

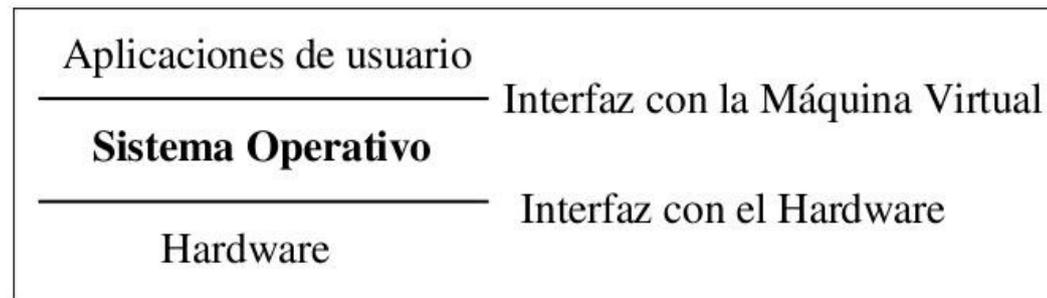


INTRODUCCIÓN



● ¿Qué es un sistema operativo?

- ◆ Es software que proporciona un **acceso sencillo y seguro** al hardware, **ocultando detalles** al usuario acerca de la implementación particular y creando la **ilusión de existencia de recursos ilimitados** (o abundantes).
- ◆ Es el de un programa que actúa como **intermediario** entre el usuario de la computadora y el hardware de la computadora.



- ◆ Es un conjunto de programas útiles para:
 - Permitir y facilitar el uso de sistemas de cómputo
 - Brinda un conjunto de servicios
 - Administrar recursos: CPUs, memoria, periféricos, interfaces red, etc.

Eficiencia
Seguridad
Equidad



● **Funciones básicas**

◆ Administración de recursos

Recursos de hardware y de redes de un sistema informático, como la CPU, memoria, dispositivos de almacenamiento secundario y periféricos de E/S.

◆ Administración de tareas/procesos

Los programas de administración de tareas de un sistema operativo administran la realización de las tareas de los usuarios finales. Controlan cuales tienen acceso al CPU y por cuánto tiempo.

◆ Administración de memoria

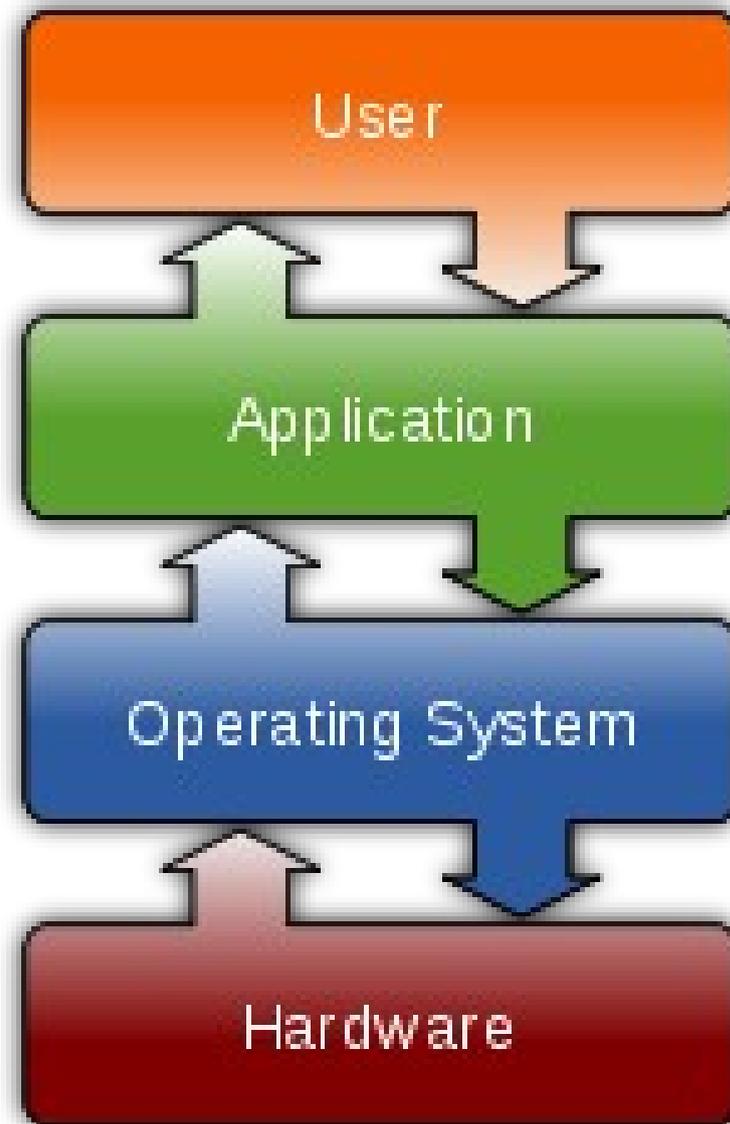
Administra toda la memoria que los procesos requieren.
Asignación, protección, virtualización.

◆ Administración de archivos

El Sistema Operativo abstrae las propiedades físicas del dispositivo de almacenamiento, proporcionando una unidad lógica del mismo.

Seguridad, Señales, etc.

● Esquema General



● **Modo usuario y modo supervisor**

- ◆ Las CPUs modernas soportan múltiples modos de operación y son controlados por el sistema operativo:

- **Modo usuario** (*userland*)

El código que se ejecuta no tiene la habilidad de acceder al hardware o referenciar memoria directamente.

Este código debe interactuar con una API para acceder a los recursos.

Debido a esta protección brindada por esta clase de aislamiento, las fallas en este modo siempre son recuperables.

De **The Jargon File** (http://es.wikipedia.org/wiki/Jargon_file)

userland: Anywhere outside the **kernel**. “That code belongs in userland.” This term has been in common use among Unix kernel hackers since at least 1985, and may have originated in that community. The earliest sighting was reported from the usenet group *net.unix-wizards*.



● Modo usuario y modo supervisor

- *Modo supervisor*

Modo de ejecución que permite la ejecución de todas las instrucciones brindadas por la CPU, donde es posible referenciar cualquier dirección de memoria y se tiene acceso al hardware subyacente.

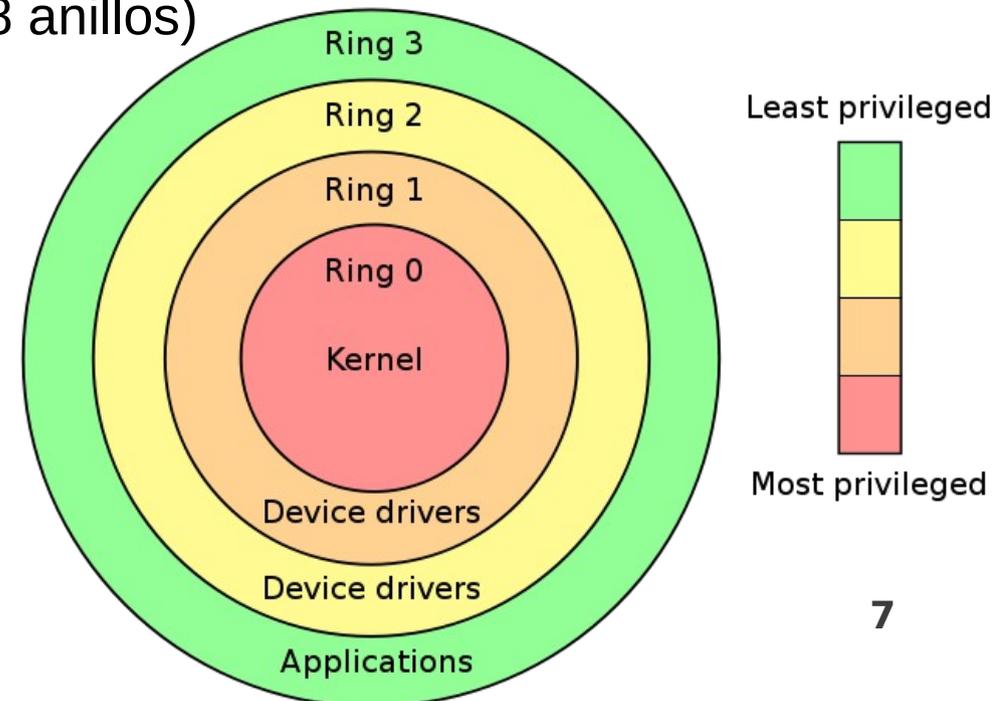
Este es el modo en el cual, generalmente, se ejecuta el sistema operativo.

Paso de un modo a otro

- interrupciones de software (trap, int, . . .)
- interrupciones de hardware (división por cero, dispositivos de E/S)

● Anillos de protección (Protection Rings)

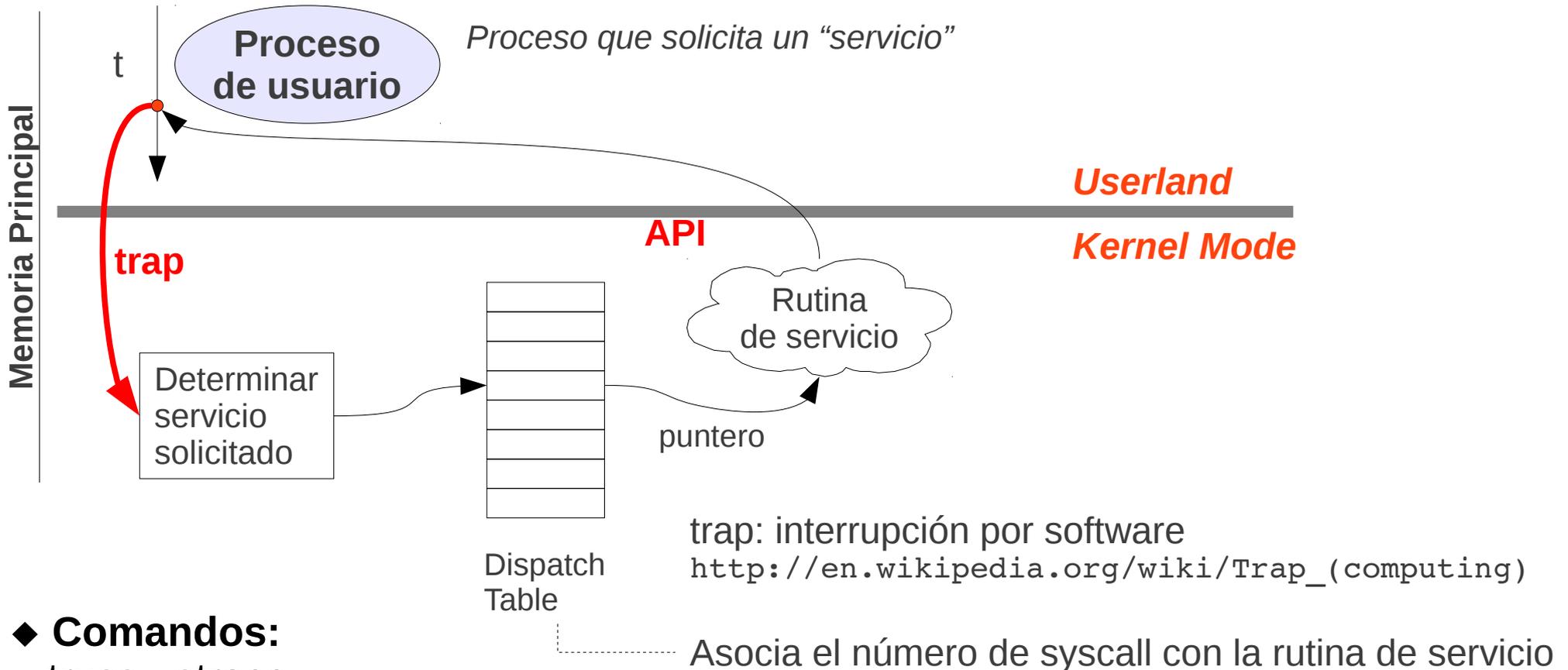
- ◆ También conocido como “*Dominios de protección jerárquicos*”
- ◆ Conforman un mecanismo para proteger datos y funcionalidades frente a fallas y comportamientos maliciosos
- ◆ Un anillo de protección es uno, dos o más niveles o capas jerárquicas de privilegios dentro de una arquitectura de un sistema, generalmente impuesto por la arquitectura de las CPUs
- ◆ El soporte de los anillos vía hardware fue uno de los conceptos más revolucionarios introducidos por **Multics**(8 anillos)
- ◆ **Ring 0**: nivel más privilegiado, es donde se interactúa más directamente con el hardware
- ◆ Se definen interfaces especiales entre los anillos para permitir a anillos externos acceder a recursos de anillos internos de manera predefinida



● Llamadas al sistema (*System Call - syscalls*)

- ◆ Conforman la interfaz mediante la cual las aplicaciones pueden acceder a los diferentes **servicios** que proporciona el Sistema Operativo
- ◆ Son implementadas por el kernel
- ◆ Permiten a las aplicaciones de espacio de usuario comunicarse con el hardware
- ◆ Comúnmente usan una instrucción especial de la CPU que causa que el procesador transfiera el control a un código privilegiado, previamente especificado por el mismo código.
- ◆ Cuando una syscall es invocada, la ejecución del programa que la invoca es interrumpida y sus datos son guardados para continuar ejecutándose luego. El procesador entonces comienza a ejecutar las instrucciones de código privilegiado, para realizar la tarea requerida. Cuando esta finaliza, se retorna al proceso original, y continúa su ejecución.

● Llamadas al sistema (*syscalls*)



◆ **Comandos:**
truss - strace

◆ **Material de lectura obligatoria:**
http://www.fceia.unr.edu.ar/~diegob/so/material/Llamadas_al_Sistema.pdf

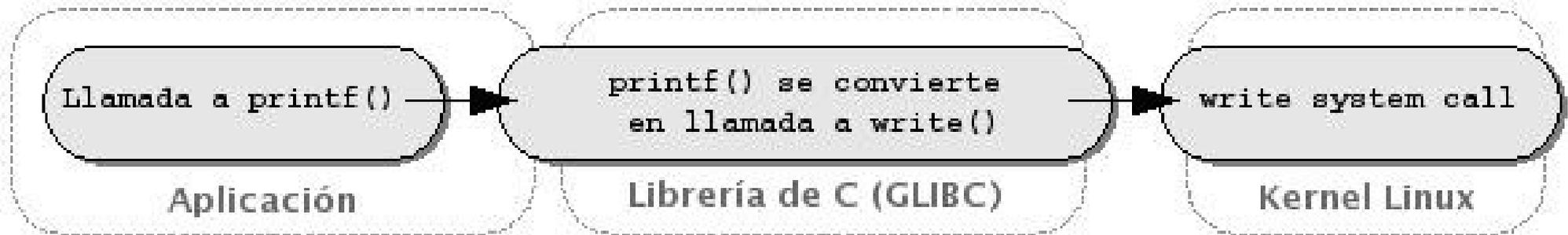
● Estándar POSIX

◆ Portable Operating System Interface [for Unix] (IEEE 1003)

Traducción aproximada: "Interfaz portable para Sistemas Operativos basados en UNIX"

- ◆ Las aplicaciones de usuario emiten syscalls a través de un API, en Linux lo proporciona la librería libc o glibc, basada en POSIX → *Linux es compliance POSIX*
- ◆ El estándar POSIX se refiere a las APIs, no a las syscalls, define un comportamiento, pero no como debe implementarse dicha API

◆ Relación: **Aplicación** ↔ **Librería** ↔ **syscall**



- Nombres de llamadas al sistema: `open`, `read`, `write`, `close`, `wait`, `exec`, `fork`, `exit`, `kill`, etc.
- El término POSIX fue sugerido por Richard Stallman en respuesta a la demanda de la IEEE, que buscaba un nombre fácil de recordar.

- Agregar una syscall a Linux:

www.kernel-labs.org/?q=syscalls

tldp.org/HOWTO/html_single/Implement-Sys-Call-Linux-2.6-i386/



● Características de los Sistemas Operativos

◆ *Tareas*

■ Monotarea:

Se ejecuta un proceso (aparte de los procesos del propio S.O.) en un momento dado. Una vez que empieza a ejecutar un proceso, continuará haciéndolo hasta su finalización y/o interrupción.

■ Multitarea:

Se ejecutan varios procesos al mismo tiempo.

Normalmente se asignan los recursos disponibles de forma alternada a los procesos que los solicitan, de manera que el usuario percibe que todos funcionan a la vez, de forma concurrente. *Pseudo-paralelismo*



● Características de los Sistemas Operativos

◆ *Administración de usuarios*

▪ Monousuario:

Sólo permite ejecutar los programas de un único usuario.

▪ Multiusuario:

Si permite que varios usuarios ejecuten simultáneamente sus programas, accediendo a la vez a los recursos de la computadora.

Estos sistemas operativos utilizan métodos de protección de datos, para que un programa no pueda usar o cambiar los datos de otro usuario.



● Características de los Sistemas Operativos

◆ *Manejo de recursos*

■ Centralizado:

Si permite utilizar los recursos de una sola computadora.

■ Distribuido:

Si permite utilizar los recursos (memoria, CPU, disco, periféricos...) de más de una computadora al mismo tiempo.

- **Tipo de sistemas operativos: *Sistemas Operativos por lotes***

- Procesan una gran cantidad de trabajos con poca o ninguna interacción con los usuarios y los programas en ejecución.
- Se reúnen todos los trabajos comunes para realizarlos al mismo tiempo, evitando la espera de dos o más trabajos como sucede en el procesamiento en serie.
- Estos sistemas son de los más tradicionales y antiguos, y fueron introducidos en los 50 para aumentar la capacidad de procesamiento de los programas.
- Cuando estos sistemas son bien planeados, pueden tener un tiempo de ejecución muy alto, porque el procesador es mejor utilizado y los Sistemas Operativos pueden ser simples, debido a la secuenciabilidad de la ejecución de los trabajos.

Ejemplos:

SCOPE, del DC6600, orientado a procesamiento científico pesado

EXEC II, para el UNIVAC 1107, orientado a procesamiento académico.



● Tipo de sistemas operativos: ***Sistemas Operativos de tiempo real***

- Aquí no tiene importancia el usuario, sino los procesos.
- Generalmente están subutilizados sus recursos con la finalidad de prestar atención a los procesos en el momento que lo requieran.
- Se utilizan en entornos donde son procesados muchos sucesos o eventos y a los cuales de les debe dar respuesta en plazos predefinidos.
- Son construidos para aplicaciones muy específicas como control de tráfico aéreo, bolsas de valores, control de refinerías, etc.

Ejemplos: VxWorks, Lyns OS y Spectra.

Características:

- Objetivo es proporcionar rápidos tiempos de respuesta.
- Procesa ráfagas de miles de interrupciones por segundo sin perder eventos.
- Proceso se activa tras ocurrencia de suceso, mediante interrupción.
- Proceso de mayor prioridad expropia recursos.
- Generalmente se utiliza planificación expropiativa basada en prioridades.
- Gestión de memoria menos exigente que tiempo compartido, usualmente procesos son residentes permanentes en memoria.
- Poco movimiento de programas entre almacenamiento secundario y memoria. **15**

- **Tipo de sistemas operativos: *Sistemas Operativos de tiempo compartido***
 - Ofrecen la ilusión de que el sistema y sus recursos son todos para cada usuario.
 - Los principales recursos del sistema: procesador, memoria, dispositivos de E/S, son continuamente utilizados entre los diversos usuarios, dando a cada usuario la ilusión de que tiene el sistema dedicado para sí mismo.
Esto trae como consecuencia una gran carga de trabajo al Sistema Operativo, principalmente en la administración de memoria principal y secundaria.

Ejemplos: Multics, OS/360 y DEC-10.

Características:

- Dan la ilusión de que cada usuario tiene una máquina para sí.
- Mayoría utilizan algoritmo de reparto circular.
- Programas se ejecutan con prioridad rotatoria que se incrementa con la espera y disminuye después de concedido el servicio.
- Evitan monopolización del sistema asignando tiempos de procesador (time slot).
- Gestión de memoria proporciona protección a programas residentes.
- Gestión de archivo debe proporcionar protección y control de acceso debido a que pueden existir múltiples usuarios accediendo un mismo archivos.

- **Tipo de sistemas operativos: *Sistemas Operativos distribuidos***

- Permiten distribuir trabajos, tareas o procesos, entre un conjunto de procesadores de uno o más equipos, en todo caso debe ser transparente para el usuario
- Dos esquemas básicos:
 - Sistema *fuertemente acoplado*: comparte la memoria y un reloj global, cuyos tiempos de acceso son similares para todos los procesadores.
 - Sistema *débilmente acoplado*: los procesadores no comparten ni memoria ni reloj, ya que cada uno cuenta con su memoria local.

Ejemplos: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos

Características:

- Colección de sistemas autónomos capaces de comunicarse y cooperar mediante interconexiones de hardware y software .
- Proporciona abstracción de máquina virtual a los usuarios.
- El objetivo clave es la transparencia.

Web

- http://en.wikipedia.org/wiki/Distributed_computing

- **Tipo de sistemas operativos: *Sistemas Operativos de red***

- Son aquellos sistemas que mantienen a dos o más computadoras unidas mediante algún medio de comunicación (físico o no), con el objetivo primordial de poder compartir los diferentes recursos y la información del sistema.
- El primer Sistema Operativo de red estaba enfocado a equipos con un procesador Motorola 68000, pasando posteriormente a procesadores Intel como Novell Netware

Ejemplos:

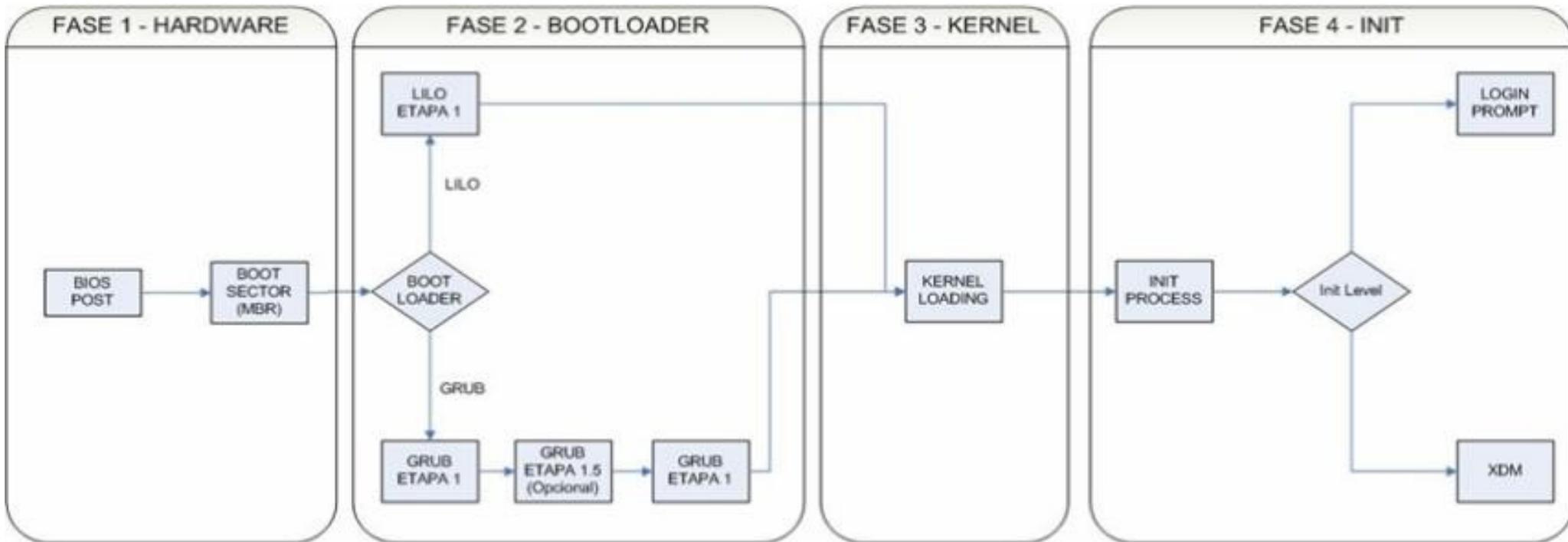
Novell Netware, Personal Netware, LAN Manager, Windows NT Server, UNIX, LANtastic.



● Arranque y Activación del sistema operativo

- El término *Bootstrapping* proviene de la leyenda del Barón de Münchhausen Cordones de los zapatos (boot straps)
- *Booting*, término utilizado para los procesos en el que un sistema simple activa a otro más complejo
- Ejemplos:
Boot loaders: LILO, GRUB, NTLDR
En compiladores: gcc

● Arranque y Activación del sistema operativo



BIOS

Realiza POST
Identifica
boot device

Inicia el bootloader

Carga parte del núcleo
en RAM

Independiente y
dependiente de la
plataforma

Descompresión
Detección y arranque
de dispositivos
integrados al kernel

Monta el fs /
Invoca a init

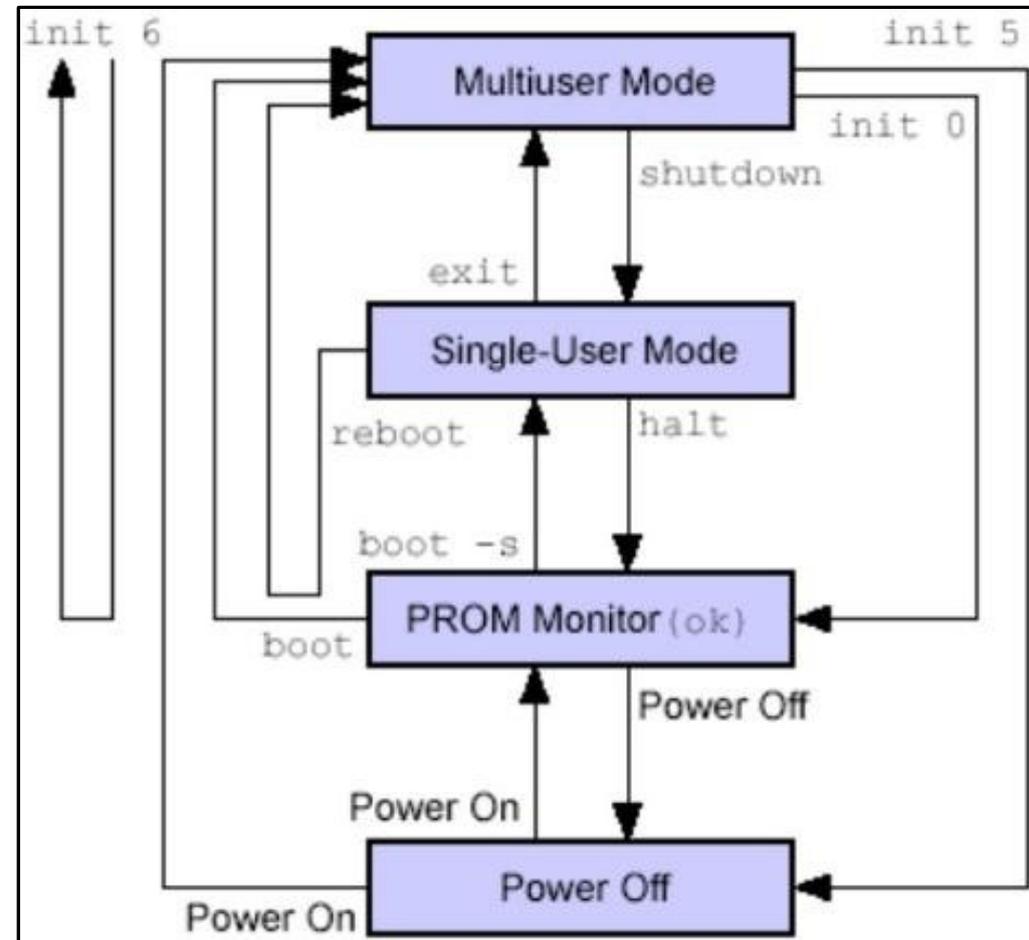
Proceso "1"
/etc/inittab

Run levels



● Run Levels

- 0 Sistema apagado (PROM)
- s, S Single user, sólo file systems críticos
- 1 Single user, todos los file systems
- 2 Multi usuario, sólo algunos recursos de red (NFS, ssh)
- 3 Multi usuario, todos los recursos de red
- 4 No implementado
- 5 Sistema apagándose
- 6 Sistema reiniciándose

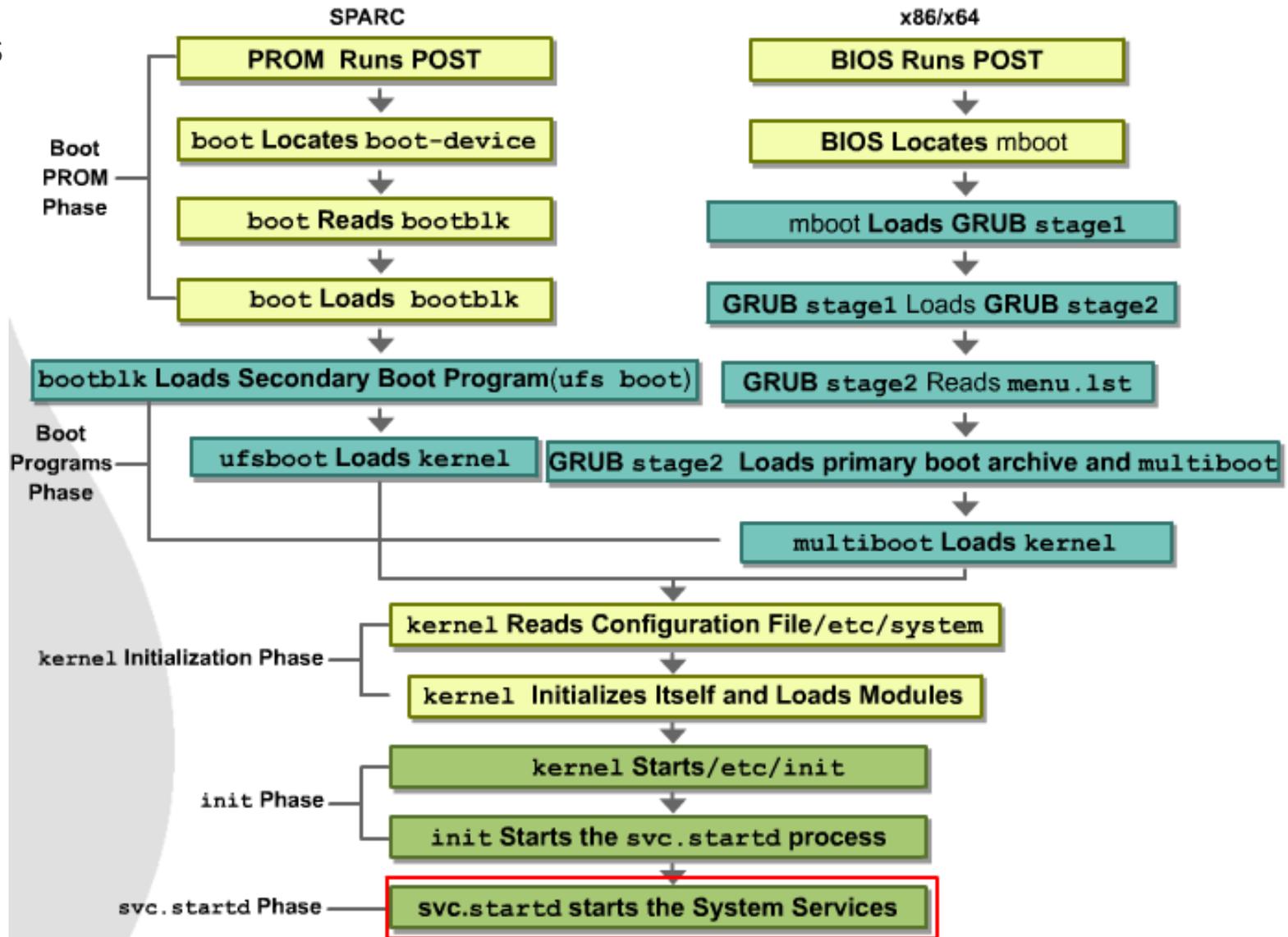


<http://en.wikipedia.org/wiki/Runlevels>



● Arranque y Activación del sistema operativo

▪ En Solaris





● Historia y evolución

▪ Material de lectura

- *Tanenbaum (Sistemas Operativos Modernos)*

Págs. 5 a 13

- *Silberschatz & Galvin (Sistemas Operativos – 5ta edición)*

Págs. 6 a 19

- *Wikipedia*

http://es.wikipedia.org/wiki/Historia_de_los_sistemas_operativos



● Wikipedia – Enlaces organizados

v d e	Operating system		[hide]
General	History • Timeline • List • Comparison • Usage share • Development • Advocacy		
Kernel	Architectures	General	Monolithic kernel • Microkernel
		Subtypes	Exokernel • Nanokernel • Hybrid
	Components	Kernel space • Server Loadable kernel module • Device driver • User space • Userland	
Process management	Process • Multiprogramming • Process control block • Interrupt • Modes: (Protected mode • Supervisor mode) • Computer multitasking • Scheduling • Context switch • Cooperative multitasking • Preemptive multitasking • CPU modes • Thread		
Memory management	Memory protection • Segmentation • Paging • Segmentation fault • General protection fault • Bus error		
Examples	UNIX • OpenSolaris • GNU • Linux • Windows • Mac OS X • AmigaOS 4 • BeOS • IBM OS/2 • ReactOS • MS-DOS • <i>more...</i>		
Miscellaneous concepts	Boot loader • PXE • API • Virtual file system • Virtual tape library • Computer network • CLI • TUI • GUI • VUI • HAL		