



Introducción

LIN - Curso 2015-2016



J.C. Sáez, M. Prieto
High Performance Computing Division
ArTeCs Group
Universidad Complutense de Madrid

Contenido



- 1** Historia de Unix/Linux
- 2** Diseño y estructura del SO
- 3** Referencias



Contenido



1 Historia de Unix/Linux

2 Diseño y estructura del SO

3 Referencias



Historia de Unix (I)

- Multics – Multiplexed Information and Computing Service (1965)
- Unics – Unix (1969)
 - Creado por Ken Thompson y portado a “C” (Dennis Ritchie) en 1973
 - V6 (1975): Código fuente público (licencia AT&T)
 - Distribuciones BSD (Billy Joy)
 - Libro de John Lions sobre UNIX V6



Claves del éxito

- 1 Licencia económica
- 2 Código fuente disponible
- 3 Código simple y relativamente fácil de modificar
- 4 Requiere pocos recursos HW



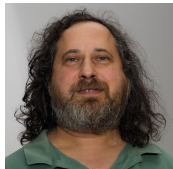
Historia de Unix (II)

- Unix (Cont.)
 - V7 (1979): AT&T prohíbe que se estudie su código
- Xenix (1980)
 - Microsoft
 - SCO
- Unix System III (1982)
- Unix System V (1983)
 - HP-UX, IBM's AIX, Sun's Solaris



Historia de Unix (III)

- Proyecto GNU (1983) - Richard Stallman
 - SO GNU: Emacs, GNU compiler collection (GCC), GNU Hurd (kernel)
- X/Open (1984)
 - Bull, Ollivetti, Philips,...
 - Sistemas Abiertos
- Minix v1 (1987) - Andrew Tanenbaum
 - Minimal Unix-like OS (Clon de Unix)
 - Fines docentes. Estructura más modular
 - Unix/Linux → eficiencia \neq Minix → diseño comprensible
 - Compatible con Unix V7 (nivel usuario), evolucionó posteriormente hacia el estándar POSIX
 - 1987 (Minix1 - i8088), 1997 (Minix2 - i386)



Richard Stallman



Andrew Tanenbaum



Llamadas al Sistema Minix

■ 53 llamadas: 6 Categorías + Especiales

■ Gestión de procesos

- `fork`, `waitpid`, `exec`, `exit`, `brk`, `getpid`, `getpgrp`, `setsid`, `ptrace`

■ Señales

- `sigaction`, `sigreturn`, `sigprocmask`, `sigpending`, `sigsuspend`, `kill`, `alarm`, `pause`

■ Gestión de Ficheros

- `creat`, `mknod`, `open`, `close`, `read`, `write`, `lseek`, `stat`, `fstat`, `dup`, `pipe`, `ioctl`, `access`, `rename`, `fcntl`

■ Gestión Dir y Sistemas de Ficheros

- `mkdir`, `rmdir`, `link`, `unlink`, `mount`, `umount`, `sync`, `chdir`, `chroot`

■ Protección

- `chmod`, `getuid`, `getgid`, `setuid`, `setgid`, `chown`, `umask`

■ Gestión de Tiempos

- `time`, `stime`, `utime`, `times`





Unix Wars (1987-1996)

■ Unix International:

- Unix System V Release 4 (SVR4) - USL (AT&T) + Sun + SCO
 - SVR3, BSD, Sun OS, Xenix
- Open Look

■ Open Software Foundation:

- OSF/1 (DEC, IBM, HP, Bull,...)
 - OSF/1 (Mach 2.5)
- Motif



Historia de Unix (V)

- Net-1 (1989), Net-2 (1991) y 386BSD (1992)
 - Hasta 4.3BSD--Tahoe, BSD no era una distribución libre
 - Necesario licencia de las fuentes de Unix (AT&T)
 - Net-2: casi todo el kernel y todas las utilidades
 - 386BSD Bill Jolitz (faltaban 6 ficheros)
- Litigio entre USL y BSDI / Universidad de California Berkeley (1991-1994)
 - Novell adquiere USL y su propiedad intelectual en 1993 y se llega a acuerdo en 1994
- 4.4BSD-Lite (1994)
 - FreeBSD, NetBSD, OpenBSD, Darwin





Y Aparece Linux... (1991)

- Unos meses después de liberarse Minix 1 se crea en **comp.os.minix**

From: torva...@klaava.Helsinki.FI (Linus Benedict Torvalds)
Date: 25 Aug 91 20:57:08 GMT Local: Sun, Aug 25 1991 9:57 pm
Subject: What would you like to see most in minix?

Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :)

Linus (torva...@kruuna.helsinki.fi)

PS. Yes it's free of any minix code and it has a multi-threaded fs. It is NOT portable (uses 386 task switching, etc), and it probably never support anything other than AT-harddisks, as that's all I have :(.



Y Aparece Linux... (1991)

■ Claves Éxito

- Unix Wars
- Litigio BSD
- Competencia Windows NT (enemigo común)
- Licencia GPL
- Internet
- ???





Y Aparece Linux... (1991)

- Pero Linux es sólo el kernel...
 - GNU/Linux..., IBM/RedHat/HP/....Linux..
- Distros (kernel + selección de herramientas precompiladas)
 - MCC Interim Linux 1992,
 - Slackware (Patrick Volkerding) , Debian (Ian Murdock) 1993
 - S.U.S.E, Red Hat (Marc Erwing, Bob Young) 1994
- Entornos de Escritorio
 - Xfree86 – Thomas Roel 1991
 - KDE – Matthias Ettrich 1996
 - Gnome – Miguel de Icaza 1997
 - ...





¿Qué es Unix?

- Fin de Unix Wars 1996 (Competencia con Windows NT)
 - 1994: Novell Trasfiere derechos de marca registrada Unix a X/Open, que crea la **Single Unix Specification (SUS)**
 - 1994: Unix international y OSF se fusiona en nueva OSF
 - 1996: OSF y X/Open se fusiona en **Open Group**
- Open Group
 - Propietario actual marca registrada UNIX
 - Certificación SUS
- Unificación SUS / IEEE Posix en 2001 (SUS Version 3)
- *Unix es quien se comporta como Unix*
 - Mac OS X Leopard (primer derivado de BSD que puede llamarse Unix)
 - Z/OS IBM



Contenido



1 Historia de Unix/Linux

2 Diseño y estructura del SO

3 Referencias



Diseño del Kernel

A beast of a different nature

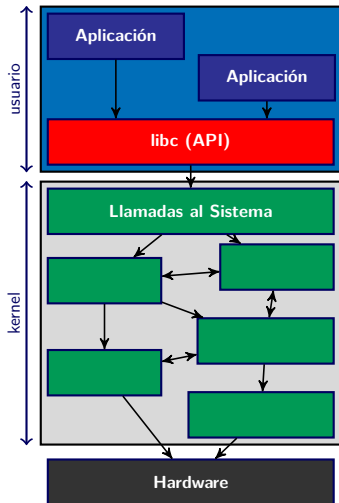
- Tanto rendimiento como portabilidad son aspectos sumamente críticos
- No pueden utilizarse muchas de las abstracciones utilizadas en las aplicaciones de usuario
 - Sin libc*
 - Sin protección de memoria – no hay SIGSEGV –
 - Espacio de pila limitado y de tamaño fijo
 - Métodos de depuración menos elaborados ...
- Se han explorado diferentes estructuras
 - 1 Monolítico
 - 2 Microkernel (Cliente/Servidor)
 - 3 Máquinas Virtuales
 - 4 Exokernel

Estructura Monolítica (I)

Estructura Monolítica

- Todo el kernel reside en un único espacio de direcciones
 - Se distinguen distintos componentes (hay una estructura, depende del SO)
 - ... pero la “comunicación” entre componentes es por invocación directa de las funciones (no hay ocultación, con los riesgos que eso tiene asociado)
 - Convencional: el *bootloader* carga la imagen del kernel en memoria a partir de un único binario
- Diseño tradicional: simple y buen rendimiento
 - Kernels Unix tradicionales (SVR4), derivados (SunOS, AIX) y clones (familia BSD, Linux)
 - DOS, Windows 9x

Estructura Monolítica (II)



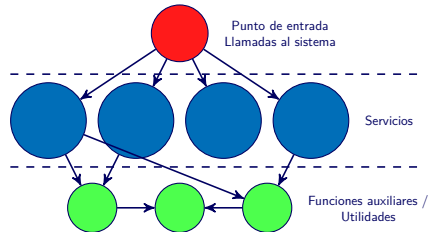
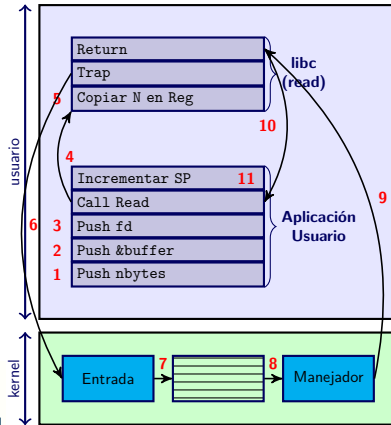
Componentes del kernel



Estructura Monolítica (III)

■ Estructura funcional – Tanenbaum (Sección 1.5.1)

■ `count = read (fd, buffer, nbytes)`



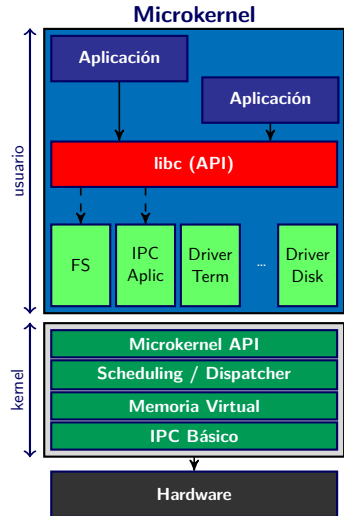
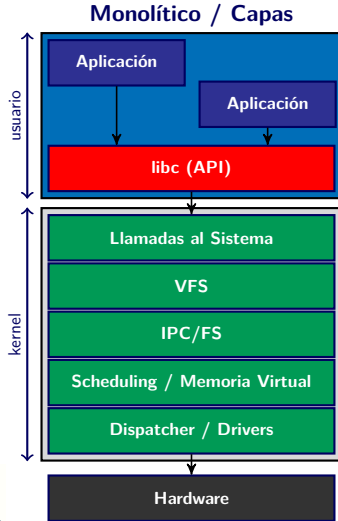


Microkernel / Cliente Servidor (I)

- Sistema operativo = Microkernel + Servidores
 - Kernel ligeros (micro-kernel) con funcionalidad mínima
 - Multiprogramación: Scheduling + Dispatcher (Cambio de Contexto)
 - Memoria Virtual
 - Mecanismos IPC básicos (mensajes)
 - Procesos servidores que ofrecen los servicios tradicionales a los proc. Usuario
 - Idealmente se ejecutan en espacio de usuario



Microkernel / Cliente Servidor (II)





Microkernel / Cliente Servidor (III)

■ Potenciales Ventajas

- Mayor modularidad
 - Potenciales ventajas asociadas: Portabilidad, Extensibilidad, Fiabilidad
- Mayor protección/seguridad
 - idealmente sólo el micro-kernel modo supervisor
- Mayor tolerancia a fallos
 - Si falla un servidor/driver no tiene porque fallar todo el sistema
- Se adaptan de forma natural a sistemas distribuidos
 - Comunicación entre procesos explícita a través de la red
- Facilidad de desarrollo
 - Los servidores se pueden depurar





Microkernel / Cliente Servidor (IV)

■ Limitaciones

- Costes asociados con los mecanismos IPC limitan la aplicabilidad Micro-kernels puros

■ Implementaciones reales (primera generación)

- Los kernels de Windows NT y Mach (MAC OS X) utilizan diseño Micro-kernel
- En sus últimas versiones ninguno de los servidores corren en espacio de usuario
- La comunicación entre servidores es por invocación directa de funciones.

■ Nuevas Generaciones

- Familia L4: http://en.wikipedia.org/wiki/L4_microkernel



Contenido



1 Historia de Unix/Linux

2 Diseño y estructura del SO

3 Referencias





Historia de Unix/Linux

- Aspectos más relevantes de los 40 años de Historia de Unix en ComputerWorld.
- Referencia clásica sobre OpenSource. Entre otros, se incluye una extensa descripción de los orígenes y evolución de Unix, BSD, el proyecto GNU y Linux
- Multics: The Multiplexed Information and Computing Service
- Descripción de las primeras versiones de Unix por Dennis M. Ritchie
- Entrevista a Steve Bourne, creador del popular bourne shell (sh)
- Entrevista a Dennis Ritchie en IEEE Spectrum

Referencias (II)



Arquitectura del SO

- Sección 1.5 de *Operating Systems Design and Implementation*. A.S. Tanenbaum y A.S. Woodhull.
- Exokernel: An Operating System Architecture for Application-Level Resource Management
- The Multikernel: A new OS architecture for scalable multicore systems





LIN - Introducción Versión 0.2

©J.C. Sáez, M. Prieto

*This work is licensed under the Creative Commons **Attribution-Share Alike 3.0 Spain License**. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/es/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.*

*Esta obra está bajo una licencia **Reconocimiento-Compartir Bajo La Misma Licencia 3.0 España de Creative Commons**. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.*

Este documento (o uno muy similar) está disponible en <https://cv4.ucm.es/moodle/course/view.php?id=62472>

