)

# Enable CONFIG_HIGH_RES_TIMERS
 -> General setup
  -> Timers subsystem
   [*] High Resolution Timer Support

# Enable CONFIG_NO_HZ_FULL
 -> General setup
  -> Timers subsystem
   -> Timer tick handling (Full dynticks system (tickless))
    (X) Full dynticks system (tickless)

# Set CONFIG_HZ_1000 (note: this is no longer in the General Setup menu, go back
twice)
 -> Processor type and features
  -> Timer frequency (1000 HZ)
   (X) 1000 HZ

# Set CPU_FREQ_DEFAULT_GOV_PERFORMANCE [=y]
 ->  Power management and ACPI options
  -> CPU Frequency scaling
   -> CPU Frequency scaling (CPU_FREQ [=y])
    -> Default CPUFreq governor (<choice> [=y])
     (X) performance
=====================================================

make -j `nproc` deb-pkg
```

En la computadora utilizada, Intel I5 10310, 1,7GHz, 8GB RAM, el tiempo de compilación fue de 100 minutos (poco más de una hora y media).

```
ls ../*deb
../linux-headers-5.9.1-rt20_5.9.1-rt20-1_amd64.deb  ../linux-image-5.9.1-rt20-
dbg_5.9.1-rt20-1_amd64.deb
../linux-image-5.9.1-rt20_5.9.1-rt20-1_amd64.deb    ../linux-libc-dev_5.9.1-rt20-
1_amd64.deb
```

Al finalizar la generación del núcleo de tiempo real, el espacio en disco utilizado quedó tal como se muestra a continuación

```
df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda6        77G   39G   35G  53% /
```

Por lo que el espacio necesario en disco para generar el núcleo de tiempo real es considerablemente grande respecto del total del espacio para el sistema operativo completo (8.3GB vs. 39GB). Para guardar y posiblemente reutilizar lo realizado hasta el momento, se puede generar un .tar.gz con todo el directorio núcleo generado (incluyendo el código fuente, de hecho).

Instalar el kernel con
sudo dpkg -i ../*.deb

Una vez instalado el nuevo kernel, el contenido de /boot es:
ls -la /boot

```
total 130218
-rw-r--r--   248237 feb 10 15:51 config-5.8.0-44-generic
-rw-r--r--   247941 mar  3 20:43 config-5.9.1-rt20
drwx------     2048 dic 31  1969 efi
drwxr-xr-x     4096 mar  4 09:45 grub
lrwxrwxrwx       27 mar  3 19:31 initrd.img -> initrd.img-5.8.0-44-generic
-rw-r--r-- 52063884 mar  3 20:09 initrd.img-5.8.0-44-generic
-rw-r--r-- 49578072 mar  3 22:38 initrd.img-5.9.1-rt20
lrwxrwxrwx       21 mar  4 09:43 initrd.img.old -> initrd.img-5.9.1-rt20
-rw-r--r--   182704 ago 18  2020 memtest86+.bin
-rw-r--r--   184380 ago 18  2020 memtest86+.elf
-rw-r--r--   184884 ago 18  2020 memtest86+_multiboot.bin
-rw-------  5519559 feb 10 15:51 System.map-5.8.0-44-generic
-rw-r--r--  5553992 mar  3 20:43 System.map-5.9.1-rt20
lrwxrwxrwx       24 mar  3 19:31 vmlinuz -> vmlinuz-5.8.0-44-generic
-rw-------  9735840 feb 10 17:51 vmlinuz-5.8.0-44-generic
-rw-r--r--  9815648 mar  3 20:43 vmlinuz-5.9.1-rt20
lrwxrwxrwx       18 mar  4 09:43 vmlinuz.old -> vmlinuz-5.9.1-rt20
```

Comparando este contenido con el anterior:

Anterior	Nuevo
config-5.4.0-42-generic	
config-5.8.0-44-generic	config-5.8.0-44-generic
	config-5.9.1-rt20
efi	efi
grub	grub
initrd.img -> initrd.img-5.8.0-44-generic	initrd.img -> initrd.img-5.8.0-44-generic
initrd.img-5.4.0-42-generic	
initrd.img-5.8.0-44-generic	initrd.img-5.8.0-44-generic
	initrd.img-5.9.1-rt20
initrd.img.old -> initrd.img-5.4.0-42-generic	initrd.img.old -> initrd.img-5.9.1-rt20
memtest86+.bin	memtest86+.bin
memtest86+.elf	memtest86+.elf
memtest86+_multiboot.bin	memtest86+_multiboot.bin
System.map-5.4.0-42-generic	
System.map-5.8.0-44-generic	System.map-5.8.0-44-generic
	System.map-5.9.1-rt20
vmlinuz -> vmlinuz-5.8.0-44-generic	vmlinuz -> vmlinuz-5.8.0-44-generic
vmlinuz-5.4.0-42-generic	
vmlinuz-5.8.0-44-generic	vmlinuz-5.8.0-44-generic
	vmlinuz-5.9.1-rt20
vmlinuz.old -> vmlinuz-5.4.0-42-generic	vmlinuz.old -> vmlinuz-5.9.1-rt20

Todos los archivos referidos a la versión 5.4.0-42 ya no están en /boot. No queda claro por qué las referencias a la versión 5.9.1-rt20 están relacionadas por los links simbólicos

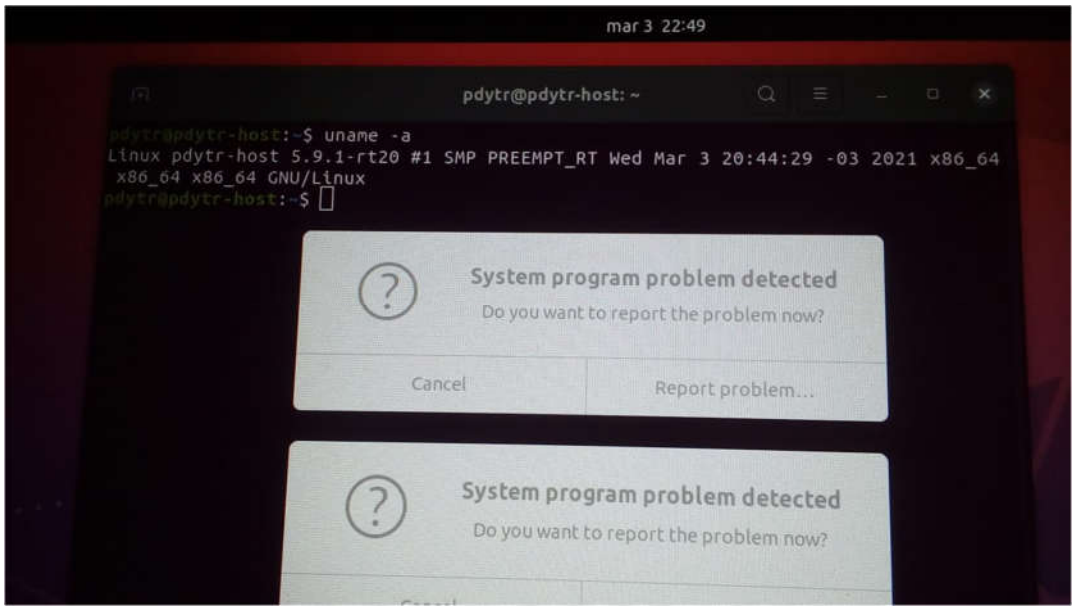
[initrd.img/vmlinuz].old, pero como para que quede consistente (por decirlo de alguna manera), se rehicieron los enlaces simbólicos de initrd.img y vmlinuz para que referencien al linux-rt, de manera tal que quedó

ls -la /boot

```
total 130226
drwxr-xr-x     4096 mar  4 13:27 .
drwxr-xr-x     4096 mar  3 19:09 ..
-rw-r--r--   248237 feb 10 15:51 config-5.8.0-44-generic
-rw-r--r--   247941 mar  3 20:43 config-5.9.1-rt20
drwx------     2048 dic 31  1969 efi
drwxr-xr-x     4096 mar  4 09:45 grub
lrwxrwxrwx       21 mar  4 13:24 initrd.img -> initrd.img-5.9.1-rt20
-rw-r--r-- 52063884 mar  3 20:09 initrd.img-5.8.0-44-generic
-rw-r--r-- 49578072 mar  3 22:38 initrd.img-5.9.1-rt20
lrwxrwxrwx       27 mar  4 13:27 initrd.img.old -> initrd.img-5.8.0-44-generic
-rw-r--r--   182704 ago 18  2020 memtest86+.bin
-rw-r--r--   184380 ago 18  2020 memtest86+.elf
-rw-r--r--   184884 ago 18  2020 memtest86+_multiboot.bin
-rw-------  5519559 feb 10 15:51 System.map-5.8.0-44-generic
-rw-r--r--  5553992 mar  3 20:43 System.map-5.9.1-rt20
Lrwxrwxrwx       18 mar  4 13:25 vmlinuz -> vmlinuz-5.9.1-rt20
-rw-------  9735840 feb 10 17:51 vmlinuz-5.8.0-44-generic
-rw-r--r--  9815648 mar  3 20:43 vmlinuz-5.9.1-rt20
Lrwxrwxrwx       24 mar  4 13:26 vmlinuz.old -> vmlinuz-5.8.0-44-generic
```

5.- Inicio con el Núcleo de Tiempo Real

Al iniciar la computadora con el núcleo de tiempo real, se muestran algunos errores en la consola de inicio, antes de pasar a la interfaz gráfica estándar. Una vez en la interfaz gráfica aparecen inicialmente algunos reportes de errores, pero en principio no parecen problemas de mucha envergadura (no se muestran detalles en las pantallas que aparecen):



Una vez iniciado el sistema, que toma un tiempo similar al sistema original, se puede tener todo lo anterior, en particular una terminal, en la cual se puede verificar el reporte del sistema operativo:

```
uname -a
Linux pdytr-host 5.9.1-rt20 #1 SMP PREEMPT_RT Wed Mar 3 20:44:29 -03 2021
x86_64 x86_64 x86_64 GNU/Linux
```

6.- Conclusiones

La generación del núcleo de tiempo real no presenta mayores inconvenientes una vez que se elige el código fuente y su patch correspondiente. Lamentablemente, la utilización de núcleos por parte de las distribuciones de Linux (como Ubuntu, en este caso) no se realiza en función de la disponibilidad del patch de tiempo real correspondiente. Esto, además de la falta (o desconocimiento) de la documentación disponible sobre la utilización de los servicios del núcleo y módulos/bibliotecas asociados por parte de las aplicaciones usadas en Ubuntu hace difícil justificar o probar de antemano que el núcleo de tiempo real funcionará de antemano sin ningún tipo de error. En última instancia, se comprobará en los propios experimentos y aplicaciones de tiempo real a utilizar (algo que de alguna manera se puede considerar una situación *clásica* en general y para el área de tiempo real en particular).

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
https://sookocheff.com/post/linux/how-to-install-the-real-time-kernel-in-ubuntu/
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
To use Ubuntu as an audio production server you need the real-time kernel. This allows applications to have some guarantee of the maximum response time of any task. High priority tasks are given the CPU within a fixed minimum amount of time allowing for processor intensive tasks to actually be completed. Follow along to see how to install the real-time kernel yourself. A vanilla install of most Linux distributions does not enable the real-time kernel, but it is easy enough to install.

sudo apt-get install linux-rt

to install it. Now, we have to change the boot loader (GRUB) to load the real-time kernel. Edit the file

sudo gedit /etc/default/grub

by commenting out the line

GRUB_HIDDEN_TIMEOUT=0

then update the GRUB configuration.

sudo update-grub

This will allow you to select which kernel to run at boot time so you can alternate between the real-time and vanilla kernels depending on your needs.

https://stackoverflow.com/questions/51669724/install-rt-linux-patch-for-ubuntu
https://hmenn.github.io/pages/UbuntuRT_patch.html

Hay varias guías disponibles, no fue usada la propia de Ubuntu sino inicialmente
https://www.partitionwizard.com/partitionmagic/install-linux-on-windows-10.html

Depende de cada computadora/motherboard, como se explica en https://www.isunshare.com/windows-password/how-to-set-your-computer-to-boot-from-usb-drive.html

En el caso de portables dell, presionar F12 en el arranque, ni bien se enciende, antes de que aparezca el inicio de Windows en pantalla.

Cuando el USB es "booteable", aparece en la lista de posibilidades

E. Instalar Paquetes Básicos

sudo add-apt-repository universe
sudo apt install joe

Muy rápidamente se nota que nada de esto tiene sentido en un USB donde está el .ISO de instalación de Ubuntu, porque ese ISO solo existe o tiene sentido para instalar, no para mantener un sistema operativo completo Up&Running, así que habría que instalar Ubuntu en un pendrive y sacar de ahí el .ISO para distribuir un Ubuntu listo para usar.

Parte II Instalar Ubuntu desde USB a otro USB

Siguiendo las instrucciones de https://www.fosslinux.com/10212/how-to-install-a-complete-ubuntu-on-a-usb-flash-drive.htm

Iniciar desde USB con el Ubuntu bajado desde el sitio de Ubuntu

try ubuntu without installing

usar gparted
- select the right disk
- format to clear
- exit gparted

select install ubuntu from the desktop

idioma/s: inglés para el SO y el del teclado que corresponde

no conectar a internet

instalación estándar

tipo de instalación ⇒ más opciones (no boot manager)