



Curso avanzado de Linux

Kernel Linux

Rafael Varela Pet

Unidad de Sistemas

Área de Tecnologías de la Información y Comunicaciones
Universidad de Santiago de Compostela

Método clásico

- Utilizamos el kernel oficial: <ftp.kernel.org>
- Mirror parcial en USC:
`ftp://ftp.usc.es/pub/mirror/kernel`
- Herramientas necesarias:
(todas ellas disponibles como paquetes Debian)
 - Compilador C
 - Librerías desarrollo
 - Utilidades

Método clásico. Configuración.

- La configuración se guarda en el fichero `.config`
- El kernel no provee ningún fichero de configuración
(se generará uno por defecto en caso de no tener ninguno)
- Podemos copiar la de alguno de los kernel instalados

```
cp /boot/config-X.Y.Z .config
```

(también se puede importar desde la herramienta de configuración)

Método clásico. Compilar

- Generación imagen kernel:
 - `make mrproper` (o `make clean`)
 - `make menuconfig` (o `config`, `xconfig`, `gconfig`)
 - configurar según lo requerido
 - Generar el paquete:
 - `make dep` (innecesario en el kernel 2.6)
 - `make bzImage`
 - Generar los módulos
 - `make modules`

Método clásico. Instalar

- Instalar:

```
make install
```

(el kernel va a /boot)

```
make modules_install
```

(los módulos van a /lib/modules/<version>)

- Configurar gestor de arranque.

Es conveniente tener un kernel alternativo para poder arrancar en caso de problemas

- Si queremos los ficheros generados en otro sitio:

```
make O=/tmp/build/kernel menuconfig
```

```
make O=/tmp/build/kernel bzImage
```

```
sudo make O=/tmp/build/kernel modules_install install
```

Método Debian

- Empleamos el paquete `kernel-package`
- Genera paquetes `.deb` con todo lo necesario:
 - `kernel`
 - `módulos`
 - `disco RAM inicial (initrd)`
 - `configuración del gestor de arranque`
- Configuración en `/etc/kernel-pkg.conf`

Método Debian

- Podemos usar paquetes Debian con el código fuente, el kernel oficial o cualquier otra variante.
- Usaremos los kernel Debian
 - `aptitude install linux-source-2.6.18`
 - Fuente en `/usr/src/linux-source-2.6.18.tar.bz2`
(usar “`dpkg -L`” para ver qué se ha instalado)
 - Descomprimir:
`tar -jxf /usr/src/linux-source-2.6.18.tar.bz2`

Método Debian. Compilar

- Generación imagen kernel:
 - `make menuconfig` (o `config`, `xconfig`, `gconfig`)
 - Configurar según lo requerido
 - Generar el paquete:

```
make-kpkg clean  
make-kpkg --revision=curso.1.0 --initrd \  
kernel_image
```

(como root, usar `sudo` o `fakeroot`)

El parámetro “revision” afecta al nombre del paquete pero no a la versión del kernel. Ver comando 'uname -r'

Método Debian. Compilar

- La configuración por defecto del cargador en Debian (LILO, GRUB) usa un disco RAM inicial.
- Es necesario incluir la opción `--initrd` o ajustar la configuración del cargador
- Ejemplo completo (para sistemas multiprocesador):

```
CONCURRENCY_LEVEL=2 fakeroot make-kpkg \  
  --revision=curso.1.0 \  
  --initrd \  
  kernel_image
```

Instalar paquete

- Instalar la imagen
`dpkg -i imagen.deb`
- La imagen del kernel se instala en /boot y los módulos en /lib/modules/<versión>
- Comprobar gestor arranque antes de reiniciar

Método Debian. Cabeceras.

- Generación cabeceras kernel:
 - Los mismos pasos que para generar la imagen binaria
 - `fakeroot make-kpkg kernel_headers`
(necesario ejecutarlo como root, usar sudo o fakeroot)
 - `dpkg -i imagen.deb`
- Generar el paquete de cabeceras no es obligatorio, pero será necesario tan pronto queramos compilar un módulo (para instalar el driver de una tarjeta gráfica, por ejemplo)

Referencias

- `/usr/share/doc/kernel-package/README.gz`