

Presentación

AISO - Introducción



Objetivos

- Profundizar los conceptos adquiridos en las asignaturas de
 - Sistemas operativos
 - Laboratorio de sistemas operativos
- Especial hincapié en las interacciones entre hardware y el software



Programa

- Tema 1: Introducción.
 - Historia Unix/Linux
 - Estructura SO
- Tema 2: Procesos
 - Arranque del sistema
- Tema 3: I/O
- Tema 4: Gestión de memoria
- Tema 5: Sistema de Ficheros



Bibliografía

- A.S. Tanenbaum and A.S. Woodhull.; Operating Systems Design and Implementation. 3rd Edition; Prentice Hall, 2006. ISBN 0131429388;
- Robert Love; Linux Kernel Development. 2nd Edition; Novell Press, 2005, ISBN: 0672327201;



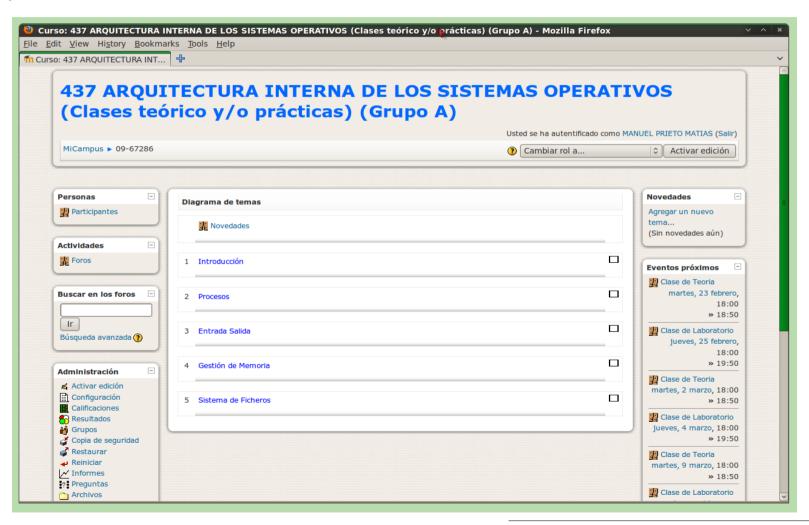
Organización de la Asignatura

- Método docente:
 - Enseñanza presencial teórica
 - Realización de prácticas individuales
 - Realización de Trabajos
- Exámenes:
 - Examen final en junio y septiembre
- Método de evaluación:
 - Examen
 - Evaluación Continua (exposiciones publicas y trabajos)



Información adicional

Campus Virtual (moodle)





Historia de Unix (I)

- Multics Multiplexed Information and Computing Service (1965)
- Unics Unix (1969)
 - V6 : Código fuente público (licencia AT&T)
 - Distribuciones BSD (Billy Joy) y Libro de John Lions
 - V7 : AT&T prohibe que se estudie su código
- Xenix (1980)
 - SCO
- Unix System III (1982)
- Unix System V (1983)
 - HP-UX, IBM' AIX, Solaris



Historia de Unix (II)

- Proyecto GNU 1983
 - Emacs, Gnu compiler collection, Gnu Hurd
- X/Open (1984)
 - Sistemas Abiertos
- Minix v1 (1987)
 - Minimal Unix-like Os
 - Fines docentes. Estructura más modular
 - Unix/Linux eficiencia <> Minix comprensible
 - Compatible con Unix V7 (nivel usuario), evolucionó posteriormente hacia el estándar POSIX



- 1987 (Minix1 i8088)
- 1997 (Minix2 i386)

Llamadas al Sistema Minix

- 53 llamadas: 6 Categorias + Especiales
 - Gestion de procesos
 - fork, waitpid, exec, exit, brk, getpid, getgprg, setsid, ptrace
 - Señales
 - sigaction, sigreturn, sigprocmask, sigpending, sigsuspend, kill, alarm, pause
 - Gestión de Ficheros
 - creat, mknod, open, close, read, write, Iseek, 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



Historia de Unix (III)

- Unix Wars (1987-1996)
 - Unix International:
 - Unix System V Release 4 (SVR4) USL + Sun
 - 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 (IV)

- 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 con 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



Y Aparece Linux... (1991)

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

From: torva...@klaava.Helsinki.Fl (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 protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.



Y Aparece Linux... (1991)

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



Y Aparece Linux... (1991)

- Pero Linux no es solo el Kernel (GNU/Linux... GNU/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



Que es Unix?

- Fin de Unix Wars 1996 (Competencia con Windows NT)
 - 1994: Novel 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



Que es Unix?

- Unificación SUS / Posix en 2001 (SUS Version 3)
- Unix es quien se comporta como Unix:
 - Mac OS X Leopard (primer derivado BSD que puede llamarse Unix)
 - Z/OS IBM



Diseño del Kernel

A beast if 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
 - Monolítico
 - Microkernel (Cliente/Servidor)
 - Máquinas Virtuales



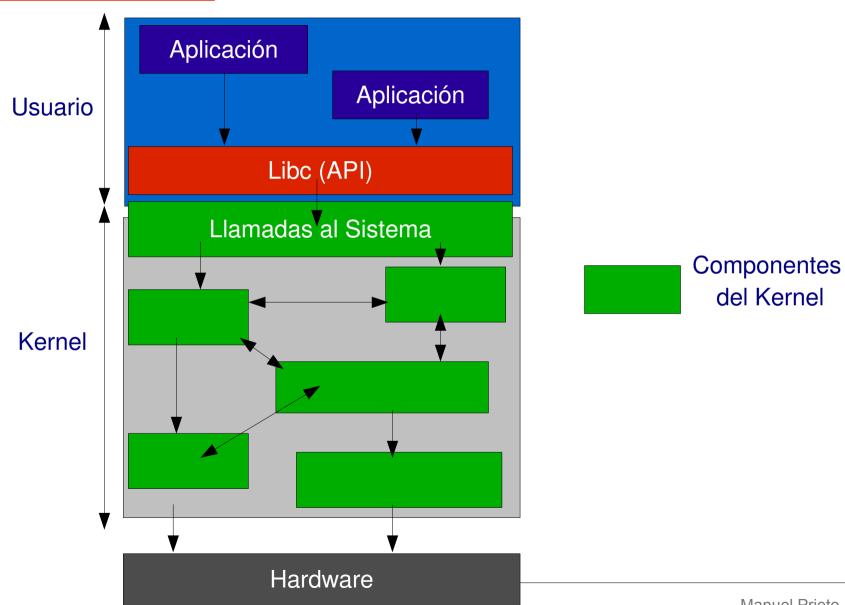
Estructura Monolítica (I)

Estructura Monolítica

- Diseño tradicional: simple y buen rendimiento
 - Kernels Unix tradicionales (SVR4), derivados (SunOS, AIX) y clones (familia BSD, Linux)
 - DOS, Windowx 9x
- Todo el código del kernel reside en un único espacio de direcciones
 - Se distinguen distintos componentes (hay una estructura, depende del OS)
 - ... pero la "comunicación" entre los módulos 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



Estructura Monolítica (II)

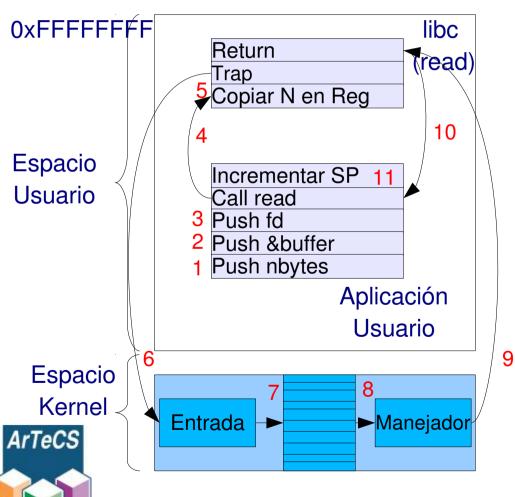


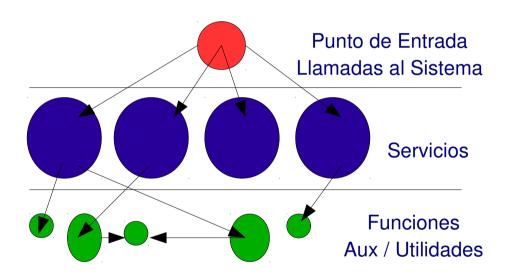


Estructura Monolítica (III)

Estructura funcional – Tanembaum (Seccion 1.5.1)

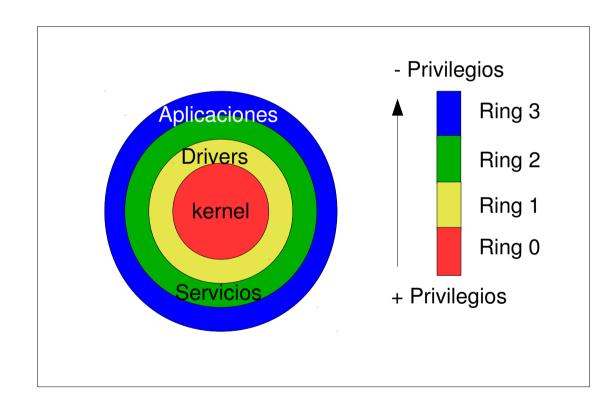
count = read (fd, buffer, nbytes)





Estructura Capas (I)

- Capas/Anillos (Generalización)
 - THE (Dijkstra 1968)
 - Multics
 - Soportado por Hw (i386)





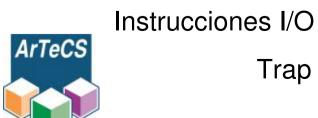
Maquinas Virtuales (I)

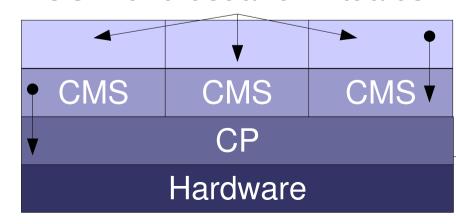
- Origen: IBM System/360 Operating System
 - OS tipo batch para las mainframes System/360 (1964)
 - Extensión oficial a tiempo compartido TSS/360 (1967)
 - Diseño convencional: el kernel gestiona la memoria y el resto de recursos Hw para que puedan compartirse por varios usuarios:
 - División memoria
 - Multiplexación en el tiempo del procesador
 - ...
 - Mismos problemas que Multics (competencia):
 - Alto coste desarrollo (50 millones \$)
 - Kernel muy pesado y con rendimiento pobre



Maquinas Virtuales (II)

- Proyecto CP/CMS (1967)
 - Tiempo compartido = multiprogramación + maquina extendida
 - CP (Control Program) Virtual Machine Monitor (hypervisor)
 - Proporciona/soporta multiprogramación entre varias maquinas virtuales
 - <u>Virtualización total</u>: cada VM es una copia exacta del HW (I/O)
 - CMS (Conversational/Cambridge Monitor System) guest OS
 - Sistema operativo <u>ligero</u> mono-usuario (interactivo)
 - Proporciona ilusión de maquina extendida propia a los usuarios
 OS Mono-usuario Virtuales



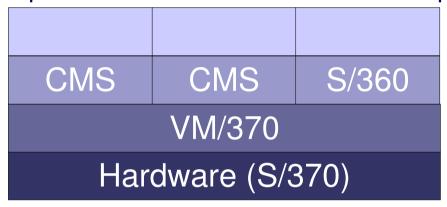


Llamada Sistema

Trap

Maquinas Virtuales (III)

- Evolución CP/CMS VM/370 (1972)
 - Otras ventajas: portabilidad software desarrollado para la serie S/360



- S/370(1972) S/390(1990) eServer Zseries(2000) IBM System Z (2006) ...
- CP/CMS se distribuyo con su código fuente
 - Uno de los primeros sistemas operativo de código abierto
 - IBM no daba soporte
 - Los usuarios (comunidad) contribuyeron al desarrollo/soporte del sistema

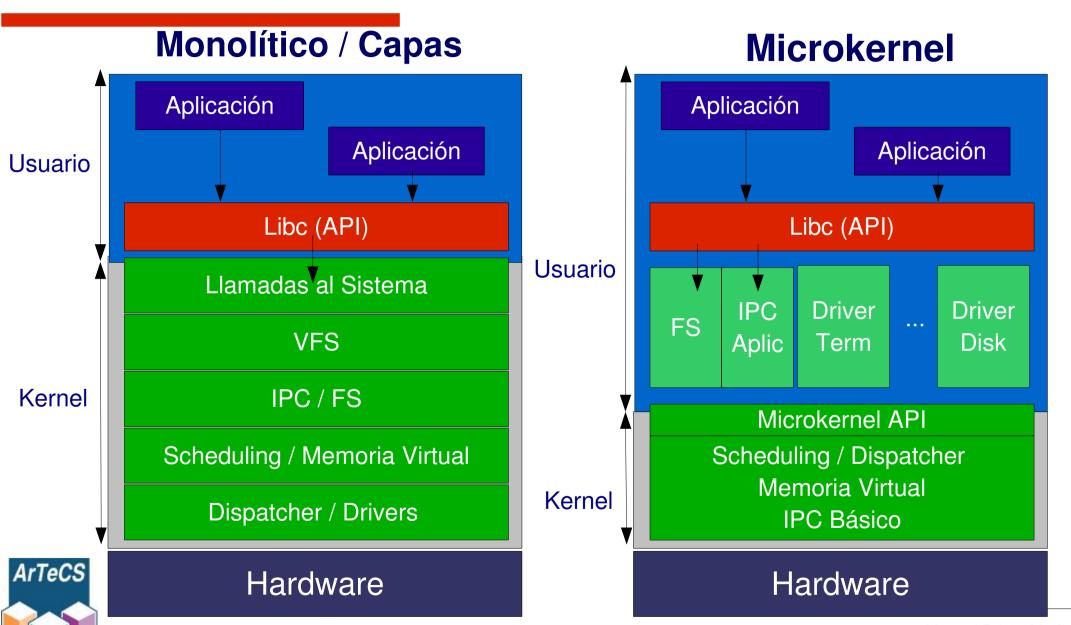


Microkernel / Cliente Servidor (I)

- 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

- 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.



Exokernel (I)

- Exokernel: concepto de diseño introducido en 1994 (MIT)
 - Exokernel ~ Hypervisor VM + Asignación explícita de recursos
 - Idea: permitir a los programadores usar sus propias abstracciones
 - El exokernel se limita a:
 - Multiplexación de recursos
 - Asegurar protección en el uso de Hw
 - A cada guest (libOS) se le pueden asignar recursos.
 - En maquinas virtuales todos los guest ven el mismo Hw virtual
- No existen versiones comerciales
- ArTeCs Interés en many-core

AISO Introducción Versión 0.1

© Manuel Prieto Matias

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 envie una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Este documento (o uno muy similar) esta disponible en https://cv2.sim.ucm.es/moodle/course/view.php?id=3235



