

El kernel linux



IES Gonzalo Nazareno
CONSEJERÍA DE EDUCACIÓN

Alberto Molina Coballes



23 de septiembre de 2018

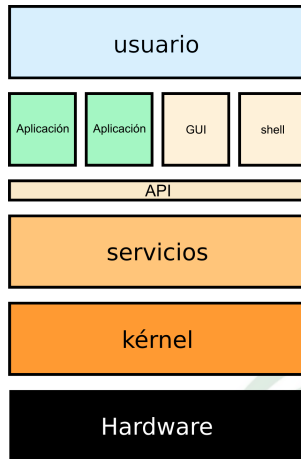
Introducción

- Funciones del sistema operativo
- Funciones del kernel
- Niveles de seguridad
- Tipos de kernel
- Principales arquitecturas CPU y ports
- Principales sistemas operativos
- Historia de UNIX
- La lucha de los sistemas móviles
- El kernel linux. Características y compilación



Funciones del sistema operativo

- Un sistema operativo es una interfaz entre el hardware y el usuario y se encarga de gestionar y compartir los recursos.
- Las principales funciones del sistema operativo son:
 - Gestión de los recursos de la computadora
 - Ejecución de servicios para las aplicaciones
 - Ejecución de las órdenes de los usuarios



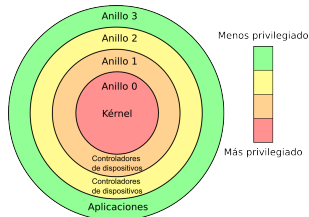
Funciones del kernel

- El kernel (núcleo) es la parte fundamental del sistema operativo y se encarga de manejar los recursos y permitir que los programas hagan uso de los mismos, siendo los principales recursos:
 - CPU
 - Memoria
 - Dispositivos de Entrada/salida
- Además el kernel es el encargado proporcionar:
 - Protección mediante diferentes niveles de acceso
 - Acceso compartido (multiplexado) a los recursos

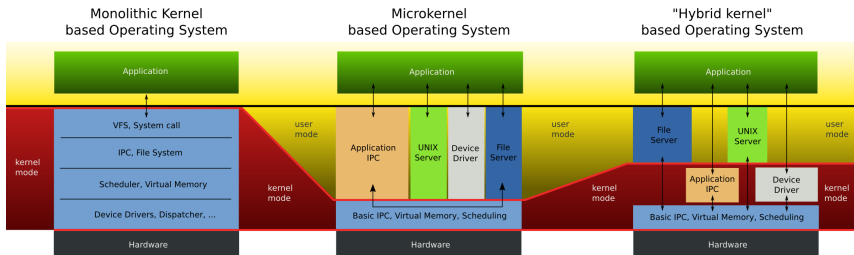


Niveles de seguridad

- Algunas CPU incluyen diferentes niveles de acceso, que se conocen como anillos (rings).
- Los diferentes kernel suelen utilizar al menos dos niveles para acceder tanto a la CPU como a la memoria:
 - kernel mode.
 - user mode
- KML: kernel mode linux
- FUSE: Filesystem in User Space (sshfs, ntfs-3g, ...)
- CUSE: Character devices in User Space



Tipos de kernel



Fuente: <http://upload.wikimedia.org/wikipedia/commons/d/d0/OS-structure2.svg>

Principales arquitecturas CPU/ports

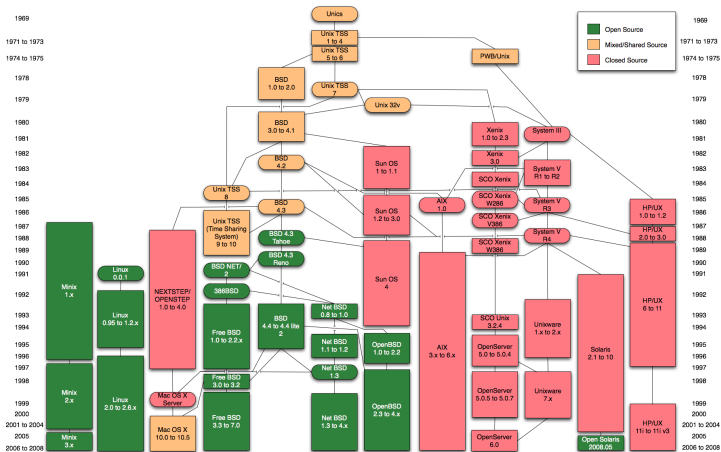
	Win NT	Darwin	Linux	Symbian	FreeBSD	Win CE
x86	✓	✓	✓	✓	✓	✓
alpha	✗	✗	✓	✗	✓	✗
arm	✓	✓	✓	✓	✓	✓
ia64	✓	✗	✓	✗	✓	✗
m68k	✗	✗	✓	✗	✗	✗
mips	✗	✗	✓	✗	✓	✓
parisc	✗	✗	✓	✗	✗	✗
sparc	✗	✗	✓	✗	✗	✗
sparc64	✗	✗	✓	✗	✓	✗
powerpc	✗	✓	✓	✗	✓	✗
x86_64	✓	✓	✓	✗	✓	✗

Sistemas Operativos

Nombre	Creador	Basado en	Licencia
FreeBSD	Comunidad	386BSD	BSD
OpenBSD	Comunidad	386BSD	BSD
NetBSD	Comunidad	386BSD	BSD
GNU/Linux	Comunidad	Unix	GPL
Windows	Microsoft		Privativa
Mac OS X	Apple		Privativa
OpenSolaris	Sun	Solaris	CDDL
Windows server	Microsoft		Privativa
Android	Google	GNU/Linux	Libre
iOS	Apple	Mac OS X	Privativa
z/OS	IBM	OS/390	Privativa

Fuente: http://en.wikipedia.org/wiki/Comparison_of_operating_systems

Historia de UNIX



Fuente: http://upload.wikimedia.org/wikipedia/commons/5/50/Unix_history-simple.png

La lucha de los dispositivos móviles

- Microprocesadores

- ARM
- Intel
- ...

- Dispositivos

- mobile phone
- tablet
- smartphone
- smartbook
- miniconsolas
- ...

- Sistemas operativos

- ~~Blackberry OS~~
- ~~Symbian~~
- Windows Mobile, Windows
- iOS
- Android
- ~~Firefox OS~~
- ~~Ubuntu~~
- ~~MeeGo/Maemo/Moblin~~
- ~~webOS~~
- LineageOS (CyanogenMOD)
- Tizen
- AOSP Extended
- ...

http://en.wikipedia.org/wiki/Mobile_operating_system



El sorpasso

Windows parece que ya no es el principal sistema operativo en los clientes

- statcounter 2009-2018
- Wikipedia - Usage share operating systems
- Ubuntu Bug nº1

Los grandes: servidores, workstations, mainframes, ...

- Microprocesadores

- x86_64
- ...
- powerpc
- SPARC
- m68000
- Alpha
- HP-PA-RISC
- z/Architecture

- Empresas

- HPE
- Dell
- IBM Lenovo
- IBM
- Fujitsu

- Sistemas operativos

- Red Hat Enterprise Linux
- Ubuntu Server
- Windows Server
- Solaris
- Debian GNU/Linux
- HP-UX
- AIX
- FreeBSD
- OpenBSD
- ...
- z/OS



Linux ganó

El kernel linux es el más utilizado en los diferentes sistemas operativos de **todos** los segmentos de ordenadores hoy en día:

- Supercomputadoras top 500
- Servidores en centros de datos
- Equipos clientes (android)
- Sistemas empujados

Aunque eso no significa siempre que gane el software libre :-/



El kernel linux

- Características principales
- Características de la compilación
- Carga del sistema
- Manejo de módulos
- Compilación de módulos
- Compilación del kernel

Características principales

- Kernel del sistema GNU/Linux, licenciado bajo la GNU GPL
- Desarrollo colaborativo de miles de personas
- Monolítico
- LKM: Loadable kernel module
- Última versión estable: 4.18.9 (23/9/2018)
- `kernel.org`
- Portado a gran cantidad de arquitecturas, desde pequeños dispositivos a grandes supercomputadoras.



Características de la compilación

- El código fuente de rama vanilla del núcleo ocupa actualmente 886MiB
- Los componentes del kernel se compilan de dos formas:
 - Se incluyen dentro de un fichero ejecutable enlazado estáticamente y que habitualmente se denomina vmlinuz o zImage
 - Se compilan individualmente en ficheros objetos con extensión .ko que se cargan en memoria a demanda (están ubicados en /lib/modules)
- Soluciones para hardware no detectado en el arranque:
 - Se aumenta el tamaño del fichero ejecutable (bzImage o big zImage)
 - Se montan temporalmente algunos módulos en memoria (initramfs)
- Distribuciones de uso general en sistemas x86:
 - Enorme variedad de hardware
 - Se incluyen gran cantidad de módulos
- Es posible compilar un kernel para un hardware determinado y reducir mucho su tamaño.

Manejo de módulos

La mayoría de los módulos se cargan automáticamente cuando es necesario, pero es posible cargarlos o descargarlos manualmente:

- `lsmod`: Módulos cargados
- `modprobe <módulo>`: Carga el módulo en memoria
- `modprobe -r <módulo>`: Descarga el módulo de la memoria
- `find /lib/modules/$(uname -r) -type f -iname '*.ko' :`
Módulos disponibles
- `modinfo <módulo>`: Información del módulo
- `depmod`: Actualiza las dependencias de los módulos

Compilación de módulos (Debian)

En algunas ocasiones es necesario compilar un módulo, normalmente algún controlador de dispositivo que no se encuentra soportado en la rama oficial del kernel.

- Instalamos los paquetes necesarios para compilar:

```
# aptitude install build-essential
```

- Instalamos los ficheros de cabeceras del kernel actual:

```
# aptitude install linux-headers-$(uname -r)
```

- Si existe un paquete Debian con el módulo sin compilar (ndiswrapper, fuse, madwifi, ...) ⇒ Utilizar module-assistant
- En el resto de casos, se descomprime el paquete que incluye los ficheros fuentes del módulo en /usr/src y se siguen las instrucciones del fichero README que debe incluir.

Compilación del kernel (Debian)

Es poco habitual tener que compilar un núcleo completo, puede ser necesario cuando se utiliza hardware muy peculiar o como en este caso simplemente con fines educativos.