

Introducción a los Sistemas Operativos

Presentación



✓ Sitio Web: <http://catedras.info.unlp.edu.ar> (único medio de comunicación)

✓ Explicaciones: (Las primeras semanas Jueves y Viernes)

✓ Turno Mañana: Jueves 8:00 Aula 5

✓ Turno Tarde: Jueves 19:00 Aula 5

✓ Integrantes:

✓ JTP Turno Mañana:

✓ Nicolás del Rio: ndelrio@info.unlp.edu.ar

✓ Alejandro Sabolansky: asabolansky@linti.unlp.edu.ar

✓ JTP Turno Tarde:

✓ Martín Baez: mbaez@info.unlp.edu.ar

✓ Emanuel Nucilli: emanuel.nucilli@gmail.com

■ Nahuel Cuesta Luengo

■ Raúl Benencia

■ Carlos Meza

■ Maximo Zarza

■ María Emilia Corrons

■ Claudia Quintana

■ Leonardo Otonelo

■ Adan Mauri

■ Gonzalo Testa

■ Erica Padovani

■ Andrés Valencia

■ Juan Manuel Filandini

■ Alejandra Sebastián



Los Parciales

- ☑ Entran todos los temas vistos en la práctica (enunciados de TP, explicaciones, material adicional brindado por la cátedra, respuestas al foro)
- ☑ La materia consta de 2 parciales:
 - ✓ Sistemas Operativos Aplicados (Prácticas 1, 2 y 3):
 - ♦ Sábado 20 de Septiembre - 9 horas
 - ♦ Sábado 4 de Octubre - 9 horas
 - ✓ Práctica Normal (Prácticas 4, 5 y 6):
 - ♦ Sábado 1 de Noviembre - 9 horas
 - ♦ Sábado 29 de Noviembre - 9 horas
 - ♦ Recuperatorio General: Sábado 13 de Diciembre
- ☑ Para rendir cada parcial se deben contestar las autoevaluaciones de las prácticas correspondientes (**OBLIGATORIAS!!!**).
- ☑ Para aprobar la cursada hay que aprobar ambos parciales



Introducción a los Sistemas Operativos

Conceptos Generales
Práctica 1



¿Qué es un Sistema Operativo?

- ☑ Es parte esencial de cualquier sistema de cómputo
- ☑ Es un programa que actúa, en principio, como intermediario entre el usuario y el hardware
- ☑ Su propósito: Crear un entorno cómodo y eficiente para la ejecución de programas
- ☑ Su obligación: Garantizar el correcto funcionamiento del sistema
- ☑ Sus funciones principales:
 - ◆ Administrar la memoria
 - ◆ Administrar la CPU
 - ◆ Administrar los dispositivos



¿Qué es un Sistema Operativo? (cont.)

✓ Según Wikipedia:

“...Es un conjunto de programas de computación destinados a realizar muchas tareas...”

✓ Según un usuario estándar: “Lo que aparece cuando prendo la PC”

✓ Según W3Counter (<http://www.w3counter.com>):

Uso de Sistemas Operativos al 2013

1	Windows 7	42.76%
2	Windows XP	20.08%
3	iOS	10.26%
4	Mac OS X	7.55%
5	Android	5.88%
6	Windows 8	4.86%
7	Windows Vista	4.49%
8	Linux	2.27%
9	BlackBerry	0.67%
10	Symbian OS	0.24%

Uso de Sistemas Operativos al 2014

1	Windows 7	39.48%
2	iOS 7	10.10%
3	Windows XP	9.64%
4	Windows 8	8.90%
5	Android 4	8.57%
6	Mac OS X	6.42%
7	Windows Vista	4.21%
8	Linux	2.14%
9	Android	1.35%
10	iOS 6	0.97%



- ✓ Es un Sistema Operativo tipo Unix, pero libre
- ✓ S.O. diseñado por miles de Programadores
- ✓ S.O. Gratuito y de libre distribución (se baja desde la web, CD, etc.)
- ✓ Existen diversas distribuciones (customizaciones)
- ✓ Y fundamentalmente es de código abierto, lo que nos permite estudiarlo, personalizarlo, auditarlo, aprovecharnos de la documentación, etc...

Podemos ver como está hecho!!!



¿GNU?



- ✓ GNU = GNU No es Unix
- ✓ Iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre (el sistema GNU, que aún no está finalizado)
- ✓ Para asegurar que el mismo fuera libre, se necesitó crear un marco regulatorio conocido como GPL (General Public Licence de GNU)
- ✓ En 1985, Stallman crea la FSF (Free Software Foundation), con el fin de financiar al proyecto GNU
- ✓ En 1990, GNU ya contaba con un editor de textos (Emacs), un compilador (GCC) y gran cantidad de bibliotecas que componen un Unix típico.
- ✓ Pero aún seguía faltando la componente principal → Núcleo



¿GNU? (cont.)



- ✓ Si bien ya se venía trabajando en un núcleo conocido como TRIX, es en 1988 que se decide abandonarlo debido a su complejidad (corría en hardware muy costoso)
- ✓ En este momento se decide adoptar como base el núcleo MACH para crear Hurd, el cual tampoco prosperó
- ✓ Linus Torvalds, ya venía trabajando desde 1991 en un Kernel denominado Linux, el cual se distribuiría bajo licencia GPL.
- ✓ En el año 1992, Torvalds y Stallman, deciden fusionar ambos proyectos, y es allí donde nace GNU/Linux
- ✓ GNU/Linux pertenece al desarrollo del software libre





- ✓ GNU Se refiere a 4 libertades principales de los usuarios del software
 - ✓ Libertad de usar el programa con cualquier propósito
 - ✓ Libertad de estudiar su funcionamiento
 - ✓ Libertad para distribuir sus copias
 - ✓ Libertad para mejorar los programas

"Los programas son una forma de expresión de ideas. Son propiedad de la humanidad y deben ser compartidas con todo el mundo."



USAR

**APRENDER
ADAPTAR**

REDISTRIBUIR

**PUBLICAR
MEJORAS**

**UNA FORMA ÉTICA
DE ENTENDER
EL SOFTWARE**

VALORES

con cualquier
propósito
(libertad 0)

el programa a
sus necesidades
(libertad 1)

copias para todos
(libertad 2)

para beneficiar a
toda la comunidad
(libertad 3)

En su desarrollo,
comercialización,
distribución y uso

Ética, creatividad,
eficiencia, colaboración,
no discriminación, ciencia,
transparencia, seguridad,
competitividad, privacidad,
solidaridad, y sobre todo,
libertad



necesita

**LICENCIA
LIBRE**

GPL (Copyleft ©),
Apache, BSD,
Creative Commons
(Share Alike)

requiere

distribuye

producen y
depuran

SOFTWARE LIBRE

para

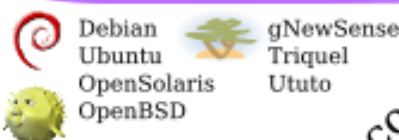
colaboran
(netiqueta)

eligen

DESARROLLADORES/AS

USUARIOS/AS

DISTRIBUCIONES



**CÓDIGO
FUENTE**

instrucciones

con

compilado

**PROGRAMAS
EJECUTABLES**

**CONTENIDOS
LIBRES**

fuentes tipográficas,
traducciones, localizaciones,
plantillas, sonidos, imágenes,
FAQs, guías, manuales...

**ESTÁNDARES
ABIERTOS**

UTF-8 de IETF
html, xml de W3C
OpenDocument de OASIS e ISO

Fundaciones como FSF,
usuarios y proyectos como:



ONGs: Wikipedia, Greenpeace
INSTITUCIONES: UNESCO, NASA
EMPRESAS: Google, IBM, SUN, Nokia
PAISES-REGIONES: Extremadura
Francia, China, Brasil, Alemania
UNIVERSIDADES: MIT, Berkeley
El 89% del TOP500 de superordenadores

reciben
y dan

SOPORTE

EMPRESAS: p.ej. en España dan soporte 150 en ASOLIF y 588 según Libro Blanco
EVENTOS: aKademy, FLISOL **JORNADAS DE** Hispalinux, JRSL, Hackmeeting
LUGS: Velug, GPUL, CafeLug **BLOGOSFERA:** bulma.net, vivalinux.com.ar
WIKIS: emacswiki, wiki.debian.org **FOROS:** preguntaslinux, ubuntu México
IRC (Chats): OFTC, freenode **LISTAS DE CORREO:** bulmailing, asturlinux
ASOCIACIONES: Linux Español, USLA, Solar

¿Software Libre? ¿Software Propietario?

☒ Características del Software Libre:

- ✓ *Una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente*
- ✓ *Generalmente es de costo nulo ← Es un gran error asociar el software libre con el software gratuito ← Pensar en software gratis que se distribuye con restricciones*
- ✓ *Es común que se distribuya junto con su código fuente*
- ✓ *Corrección mas rápida ante fallas*
- ✓ *Características que se refieren a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.*



¿Software Libre? ¿Software Propietario? (cont.)

☑ Características del Software Propietario:

- ✓ *Generalmente tiene un costo asociado*
- ✓ *No se lo puede distribuir libremente*
- ✓ *Generalmente no permite su modificación*
- ✓ *Normalmente no se distribuye junto con su código fuente*
- ✓ *Corrección de fallas está a cargo del propietario*
- ✓ *Generalmente al contar con una inversión mayor, proveen mas avances y funcionalidad*
- ✓ *Menor necesidad de técnicos especializados*



GPL: General Public License

- ✓ Licencia Pública General de GNU
- ✓ Creada en el año 1989 por la FSF
- ✓ Su objetivo principal es proteger la libre distribución, modificación y uso del software GNU
- ✓ Su propósito es declarar que todo software publicado bajo esta licencia es libre y protegerlo teniendo en cuenta las 4 libertades principales ya vistas
- ✓ La versión actual de la licencia es la 3



Características generales de GNU/Linux

- ✓ Es multiusuario
- ✓ Es multitarea y Multiprocesador
- ✓ Es altamente portable
- ✓ Posee diversos intérpretes de comandos y algunos de ellos son programables
- ✓ Permite el manejo de usuarios y permisos
- ✓ Todo es un archivo (hasta los dispositivos y directorios)
- ✓ Cada directorio puede estar en una partición diferente (ejemplos /tmp, /home, etc)
- ✓ Es case sensitive
- ✓ Es de código abierto



- ✓ Fue desarrollado buscando la portabilidad de los fuentes
- ✓ Desarrollo en capas
 - ✓ Separación de funciones
 - ✓ Cada capa actúa como una caja negra hacia las otras
 - ✓ Posibilita el desarrollo distribuido
- ✓ Soporte para diversos File Systems
- ✓ Memoria virtual = RAM + SWAP
- ✓ Desarrollo mayoritario en C y assembler
- ✓ Otros lenguajes: java, perl, python, etc.



Estructura básica del S.O. - Núcleo

- ✓ También conocido como Kernel
- ✓ Ejecuta programas y gestiona dispositivos de hardware
- ✓ Es el encargado de que el software y el hardware puedan trabajar juntos
- ✓ Sus funciones más importantes son la administración de memoria y CPU
- ✓ En si, y en un sentido estricto, es el sistema operativo
- ✓ Es un núcleo monolítico híbrido:
 - ✓ Los drivers y código del Kernel se ejecutan en modo privilegiado
 - ✓ Lo que lo hace híbrido es la posibilidad de cargar y descargar funcionalidad a través de módulos
- ✓ Está licenciado bajo la licencia GPL v2

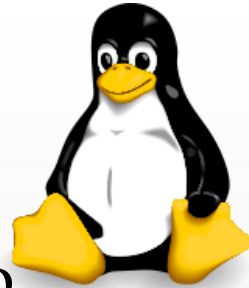


Núcleo - Un poco de Historia

- ✓ En 1991 Linus Torvalds inicia la programación de un Kernel Linux basado en Minix (Clon de Unix desarrollado por Tanenbaum en 1987 con el fin de crear un SO de uso didáctico).
- ✓ El 5 de octubre de 1991, se anuncia la primera versión "oficial" de Linux (0.02)
- ✓ En 1992 se combina su desarrollo con GNU, formando GNU/Linux
- ✓ La versión 1.0 apareció el 14 de marzo de 1994
- ✓ Desarrollo continuado por miles de programadores alrededor del mundo



Núcleo - Un poco de Historia (cont.)



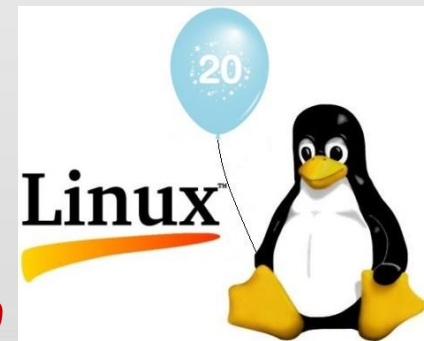
- ✓ En mayo de 1996 se decide adoptar a Tux como mascota oficial de Linux
- ✓ En julio de 1996 se lanza la versión 2.0 y se define un sistema de nomenclatura. Se desarrolló hasta Febrero de 2004 y terminó con la 2.0.40
- ✓ En enero de 1999 se lanza la versión 2.2, que provee mejoras de portabilidad entre otras y se desarrolla hasta Febrero de 2004 terminando en la versión 2.2.26
- ✓ En 2001 se lanza la versión 2.4 y se deja de desarrollar a fines del 2010 con la 2.4.37.11
 - ✓ La versión 2.4 fue la que catapultó a GNU/Linux como un SO estable y robusto. Durante este período es que se comienza a utilizar Linux mas asiduamente



Núcleo - Un poco de Historia (cont.)

- ✓ A fines del 2003 se lanza la versión 2.6
- ✓ Esta versión ha tenido muchas mejoras para el SO dentro de las que se destacan soporte de hilos, mejoras en la planificación y soporte de nuevo hardware
- ✓ El 3 de agosto de 2011 se lanza la versión 2.6.39.4 anunciándose la misma desde meses previos como la última en su revisión
- ✓ En 17 Julio de 2011 se lanza la versión 3.0
 - ✓ No agrega mayores cambios. La decisión del cambio son los 20 años del SO y no superar los 40 números de revisión
 - ✓ Totalmente compatible con 2.6
 - ✓ La versión actual es la 3.10.9
 - ✓ Mejoras en Virtualización, File Systems, y soporte de Hardware

http://kernelnewbies.org/Linux_3.0



☑ Modo de Versionar: A.B.C.[D]

- ✓ **A:** Denota Versión. Cambia con menor Frecuencia. en 1994 (versión 1.0) y en 1996 (versión 2.0)
- ✓ **B:** Denota Mayor revisión. Antes de la versión 2.6, los números impares indicaban desarrollo, los pares producción
- ✓ **C:** Denota Menor revisión. Solo cambia cuando hay nuevos drivers o características
- ✓ **D:** Cambia cuando se corrige un grave error sin agregar nueva funcionalidad ← *Casi no se usa en la rama 3.X y esto se ve reflejado en C*



- ✓ También conocido como CLI (Command Line Interface)
- ✓ Modo de comunicación entre el usuario y el SO
- ✓ Ejecuta programas a partir del ingreso de comandos
- ✓ Cada usuario tiene una interfaz o shell
- ✓ Se pueden Personalizar
- ✓ Son programables
- ✓ Bourne Shell (sh), Korn Shell (ksh), Bourne Again Shell (bash) (autocompletado, history, alias)



- ✓ Organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento (fat, ntfs, ext2, ext3, reiser, ...)
- ✓ El adoptado por GNU/Linux es el Extended (v2, v3, v4)
- ✓ Hace un tiempo se está debatiendo el reemplazo de ext por Btrfs (B-tree FS) de Oracle
 - ✓ Soporte de mayor tamaño de archivos
 - ✓ Mas tolerante a fallas y comprobación sin necesidad de desmontar los FS
 - ✓ Indexación
 - ✓ Snapshots
 - ✓ Compresión
 - ✓ Defragmentación



- ☑ Directorios mas importantes según FHS (File Hierarchy Standar):
 - ✓ / Tope de la estructura de directorios. Es como el C:\
 - ✓ /**home** Se almacenan archivos de usuarios (Mis Documentos)
 - ✓ /**var** Información que varía de tamaño (logs, BD, spools)
 - ✓ /**etc** Archivos de Configuración
 - ✓ /**bin** Archivos binarios y ejecutables
 - ✓ /**dev** Enlace a dispositivos
 - ✓ /**usr** Aplicaciones de usuarios



Estructura básica del S.O. – Utilidades

- ☑ Paquete de software que permite diferenciar una distribución de otra.
- ☑ Editores de texto:
 - ✓ vi
 - ✓ emacs
 - ✓ joe
- ☑ Herramientas de Networking:
 - ✓ wireshark
 - ✓ tcpdump
- ☑ Paquetes de oficina:
 - ✓ OpenOffice
- ☑ Interfaces Gráficas:
 - ✓ GNOME / CINNAMON
 - ✓ KDE
 - ✓ LXDE



Distribuciones

- ✓ Una distribución es una customización de GNU/Linux formada por una versión de kernel y determinados programas con sus configuraciones:

<http://www.linux.com/directory/Distributions>

distribution

e.g. Debian, Red Hat, SUSE, Mandriva

distribution specific applications
(e.g. for configuration, installation like 'fst, moc)

proprietary applications

(e.g. Adobe Reader, graphics driver)

Linux kernel

free applications

(e.g. KDE, OpenOffice, Apache)

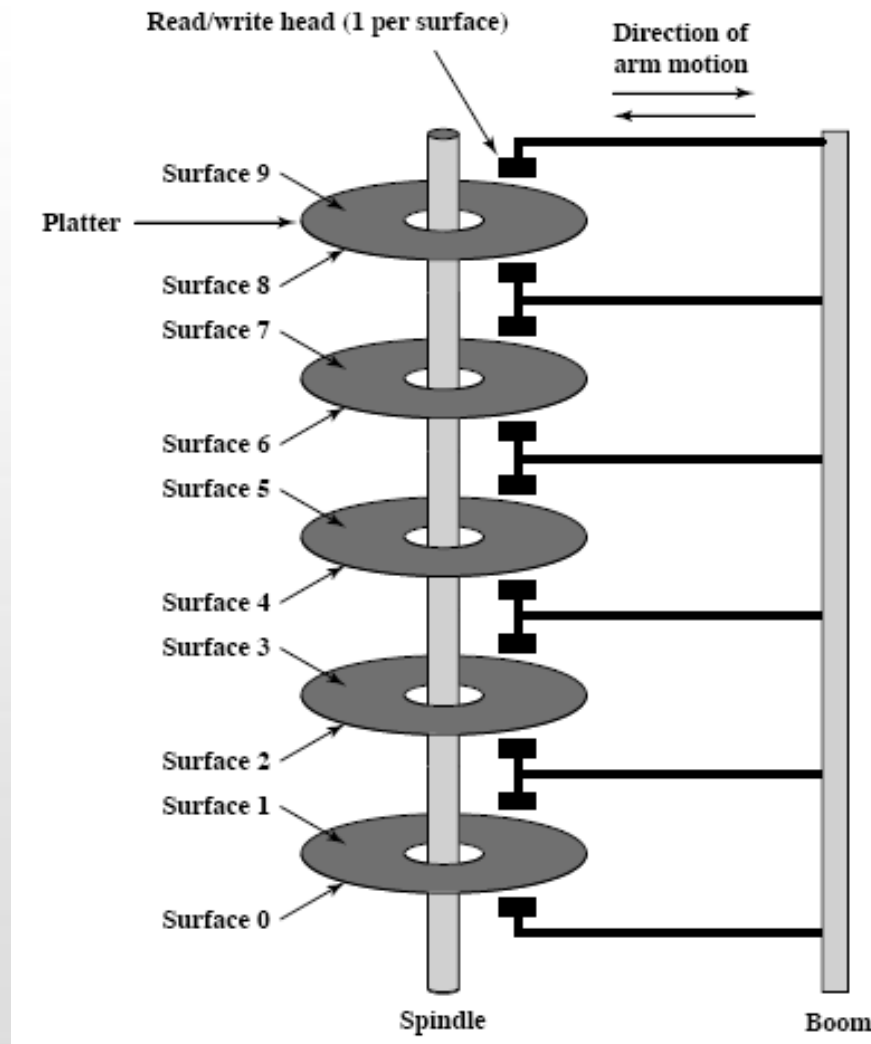
Manuals

Support

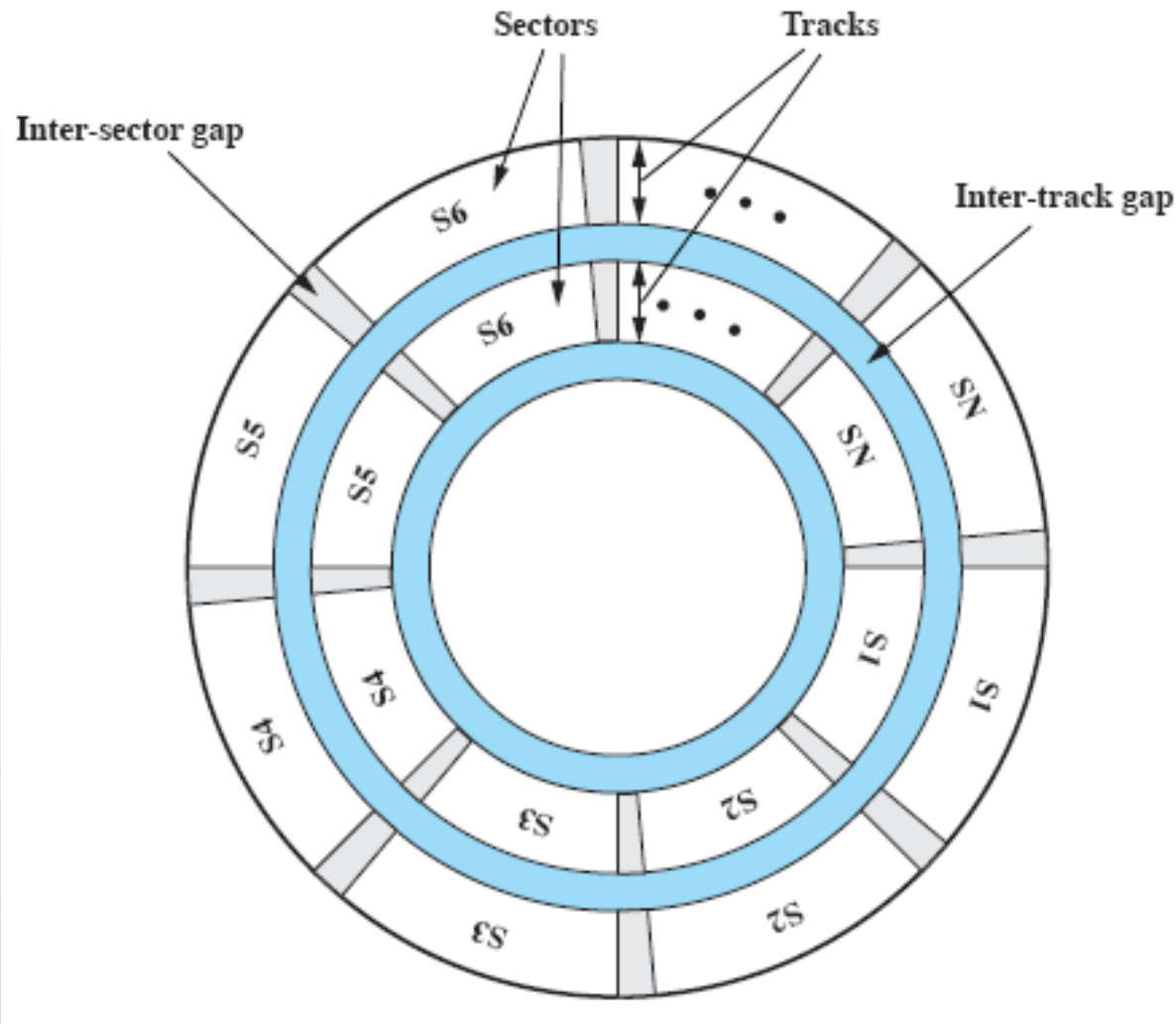
(phone mail, etc)



Organización Física de Discos



Organización Física de Discos (Cont.)



Conceptos previos a la Instalación - MBR

- ✓ Sector reservado del disco físico (cilindro 0, cabeza 0, sector 1)
- ✓ Existe un MBR en todos los discos.
- ✓ Si existiese más de un disco rígido en la máquina, sólo uno es designado como *Primary Master Disk*
- ✓ El tamaño del MBR coincide con el tamaño estándar de sector: 512 bytes:
 - ✓ Los primeros bytes corresponden al *Master Boot Code (MBC)*
 - ✓ A partir del byte 446 está la tabla de particiones. Es de 64 bytes
 - ✓ Al final existen 2 bytes libres
- ✓ Es creado con algún utilitario



Conceptos previos a la Instalación – MBR (cont.)

- ✓ EL MBR es un pequeño código que permite arrancar el SO
- ✓ La última acción del BIOS es leer el MBR. Lo lleva a memoria y lo ejecuta.
- ✓ El proceso de inicio de una máquina y carga del sistema operativo se denomina *proceso de bootstrap*
- ✓ Si se tiene un sistema instalado → MBR típico. Sinó → uno diferente (bootadores)
- ✓ El proceso de booteo puede verse como una serie de pequeños programas cuya ejecución se va encadenando
- ✓ La finalidad del Bootloader es la de cargar una imagen de Kernel de alguna partición para su ejecución



Conceptos previos a la Instalación – Particiones

- ☑ Es una forma de dividir lógicamente el disco físico:
 - ✓ DOS y W95 no pueden manejar filesystems mayores a 2 GB
 - ✓ Cada sistema operativo es instalado en una partición separada
 - ✓ Cada partición se formatea con un tipo de File System distinto (fat, ntfs, ext, etcétera)
 - ✓ Es una buena práctica separar los datos del usuario de las aplicaciones y/o Sistema Operativo instalado
 - ✓ Tener una partición de Restore de todo el sistema
 - ✓ Poder ubicar el Kernel en una partición de solo lectura, o una que ni siquiera se monta (no está disponible a los usuarios)
 - ✓ Particionar demasiado un disco puede tener desventajas:
¡Pensar...!



Conceptos previos a la Instalación – Particiones (cont.)

- ✓ Debido al tamaño acotado en el MBR para la tabla de particiones:
 - ✓ Se restringe a 4 la cantidad de particiones primarias
 - ✓ o 3 particiones primarias y una extendida con sus volúmenes o particiones lógicas
- ✓ Una de las 4 particiones puede ser extendida, la cual se subdivide en volúmenes lógicos
- ✓ Partición Primaria: División cruda del disco (puede haber hasta 4 por disco). Se almacena información de la misma en el MBR
- ✓ Partición Extendida: Sirve para contener unidades lógicas en su interior. Solo puede existir una partición de este tipo por disco. No se define un tipo de FS directamente sobre ella
- ✓ Partición Lógica: Ocupa la totalidad o parte de la partición extendida y se le define un tipo de FS directamente.



Conceptos previos a la Instalación – Particiones (cont.)

- ✓ Como mínimo es necesario una partición (para el /)
- ✓ Es recomendable crear al menos 2 (/ y SWAP)
- ✓ Para crearlas, se utiliza software denominado Particionador. Existen 2 tipos:
 - ✓ Destructivos: Permiten crear y eliminar particiones (fdisk)
 - ✓ No destructivos: Permiten crear, eliminar y MODIFICAR particiones (fips, gparted) ←
Generalmente las distribuciones permiten hacerlo desde la interfaz de instalación

¿Para qué podríamos crear otras particiones?



- ✓ Vamos a trabajar en un ambiente controlado → VirtualBox
- ✓ Necesitamos crear una máquina virtual, asignarle recursos y un disco
- ✓ Booteamos la máquina virtual iniciando desde algún medio de instalación
- ✓ Seguimos las instrucciones de instalación
- ✓ Verificamos que podamos arrancar el/los Sistemas Operativos instalados



- ☑ Particionando – 3 escenarios posibles:
 - ◆ Usar espacio libre no particionado
 - ◆ Usar una partición no usada
 - ◆ Usar espacio libre de una partición activa (mas complicado)
 - ◆ Cambio destructivo
 - ◆ Cambio no destructivo

☑ En nuestra instalación

- ✓ /dev/hda1: DOS con Windows (2 G.) ← *Es mucho!*
- ✓ /dev/hda2: /boot: 60 MB. Aprox **(B)**
- ✓ /dev/hda3: /: 6 GB Aprox.
- ✓ /dev/hda4: Área de Intercambio (SWAP)



Verificando la configuración de nuestro disco

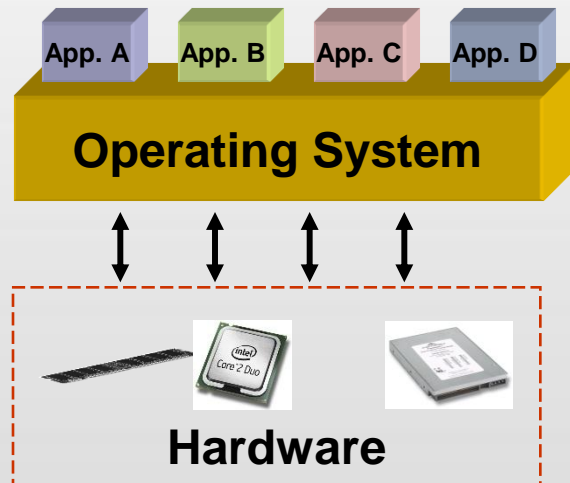
- ✓ Desde Windows: Utilizamos el administrador de discos
- ✓ Desde DOS: usamos fdisk
- ✓ Para instalar un nuevo SO necesitamos espacio libre sin particionar
- ✓ Si no lo tenemos, debemos generarlo. Para esto existen diversos escenarios:
 - ✓ Eliminar una partición existente
 - ✓ Redimensionar una partición

¿Qué ocurre en cada caso? ¿Qué software vamos a usar?

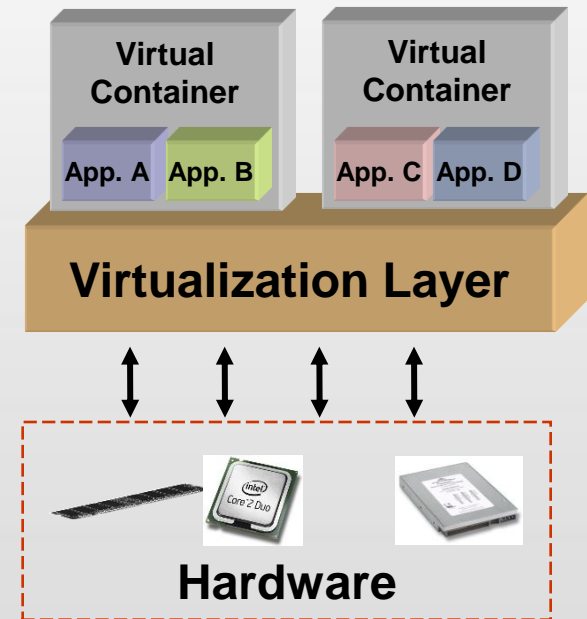


Conceptos previos a la Instalación – Emuladores/Virtualizadores

- ✓ Para los fines de este curso, permiten virtualizar plataformas
- ✓ Permite que en un equipo puedan correr varios S.O. en forma simultánea compartiendo recursos de hardware
- ✓ El pionero en esta tecnología fue IBM con el IBM System/360 en los años 1960



Vs.



✓ Básicamente se pueden considerar 3 tipos:

✓ Emulación:

- ◆ Emulan Hardware
- ◆ Tienen que implementar todas las instrucciones de la CPU
- ◆ Es muy costosa y poco eficiente
- ◆ Permite ejecutar arquitecturas diferentes a las soportadas por el hard.

✓ Virtualización Completa:

- ◆ Permiten ejecutar SO huéspedes en un sistema anfitrión (host)
- ◆ Utilizan en el medio un hypervisor o monitor de máquinas virtuales
- ◆ El SO huésped debe estar soportado en la arquitectura anfitriona
- ◆ Es mas eficiente que la Emulación (Intel-VT y AMD-V)

✓ Paravirtualización

- ◆ Permite correr SOs modificados exclusivamente para actuar en entornos virtualizados
- ◆ Mayor eficiencia que la virtualización



Conceptos previos a la Instalación – Emuladores/Virtualizadores

- ✓ Las principales diferencias entre ellos son:
 - ✓ Los virtualizadores aprovechan la CPU sobre la que están trabajando, lo cual los hace más veloces
 - ✓ En un emulador se puede correr cualquier arquitectura. En un virtualizador solo se puede correr la arquitectura virtualizada.

Emuladores:



Virtualizadores



ParaVirtualizadores



Gestores de Arranque

- ✓ Permiten la carga del sistema Operativo.
- ✓ Se ejecuta luego del código de la BIOS
- ✓ Existen 2 modos de instalación
 - ✓ En el MBR
 - ✓ En el sector de arranque de la partición raíz o activa
- ✓ GRUB, LILO, NTLDR, GAG, YaST, etc...



Proceso de Arranque

- ✓ Se denomina bootstrap (carga)
- ✓ En las arquitecturas x86, el *BIOS (Basic I/O System)* es el responsable de iniciar la carga del SO a través del MBC
 - ✓ Está grabado en un chip (ROM, NVRAM)
 - ✓ En otras arquitecturas también existe, pero se lo conoce con otro nombre:
 - ◆ Power on Reset + IPL en mainframe
 - ◆ OBP (OpenBoot PROM): en SPARC
- ✓ Carga el programa de booteo (desde el MBR)
- ✓ El gestor de arranque lanzado desde el MBC carga el Kernel
 - ✓ Prueba e inicializa los dispositivos
 - ✓ Luego pasa el control al proceso **init**
- ✓ El proceso de arranque se ve como una serie de pequeños programas de ejecución encadenada



Grub Legacy

- ✓ **GRand Unified Bootloader:** Gestor de arranque múltiple mas utilizado
- ✓ En el MBR solo se encuentra la Fase 1 del Grub (Stage 1) que solo se encarga de cargar la fase 1.5
- ✓ La Fase 1.5 (Stage 1.5) se encuentra ubicado en los siguientes 30 KB. del disco y carga la fase 2
- ✓ La Fase 2 (Stage 2) almacenada en disco, presenta una interfaz al usuario y carga el Kernel Seleccionado
- ✓ Se configura a través del archivo `/boot/grub/menu.lst`
- ✓ Algunas líneas:

- ✓ default: Define el SO por defecto a bootear

- ✓ timeout: Tiempo de espera para cargar el SO por defecto

```
title Debian GNU/Linux
```

```
root (hd0,1) #(Disco,Particion)
```

```
kernel /vmlinuz-2.6.26 ro quiet root=/dev/hda3
```

```
initrd /initrd-2.6.26.img
```



- ✓ Actualmente el desarrollo está enfocado a Grub2, y se está comenzando a utilizar en la mayoría de las distribuciones.
- ✓ Dentro de sus mejoras, se incluye el soporte en nuevas arquitecturas, soporte de caracteres no ASCII, idiomas, customización de menús y otras
- ✓ En Grub 2 la fase 1.5 ya no existe mas
- ✓ El archivo de configuración ahora es `/boot/grub/grub.cfg` y no debería editarse manualmente.
- ✓ La información del archivo de configuración se actualiza al ejecutar el comando `update-grub`

Mas información en:

<https://help.ubuntu.com/community/Grub2>



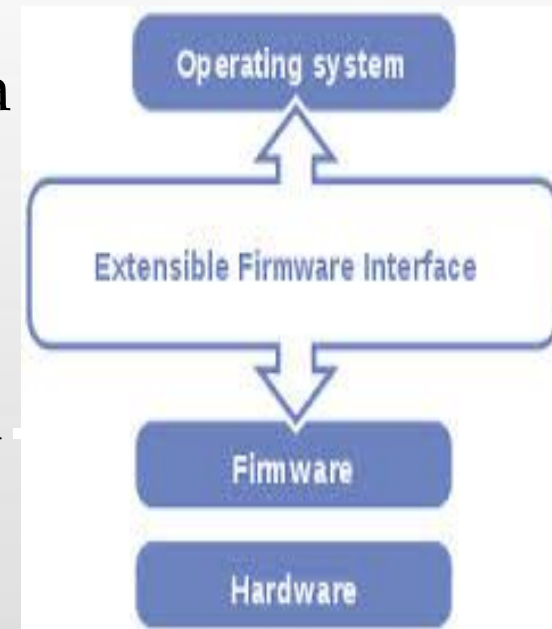
Problemas del arranque basado en BIOS

- ☑ Problemas del arranque basado en BIOS
 - ✓ Data de la década del 80
 - ✓ La última acción del BIOS es leer el MBC del MBR
 - ✓ El firmware del BIOS no facilita la lectura de file systems
 - ✓ El MBC no puede ocupar mas de 440 bytes
 - ◆ Por ejemplo Grub, utiliza los 440 Bytes y luego utiliza sectores del disco adyacentes que deberían estar libres (MBR Gap)



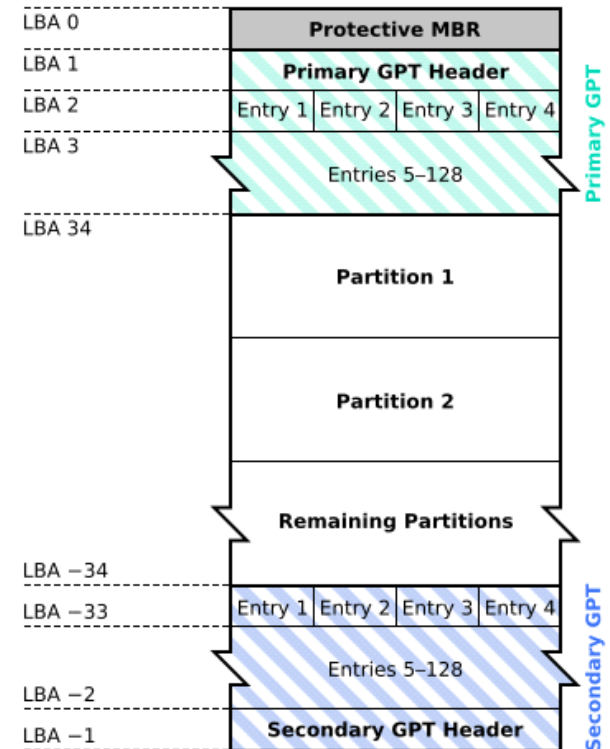
Extensible Firmware Interface

- ✓ EFI: Nexa entre el SO y el firmware
- ✓ Utiliza el sistema GPT (GUID Partition table) para solucionar limitaciones del MBR, como la cantidad de particiones.
- ✓ GPT especifica la ubicación y formato de la tabla de particiones en un disco duro.
- ✓ Es parte de EFI. Puede verse como una sustitución del MBR
- ✓ La especificación EFI es propiedad de Intel



- ✓ Se mantiene un MBR para tener compatibilidad con el esquema BIOS
- ✓ GPT usa modo de direccionamiento lógico (LBA, logical block addressing)
- ✓ El MBR “heredado” se almacena en el LBA 0
- ✓ En el LBA 1 está la cabecera GPT. La tabla de particiones en sí está en los bloques sucesivos
- ✓ La cabecera GPT y la tabla de particiones están escritas al principio y al final del disco

GUID Partition Table Scheme



☑ UEFI Forum

- ✓ Alianza entre varias compañías con el objetivo de modernizar el proceso de arranque
- ✓ Representantes de AMD, American Megatrends, Apple, HP, Dell, IBM, Insyde Software, Intel, Lenovo, Microsoft, Phoenix Technologies
- ✓ EFI es propiedad de Intel.
- ✓ UEFI es propiedad del UEFI Forum
- ✓ UEFI aporta criptografía, autenticación de red, y una interfaz gráfica



- ☑ Define la ubicación de gestor de arranque
- ☑ Define la interfaz entre el gestor de arranque y el firmware
- ☑ Expone información en tablas en memoria para los gestores de arranque con:
 - ✓ Información de hard y configuración del firmware
 - ✓ Punteros a rutinas que implementan los servicios que el firmware ofrece a los bootloaders u otras aplicaciones UEFI
 - ✓ Provee un BootManager para cargar aplicaciones UEFI(ej: GRUB) y drivers desde un UEFI File System
 - ✓ El BootLoader ahora es un tipo de aplicación UEFI
 - ◆ El GRUB será una aplicación UEFI, que vive en el UEFI File System donde están los drivers necesarios para arrancar el sistema(FAT32).
 - ◆ De esta forma para el GRUB deja de ser necesario el arranque en varias etapas!



Secure Boot

- ✓ Propone mecanismos para un arranque libre de código malicioso.
- ✓ Las aplicaciones y drivers UEFI (imágenes UEFI) son validadas para verificar que no fueron alteradas.
- ✓ Se utilizan pares de claves asimétricas
- ✓ Se almacenan en el firmware una serie de claves públicas que sirven para validar que las imágenes estén firmadas por un proveedor autorizado.
- ✓ Si la clave privada está vencida o fue revocada la verificación puede fallar

