Capítulo 4

“Instalación de UML”

4.1 Introducción

Antes de dedicar tiempo de explicación a la instalación de *User Mode Linux*, será necesario tener en claro algunos aspectos relacionados con el manejo del sistema operativo Linux. La presente sección tiene como objetivo brindar los conocimientos necesarios para tal fin.

Se debe tener en cuenta que en Linux los archivos para instalar distintas aplicaciones suelen tener terminaciones como tar.gz, tar.bz2, bz2, rpm [25]. Los archivos terminados en .tar constituyen un tipo de formato muy común en *Unix* y a veces se los denomina tarfiles o tarballs. Se los puede abrir con la herramienta tar. Este formato lo único que hace es "unir" archivos unos a otros y crear un archivo que los contiene a todos. No hay ningún tipo de compresión asociada con este mecanismo.

Los archivos terminados en gz están comprimidos usando el formato gzip. Para descomprimirlos se usa el programa gzip. De igual forma los archivos terminados en bz2 se descomprimen mediante el programa bzip2. Un archivo tar.gz (tar.bz2) es un archivo tar comprimido utilizando gzip (bzip2). Para descomprimirlo utiliza gzip (bzip2) y para desarchivarlo tar. En ambientes *Unix* toda esta secuencia se pude efectuar mediante un único comando:

```
# gzip -dc archivo.tar.gz  | tar xvf -

# bzip2 -dc archivo.tar.bz2  | tar xvf -
```

Esto es tan frecuente que la versión GNU de tar (la que Linux usa) es capaz de hacerlo por si mismo, con:

```
#tar xvzf archivo.tar.gz

#tar --bzip -xvf archivo.tar.bz2
```

El RPM se lo considera como un programa sencillo y seguro para gestionar los paquetes de un sistema. Si el paquete RPM es una aplicación, se lo denomina fichero binario precompilado. RPM tiene cinco modos de operación básicos (sin contar la construcción de paquetes): instalación, desinstalación, actualización, consulta y verificación.

Los paquetes RPM normalmente tienen nombres foo-1.0-1.i386.rpm. El nombre de fichero incluye el nombre de paquete (foo), versión (1.0), lanzamiento (1) y arquitectura (i386). La instalación de un paquete es tan simple como teclear el siguiente comando en el Intérprete de comandos de shell:

```
# rpm -ivh foo-1.0-1.i386.rpm

   foo    ##################################
```

Ejecutado el comando RPM, imprime el nombre del paquete y el progreso mientras se instala

4.2 Instalación de UML

4.2.1 Aspectos Generales

Tres elementos de *software* son necesarios para la instalación de UML: una versión del núcleo Linux [27] independiente de la del *host* anfitrión pues el núcleo de Linux tiene la capacidad de ejecutarse en el espacio de usuario como cualquier otro proceso de usuario, un *filesystem* basado en alguna de las numerosas versiones existentes de Linux (igual o diferente a la existente en el *host* anfitrión) y las utilidades UML. Las versiones de núcleos previas a la 2.6.9.x necesitan de un elemento de *software* adicional a efectos de construir un núcleo soportando la arquitectura UML. Una vez que se dispone de los elementos de *software* mencionados, el proceso de instalación es sencillo y existe abundante literatura sobre ello.

4.2.2 Premisas de Instalación

Se debe descomprimir el *kernel* linux (linux-2.6.10.tar.bz2) en una carpeta creada para tal fin, se recomienda crearla en /usr/local/src/ [1].

```
[root@Ryc2 root]# cd /usr/local/src/
[root@Ryc2 src]# mkdir UML
```

De esta manera se genera una carpeta UML, donde se colocan todos los fuentes necesarios para la instalación.

El siguiente paso consiste en descomprimir el *kernel*.

```
[root@Ryc2 UML]# tar -xjvf linux-2.6.10.tar.bz2
```

De esta forma se genera una carpeta dentro de /UML, llamada /linux-2.6.10 que contiene los archivos descomprimidos del *kernel*.

El siguiente paso se debe llevar a cabo si se utiliza un *kernel* cuya versión es inferior a 2.6.10, como por ejemplo el *kernel* linux 2.6.8.1.

Una vez descomprimidos los archivos del *kernel* se debe copiar en la carpeta /linux-2.6.8.1 el patch UML (UML-patch-2.6.8.1-1.bz2). ahora se deberá descomprimir y patchear el *kernel*.

```
[root@Ryc2 UML]# cp UML-patch-2.6.8.1-1.bz2 /usr/local/src/UML/linux-2.6.8.1/
[root@Ryc2 UML]# cd linux-2.6.8.1
[root@Ryc2 linux-2.6.8.1]#bzcat  UML-patch-2.6.8.1-1.bz2 | patch -p1
```

El siguiente paso consiste en configurar el *kernel* bajo la arquitectura UML, utilizando el comando que se detalla a continuación:

[root@Ryc2 linux-2.6.10]#make xconfig ARCH=um

Como resultado al comando anterior, emergerá una ventana tal como se ilustra en la Figura 4.1:

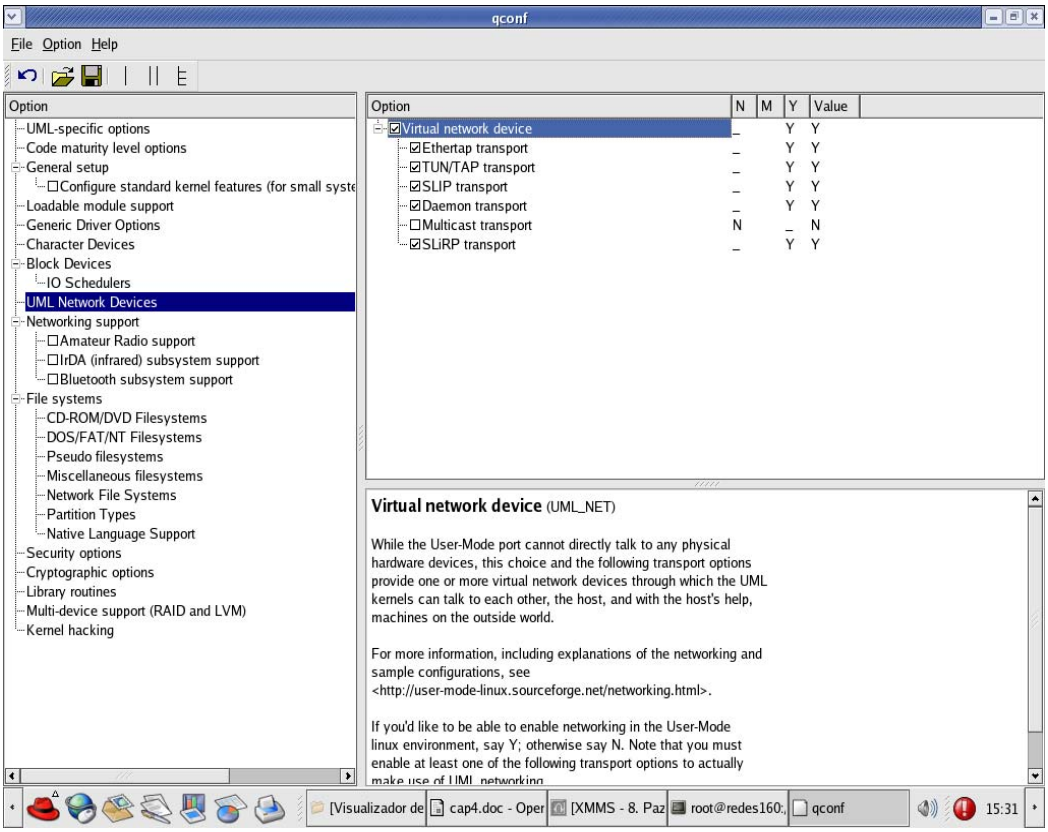

Figura 4. 1: Configuración del Kernel Linux según la Arquitectura UML

Una vez que se ha configurado el *kernel* UML, el siguiente paso es crear el archivo binario linux, que posibilitará ejecutar las máquinas virtuales (ver Figura 4.2):

[root@Ryc2 linux-2.6.10]#make linux ARCH=um

Al llegar a este punto queda concluida la instalación del *kernel* UML. Ahora es necesario dar comienzo a la instalación de las utilidades. Para ello se debe copiar el paquete de utilidades en la carpeta /linux-2.6.10 [1].

[root@Ryc2 UML]# cp UML_utilities_20040406.tar.bz2 /usr/local/src/UML/linux-2.6.10/

[root@Ryc2 UML]# cd linux-2.6.10

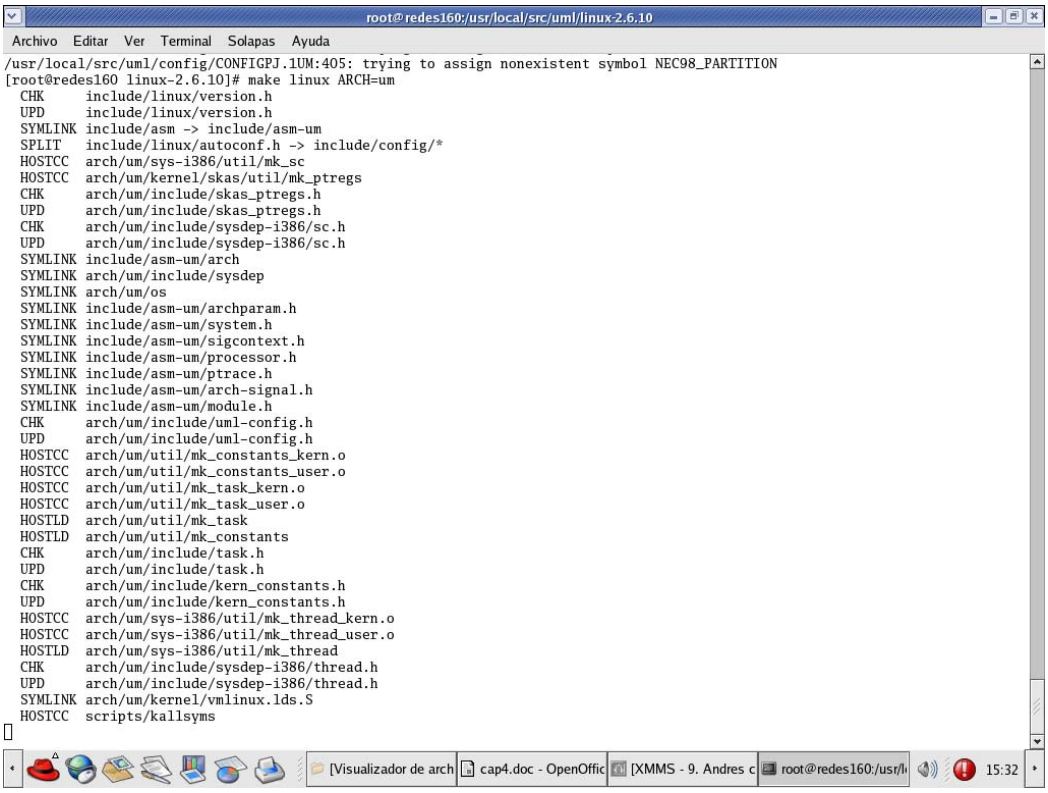

Figura 4. 2: Compilación del Kernel Arquitectura UML

Luego se debe descomprimir el paquete de utilidades con el siguiente comando.

[root@Ryc2 linux-2.6.10]#bunzip2 UML_utilities_20040406.tar.bz2

[root@Ryc2 linux-2.6.10]#tar -xvf UML_utilities_20040406.tar

Tras ejecutar el último comando se crea automáticamente una carpeta en /linux-2.6.10 llamada /tools en donde se encuentran todas las herramientas que UML necesita para poder construir distintos escenarios de red.

Finalmente es necesario compilar e instalar las utilidades UML. A continuación se listan los comandos para llevar a cabo esta operación.

[root@Ryc2 linux-2.6.10]#cd tools

[root@Ryc2 tools]#make && make install

De esta forma se ha culminado con el proceso de instalación de UML. El capitulo siguiente trata sobre como ejecutar múltiples máquinas virtuales minimizando el espacio de memoria requerido y los procedimientos necesarios para su interconexión en red.