

Los componentes del núcleo

El núcleo Linux se encarga de administrar los recursos materiales de la máquina y de ponerlos a disposición de los diferentes programas que se ejecutan en esta máquina. De entre sus principales tareas, podemos citar:

- ✓ Gestión del o de los microprocesadores: da acceso al procesador o a los procesadores a los diferentes programas, en función de su prioridad para que todos puedan acceder al recurso y que un programa no monopolice el o los procesadores (**multitarea apropiativa**).
- ✓ Gestión de dispositivos: junto con los drivers de los dispositivos (*drivers*), gestiona los diferentes dispositivos de software y trata las solicitudes de los programas que quieren acceder a ellos.
- ✓ Gestión de la memoria física y virtual: determina la reparto de memoria entre él mismo y los programas, y gestiona un mecanismo de memoria virtual que permite asignar más memoria de la que dispone físicamente la máquina.
- ✓ Gestión del almacenamiento: gestiona el conjunto de los dispositivos de almacenamiento (con drivers de dispositivos y drivers de sistemas de archivos) y presenta a los programas una interfaz de acceso único, bajo la forma de una arborescencia de directorios y de archivos, construida a partir de diferentes sistemas de archivos.
- ✓ Gestión de la red: toma en cuenta la comunicación en las redes en las que la máquina participa, a través de la tarjetas de interfaz de red y de sus drivers y permite que los diferentes programas emitan y reciban mensajes.
- ✓ Gestión de los procesos: asegura la carga, la ejecución, el control y la finalización de los programas, bajo la forma de **procesos**, lo que les permite ejecutarse en simultaneidad aparente o real (según las capacidades del procesador de la máquina). Gestiona la seguridad mediante el control del acceso a los recursos en función de las habilitaciones de los procesos y a través del aislamiento de las zonas de memoria que le son asignadas.
- ✓ Gestión del control de acceso a los recursos: con un modelo de seguridad basado en las nociones de cuenta de usuario y de grupo, proporciona un mecanismo que permite definir y controlar los permisos de acceso a los diferentes recursos que gestiona.

Para realizar estas tareas, el núcleo puede usar módulos de tipo software especializados, que pueden estar incluidos directamente en el código del núcleo o ser independientes, cargables y descargables dinámicamente (LKM, *Loadable Kernel Module*).

La mayoría de los drivers de dispositivos son módulos LKM, independientes del programa principal del núcleo (*core kernel*), lo que facilita su mantenimiento y evita tener que cargarlos durante el arranque del sistema si no son utilizados.

El núcleo está constituido por el conjunto de imagen del núcleo y de los módulos.

1. El núcleo

Lo que podemos llamar el programa principal del núcleo (*core kernel*), es la parte esencial del núcleo. Está almacenado en un archivo de imagen comprimido, que será descomprimido y cargado en memoria durante el arranque del sistema. Este programa se hace cargo de las funciones esenciales del núcleo e incluye algunos módulos estáticos.

2. Los módulos de núcleo

Los módulos de núcleo se ocupan de los dispositivos de hardware o conjuntos de funciones (gestión de un tipo de sistema de archivos, de una forma de almacenamiento, de un protocolo de red ...). Funcionan en modo núcleo, pero constituyen programas independientes del programa principal del núcleo.

Existen dos tipos de módulos de núcleo:

- ~ Algunos módulos necesarios permanentemente para el buen funcionamiento del sistema están integrados directamente en el archivo imagen del núcleo y son, por lo tanto, cargados junto a él durante el arranque. Se trata de **módulos estáticos**.
- ~ Los otros módulos están cargados en memoria dinámicamente, en función de las necesidades, y pueden ser descargados de la memoria si ya no se utilizan. Se les llama módulos LKM (*Loadable Kernel Module*). Cada módulo LKM está almacenado en un archivo específico.

Si instalamos un nuevo tipo de hardware, o si queremos usar una nueva funcionalidad (un tipo de sistema de archivos, por ejemplo), es posible que el módulo de software correspondiente no esté integrado en el núcleo usado ni instalado como módulo dinámico. En ese caso hay que instalar el nuevo módulo si es dinámico, o integrarlo en el núcleo y generar una nueva imagen de este último.

3. Las versiones del núcleo

El núcleo Linux, creado por **Linus Torvalds** en 1991, es un programa de software libre, proporcionado gratuitamente bajo licencia GNU GPL v2 (GNU *General Public License*). Su desarrollo está mantenido por una fundación, *The Linux Foundation*, financiada por numerosas empresas y organizaciones.

El desarrollo del núcleo está asegurado por una gran comunidad de programadores, a través de Internet, bajo la dirección técnica de Linus Torvalds y del desarrollador principal, **Greg Kroah-Hartman**.

El núcleo Linux está en constante evolución para corregir problemas, optimizar las funciones ya existentes e integrar nuevos dispositivos o nuevas funcionalidades. Se publican nuevas versiones frecuentemente, cada dos o cuatro meses de media.

Para asegurar el mantenimiento de las versiones en producción, la fundación Linux soporta versiones del núcleo estables en curso y versiones estables mantenidas a largo plazo, proporcionando correcciones de bugs o de problemas de seguridad. Actualmente hay dos versiones en curso estables y seis versiones de núcleo de soporte a largo plazo, soportadas durante dos años, como mínimo.

Las versiones del núcleo están numeradas bajo la forma X.Y.Z.

X proporciona el número de versión mayor (*major*).

Y proporciona el nivel en curso de la versión principal (*minor*). El conjunto X.Y identifica una versión principal del núcleo.

Z es un índice, aumentado en cada evolución menor de la versión principal (correcciones de bugs, nuevos dispositivos ...).

En las primeras versiones del núcleo, las versiones de producción tenían un valor menor Y par, las que estaban siendo desarrolladas y en modo test presentaban un valor impar. Esta convención ha sido abandonada.

En 2011 apareció la versión 3.0, hecho que fue más destacado por el vigésimo aniversario del núcleo que por aportar una evolución mayor al sistema.

La versión principal actual es la 5.Y.

Ejemplo

La página www.kernel.org proporciona actualmente la tabla siguiente para las versiones estables y de soporte a largo plazo:

stable	5.14.12
stable	5.13.19
longterm	5.10.73
longterm	5.4.153
longterm	4.19.211
longterm	4.14.250
longterm	4.9.286
longterm	4.4.288

Las distribuciones integran versiones del núcleo generalmente más antiguas que las versiones en curso, porque integrar una versión nueva del núcleo demanda muchas pruebas complejas.

El comando `uname -r` permite mostrar la versión del núcleo que se está ejecutando.

Ejemplo

Version de núcleo en una distribución CentOS 8 y en una distribución Debian 10:

CentOS 8:

```
uname -r
4.18.0-305.19.1.el8_4.x86_64
```

Debian 10:

```
uname -r
4.19.0-17-amd64
```

La versión presenta un guion e información sobre la versión de la distribución (`e18`: Enterprise Linux 8) y la arquitectura del procesador (Intel 64 bits y AMD 64 bits).

4. Los archivos imágenes del núcleo

El archivo imagen del núcleo contiene el programa principal del núcleo, incluyendo los módulos estáticos comprimidos. Este archivo se descomprimirá dinámicamente y se cargará en memoria durante el arranque del sistema.

Puede haber algunos archivos imágenes del núcleo en un mismo sistema, diferenciados generalmente por un sufijo que identificará la versión.

Existen dos formatos de archivos imágenes del núcleo, dependiendo de la versión del núcleo y de las características del hardware de la máquina en cuestión:

zImage

Usado en las antiguas versiones del núcleo, este formato presupone que el tamaño

máximo de la imagen (descomprimida) será inferior a 512 KB. La compresión se hace en formato gzip o bzip2. Este tipo de imagen solo se utiliza en sistemas antiguos o en máquinas específicas (routers y sistemas embarcados) que disponen de poca memoria viva.

bzImage

Es el formato estándar para las versiones recientes. El límite de tamaño de la imagen descomprimida es alto. Al contrario de lo que podríamos pensar, la compresión no se hace obligatoriamente en formato bzip2, **bz** significa *big z-image*.

Para saber cuál es el tipo de un archivo de imagen, podemos usar el comando **file**.

Ejemplo

Tipo y tamaño de una imagen de núcleo en una distribución CentOS 8:

```
file /boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64
/boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64: Linux kernel x86 boot executable
bzImage, version 4.18.0-305.19.1.el8_4.x86_64 (mockbuild@kbuilder.bsys.
centos.org) #1 SMP Wed Sep 15 15:39:39 UT, RO-rootFS, swap_dev 0x9, Normal
VGA
ls -l /boot/vmlinuz-4.18.0-305.19.1.el8_4.x86_64
-rwxr-xr-x. 1 root root 10030216 sep 15 11:48 /boot/vmlinuz-4.18.0-
305.19.1.el8_4.x86_64
```

Se trata de una imagen reciente, para una arquitectura Intel de 64 bits. Su formato es **bzImage**, su tamaño comprimido es 9,6 MB, aproximadamente.

Las versiones recientes de archivos de imágenes del núcleo pueden usar diferentes formatos de compresión, entre los que se encuentra el del comando **xz**.

5. El archivo disco virtual complementario del núcleo

Algunos módulos dinámicos LKM pueden ser necesarios en el momento de la carga y de la inicialización del núcleo, con el objetivo de ocuparse de ciertos dispositivos. Estos módulos están agrupados en un archivo de disco virtual (*RAM disk*), cargado en memoria y en el núcleo a la vez.

Tradicionalmente, este archivo presentaba el formato `initrd`, que necesitaba que un driver de sistema de archivos (ext2) fuera integrado en el núcleo. Cada vez más a menudo, el archivo presenta el formato `initramfs`, lo que hace que no se necesite ningún driver de sistema de archivos integrado en el núcleo, lo que resulta en una optimización.

6. La documentación del núcleo

La documentación del núcleo resulta a menudo útil, a veces indispensable, para configurar y generar una nueva versión ejecutable del núcleo.

En las antiguas versiones, estaba incluida en el código fuente del núcleo, e instalada por defecto en el directorio `/usr/src/linux/Documentation`.

Desde la versión 3.0 del núcleo, la documentación se proporciona aparte, en el paquete `kernel-doc`. Este último instala la documentación, por defecto, en el directorio específico de la versión del núcleo: `/usr/share/doc/kernel-doc*/Documentation`.

La URL <https://www.kernel.org/doc> da acceso a toda la documentación referente al núcleo.



Esta documentación es muy técnica. Está constituida por muchos archivos de texto, en inglés, con nombres a veces enigmáticos. Sin embargo es muy completa y está muy detallada.

[Ejemplo](#)

Instalación de la documentación del núcleo en una distribución de tipo Red Hat:

```

yum install kernel-doc CentOS Linux 8 - AppStream          9.4 kB/s | 4.3 kB  00:00
CentOS Linux 8 - AppStream          4.0 MB/s | 9.3 MB  00:02
CentOS Linux 8 - BaseOS             10 kB/s | 3.9 kB  00:00
CentOS Linux 8 - Extras             3.7 kB/s | 1.5 kB  00:00
Dependencias resueltas.
=====
Paquete                Arquitectura          Versión                Repositorio
=====
Instalando:
kernel-doc             noarch                4.18.0-305.19.1.el8_4
                        baseos                23 M
Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 23 M
Tamaño instalado: 86 M
¿Está de acuerdo [s/N]?: s
[...]
Instalado:
kernel-doc-4.18.0-305.19.1.el8_4.noarch

¡Instalación terminada!
ls -l /usr/share/doc/kernel-doc-4.18.0/Documentation/
total 1224
total 1124
-r--r--r--. 1 root root 14465 sep 7 06:15 00-INDEX
drwxr-xr-x. 7 root root  92 oct 14 15:48 ABI
drwxr-xr-x. 2 root root  22 oct 14 15:48 accelerators
drwxr-xr-x. 2 root root 121 oct 14 15:48 accounting
drwxr-xr-x. 4 root root 4096 oct 14 15:48 acpi
drwxr-xr-x. 8 root root 4096 oct 14 15:48 admin-guide
drwxr-xr-x. 2 root root 112 oct 14 15:48 aoe
drwxr-xr-x. 17 root root 4096 oct 14 15:48 arm
drwxr-xr-x. 2 root root 4096 oct 14 15:48 arm64
-r--r--r--. 1 root root 9349 sep 7 06:15 asm-annotations.rst
-r--r--r--. 1 root root 1551 sep 7 06:15 atomic_bitops.txt

```



```
-r--r--r--. 1 root root 5750 sep 7 06:15 atomic_t.txt
[...]
drwxr-xr-x. 2 root root 40 oct 14 15:48 xtensa
-r--r--r--. 1 root root 5660 sep 7 06:15 xz.txt
-r--r--r--. 1 root root 2971 sep 7 06:15 zorro.txt
```

El último archivo de la lista, con el extraño nombre de `zorro.txt`, está relacionado con un bus específico de las máquinas Amiga, ¡cuya última actualización se hizo en 2003!

El archivo `cdrom/ide-cd` contiene la documentación del driver del lector de CD CD IDE:

```
vi /usr/share/doc/kernel-doc-4.18.0/Documentation/cdrom/ide-cd
```

```
IDE-CD driver documentation
```

```
Originally by scott snetder <snetder@fnald0.fnal.gov> (19 May 1996)
```

```
Carrying on the torch is: Erik Andersen <andersee@debian.org>
```

```
New maintainers (19 Oct 1998): Jens Axboe <axboe@image.dk>
```

```
1. Introduction
```

```
-----
```

```
The ide-cd driver should work with all ATAPI ver 1.2 to ATAPI 2.6 compliant
CDROM drives which attach to an IDE interface. Note that some CDROM vendors
[...]
```