

Sistemas Operativos

Introducción

Matías Zabaljáuregui
matiasz@info.unlp.edu.ar

Por qué estudiar sistemas operativos ?

- comprender mejor la arquitectura de hardware de una computadora
- entender el software de bajo nivel que soporta a nuestros programas
- estudiar el diseño, arquitectura y modelo de desarrollo de proyectos de software de varios millones de líneas de código
- incorporar conceptos de performance generalmente ignorados en la programación de alto nivel
- formación necesaria para cualquier programador de back-end, system programmer,
- formación complementaria para los interesados en seguridad, networking, sys admins, integradores, compiladores, HPC, etc
- información que puede ser útil para el resto: alocaión/alineación RAM, concurrencia/sincronización, SMP/CMP/multicore/manycore, filesystems específicos para DBMS, etc

Por qué Linux en la academia?

- El código es libre
- Hay mucha documentación
- El sistema más usado en entornos científicos (por ejemplo CERN, HPC, GRID, CLUSTERS, REDES)
- Es el más usado por investigadores en sistemas operativos (Minding the Gap: R&D in the Linux Kernel. IBM/Princeton. ACM Operating Systems 2008)

24 directorios de arquitecturas, con sus variantes:

alpha: Hewlett-Packard's Alpha workstations

arm, arm26: ARM processor-based computers such as PDAs and embedded devices

cris: "Code Reduced Instruction Set" CPUs used by Axis in its thin-servers, such as web cameras or development boards

frv: Embedded systems based on microprocessors of the Fujitsu's FR-V family

h8300: Hitachi h8/300 and h8S RISC 8/16-bit microprocessors

i386 : IBM-compatible personal computers based on 80x86 microprocessors

ia64: Workstations based on the Intel 64-bit Itanium microprocessor

m32r: Computers based on the Renesas M32R family of microprocessors

m68k, m68knommu: Personal computers based on Motorola MC680x0 microprocessors

mips: Workstations based on MIPS microprocessors, such as those marketed by Silicon Graphics

parisc: Workstations based on Hewlett Packard HP 9000 PA-RISC microprocessors

ppc, ppc64: Workstations based on the 32-bit and 64-bit Motorola-IBM PowerPC microprocessors

s390: IBM ESA/390 and zSeries mainframes

sh, sh64: Embedded systems based on SuperH microprocessors developed by Hitachi and STMicroelectronics

sparc, sparc64: Workstations based on Sun Microsystems SPARC and 64-bit Ultra SPARC microprocessors

um: User Mode Linux, a virtual platform that allows developers to run a kernel in User Mode

v850: NEC V850 microcontrollers that incorporate a 32-bit RISC core based on the Harvard architecture

x86_64: Workstations based on the AMD's 64-bit microprocessors such as Athlon and Opteron and Intel's ia32e/EM64T 64-bit microprocessors

...HPC, embebidos/RT, dispositivos, virtualización.

- Linux soporta todas las tecnologías de procesamiento paralelo: desde hyperthreading y procesadores multicore a grandes máquinas SMP y cc-NUMA. (Informe). Mayoría de supercomputadoras del top500 corren el kernel Linux (top500.org)
- En el mercado embebidos, el uso de Linux viene creciendo desde hace tiempo (android, tablets, industrial) (encuesta anual *linux devices*)
- El proyecto de drivers está ocioso porque no parecen faltar drivers para dispositivos que tengan liberadas las especificaciones (linux drivers project)
- Desktop: Crecimiento del 82 % entre 2007 y 2008. Aunque en términos absolutos todavía es muy poco (2%).
http://news.cnet.com/8301-13505_3-9910263-16.html
- Se proyecta como la base de las soluciones de virtualización

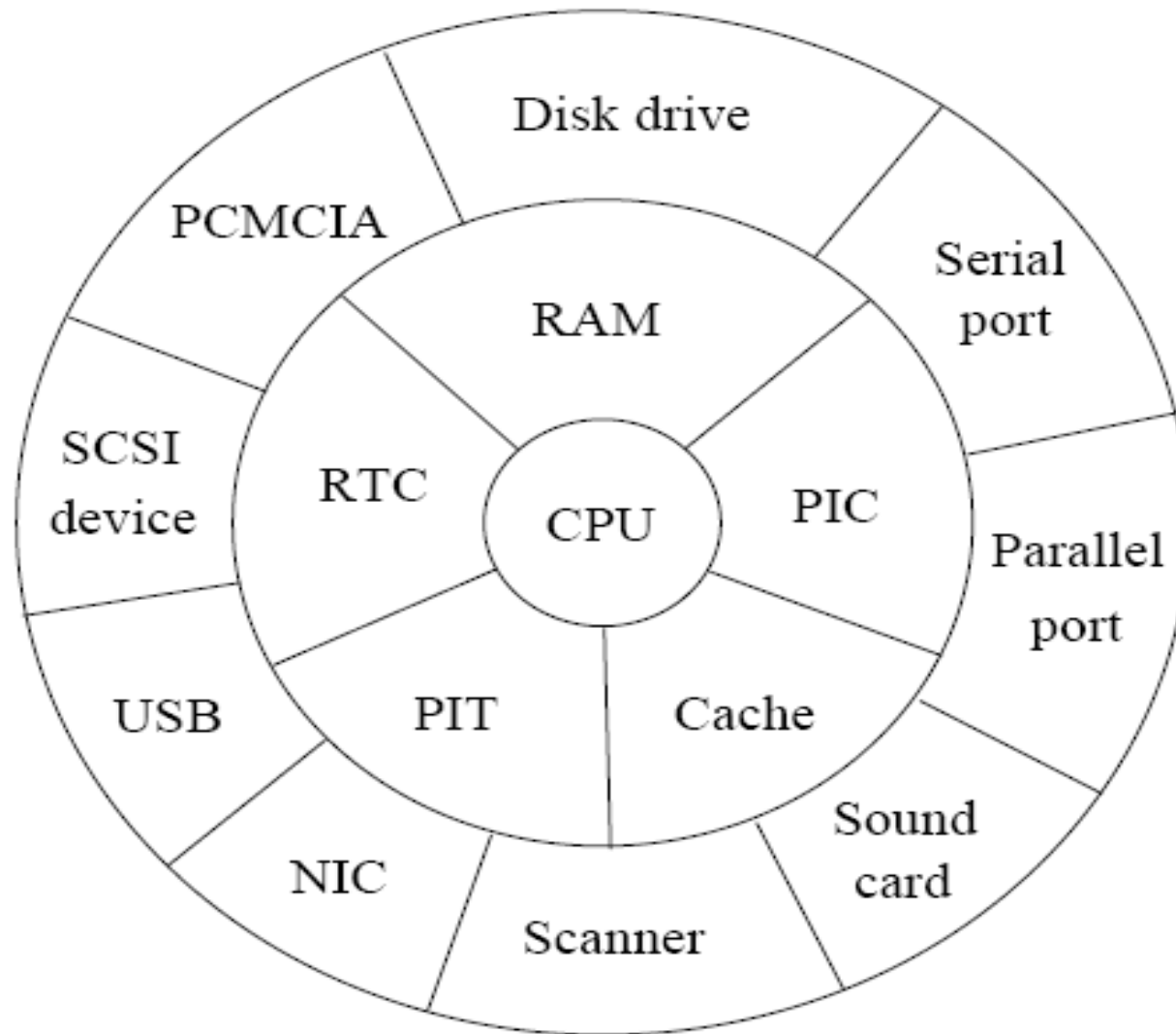
Objetivos

- Profundizar y relacionar los temas vistos en materias de arquitecturas y sistemas operativos
- Comprender como se construyen las abstracciones de software sobre el hardware, tanto a nivel diseño como implementación
- Asumimos “hardware moderno”: SMP, CMP y multicore, APIC, NICs rápidas, soporte de virtualización, etc
- Nos acercamos al código del kernel Linux, vemos algunas APIs como referencia, estudiamos algunas implementaciones de ejemplo, etc
- Posibles trabajos de investigación en lugar de examen final tradicional

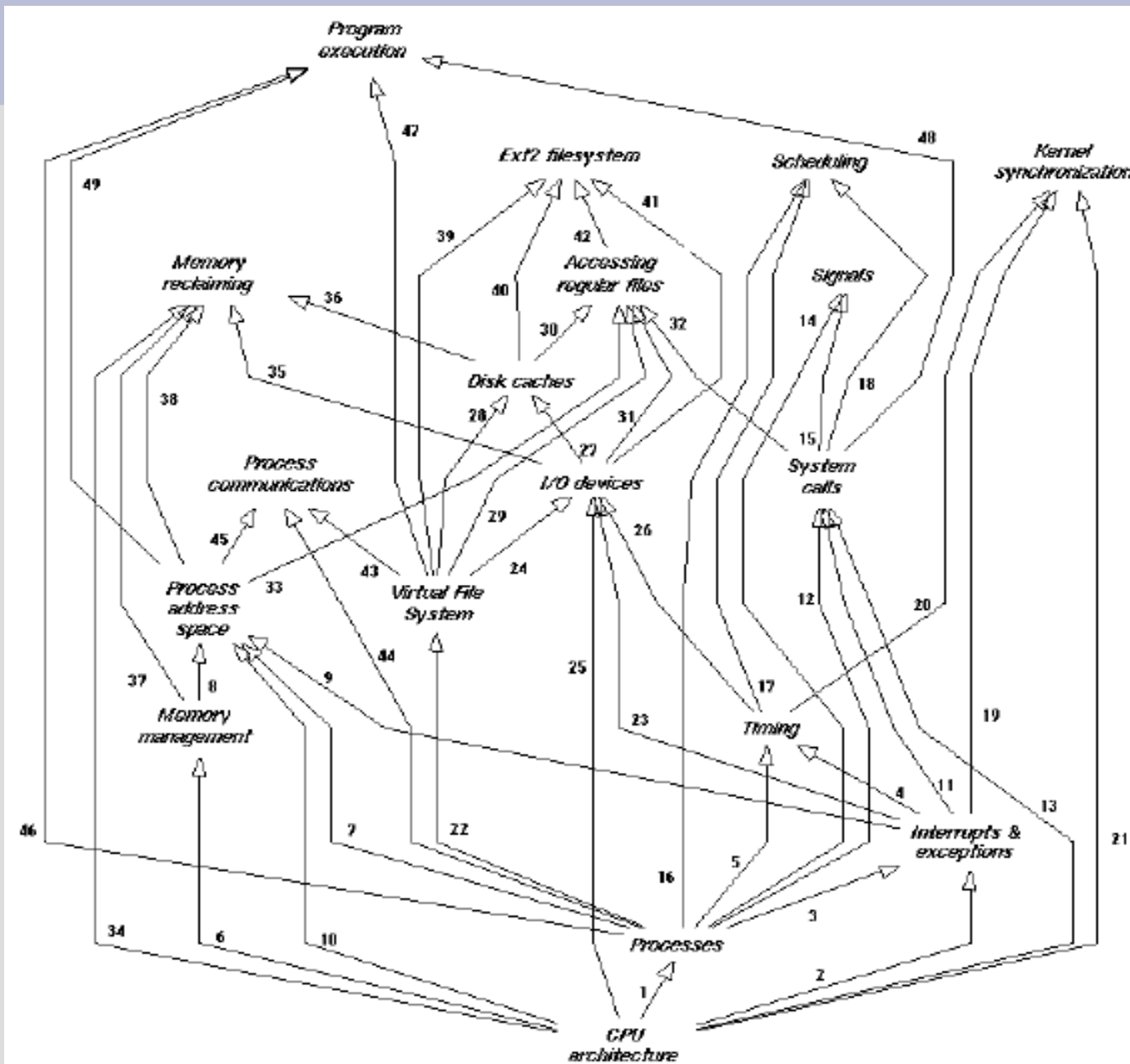
Nuestra aproximación

- Estudio del kernel de un sistema operativo UNIX moderno y usado en producción.
 - “A Real Bottom-Up Operating Systems Course”, Daniel Bovet & Marco Cesati
- Metodología que estamos compartiendo con algunos profesores de Sistemas Operativos de la UBA
- Muchas (de las más importantes) universidades del exterior plantean sus materias de Sistemas Operativos de esta forma

Hardware de una PC



Dependencias entre temas:



Nuestro orden:

“ordered in such a way to minimize the forward references to arguments yet to be explained”

- Segmentación y Paginación
- Procesos
- Interrupciones
- Sincronización
- Tiempo
- Schedulling
- Manejo de Memoria
- Espacio de direccionamiento de procesos

Cuestiones...

- Sería bueno lleven los temas al día
- Un rato después de clase para consultas
- Usemos los foros y si necesitan, hacemos sesiones de chat
- Los primeros 20 minutos de clase tendrán un cuestionario multiple-choice del tema anterior.
- Aprobando todos más un trabajo final (difícil) pueden promocionar.

Bibliografía

- Understanding Linux Kernel 3rd edition
- Linux kernel development 2nd edition
- Manuales de Intel IA-32
- Tradicionales de sistemas operativos