

# 1. INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS



# SISTEMA OPERATIVO

Colección de Software que sirve de intermediario entre el hardware y las aplicaciones de usuario

**Abraham Silberschatz asegura que**

"Desde el punto de vista de la computadora, el sistema operativo es el programa mas íntimamente relacionado con el hardware"

**Según Andrew S. Tanenbaum**

"Los verdaderos clientes del sistema operativo son los programas de aplicaciones. Son los que tratan directamente con el sistema operativo y sus abstracciones"

El Sistema Operativo administra eficientemente los recursos de hardware de una computadora



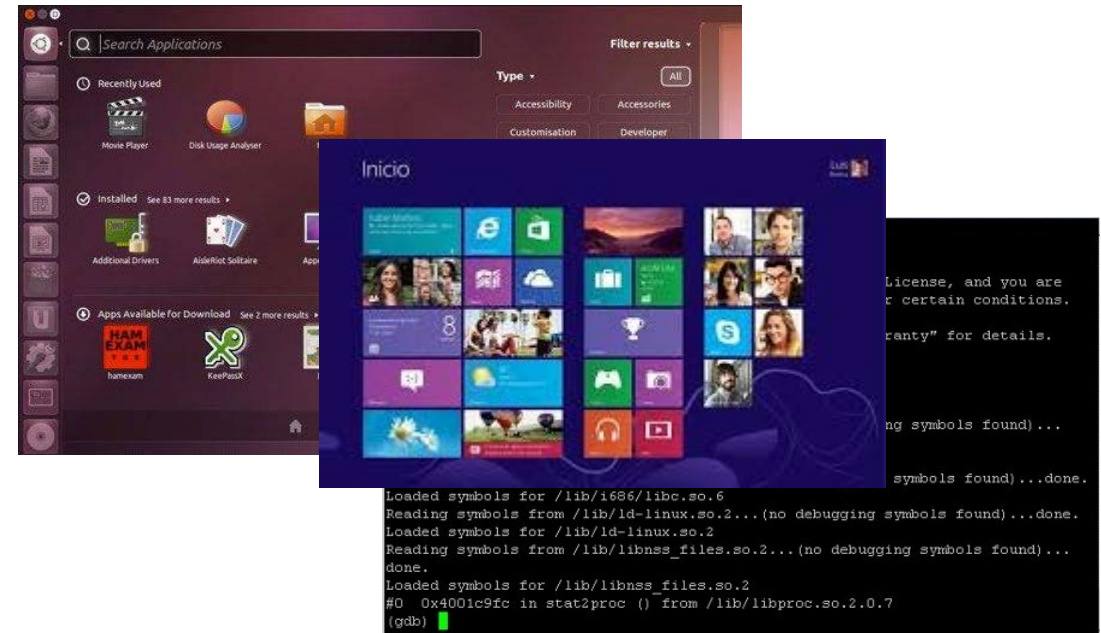
# SISTEMA OPERATIVO

- Modo Supervisor o Kernel: Acceso completo a todo el hardware, puede ejecutar cualquier instrucción de maquina.
- Modo Usuario: Solo un subconjunto de instrucciones.



# SISTEMA OPERATIVO

- Que ve el Usuario?
  - GUI Interfaz Grafica de Usuarios
  - Interfaz de Línea de Comandos (Programables)

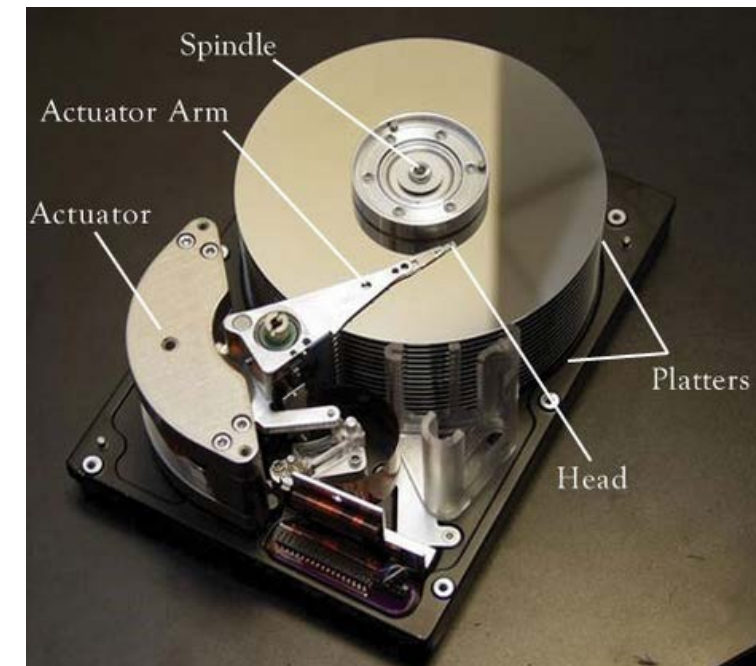
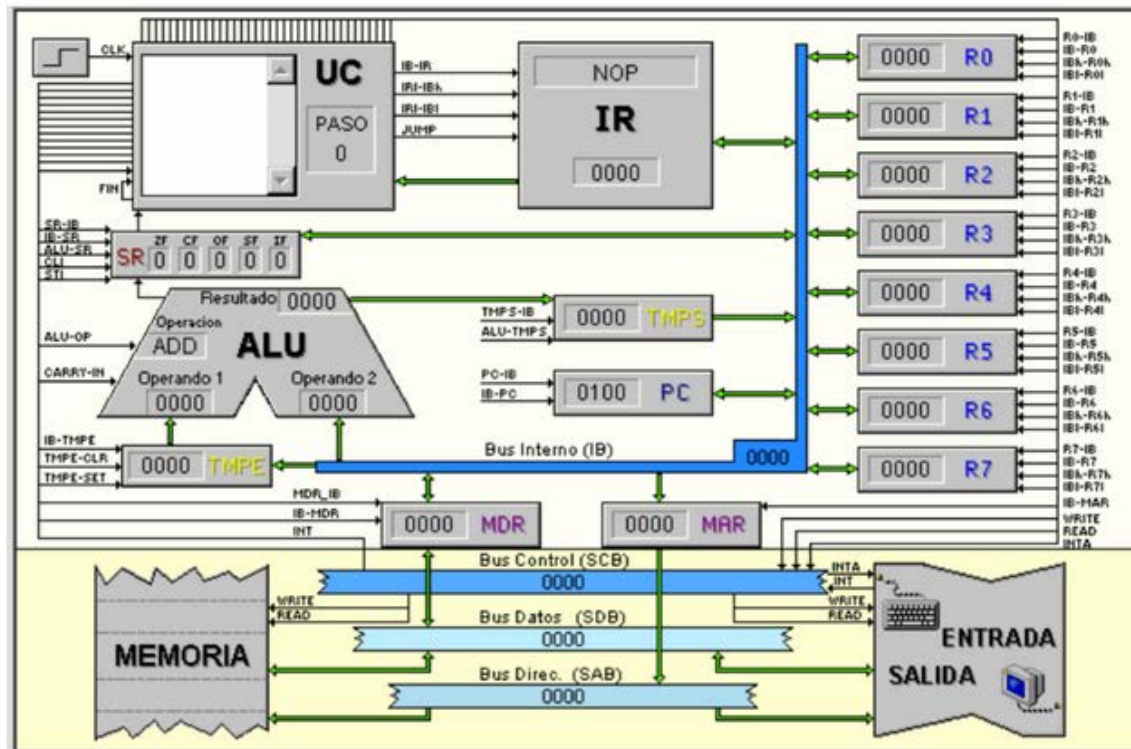


- 2 funciones principales
  - **Presentar un conjunto abstracto de recursos simples a los programas de usuario y a los programadores**
  - **Administra recursos**



# COMO PROGRAMADORES

- Que pasaría si no hubiera SO????





# ADMINISTRACIÓN DE RECURSOS DEL SO

- Administración Memoria

- Verificar la validez de cada solicitud por espacio de memoria , si es legal, asigna una porción de la memoria que no esta en uso. Deasignarla.
- Multiusuario Tabla para seguir la pista de quien esta usando que sección de la memoria.
- Proteger el espacio ocupado por el SO en memoria y por otros procesos
- Los procesos se pueden intercambiar a almacenamiento secundario
- Intercambio (Swaping), Segmentación

- Administración procesador

- Decida como asignar la CPU, cual proceso y durante cuanto tiempo
- Seguir la pista del estado de cada proceso (instancia de ejecución de un programa)



# ADMINISTRACIÓN DE RECURSOS DEL SO

- Administración Dispositivos
  - Supervisa cada dispositivo, canal y unidad de control
  - Escoger la manera mas eficiente para asignar los dispositivos del sistema: impresoras, puertos, unidades de disco, etc. con base en una política de planificación
  - Asigna el recurso, empieza su operación y desasigna el dispositivo poniéndolo a disposición de otro proceso
- Administración de archivos
  - Sigue la pista de todos los archivos en el sistema: datos, archivos de programa, compiladores y aplicaciones
  - Políticas de acceso predeterminado para restringir quien tiene acceso a que.
  - Control de acceso: lo que se permite a los usuarios una vez que les dio acceso (leer, escribir, crear, borrar)
  - Concurrencia: Múltiples accesos



# ADMINISTRACIÓN DE RECURSOS DEL SO

- Administración de redes
  - El sistema debe permitir que los usuarios puedan participar de recursos y controlar el acceso de otros usuarios a los mismos
  - Hardware (CPU, memoria, impresoras, etc) Software (compiladores, programas de aplicación, archivos de datos, etc)
- Administración Usuarios
  - Personas autorizadas para ejecutar ciertos procesos
  - Usuarios, Grupos
  - Permisos de acceso a recursos (Procesos, archivos, etc)
  - SuperUsuarios: Administrador, root





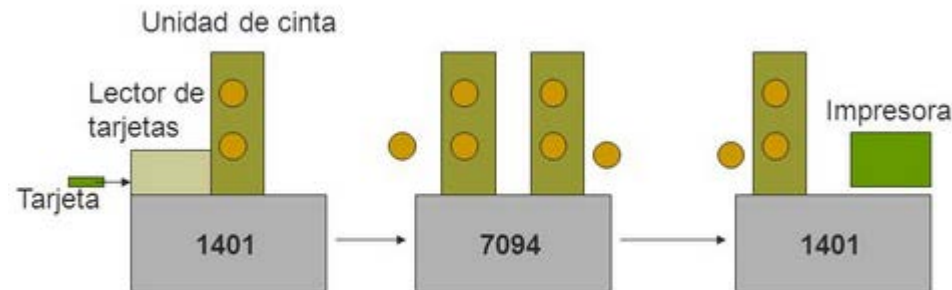
# 1.1. CLASIFICACIÓN Y ESTRUCTURAS GENÉRICAS DE LOS SISTEMAS OPERATIVOS VIGENTES

- Clasificación de los SOs
  - ADECUACION DE LOS SISTEMAS A ENTORNOS CAMBIANTES (HARDWARE, NECESIDADES)
  - Sistemas Operativos por lotes
  - Sistemas Operativos Multiprogramación
  - Sistemas Operativos Multiusuario
  - Sistemas Operativos Tiempo Compartido
  - Sistemas Operativos Tiempo Real



# SISTEMAS OPERATIVOS POR LOTES

- Requieren información reunida en bloques o lotes: el programa, los datos, las instrucciones.
- No interacción Usuarios
- Planificación Sencilla: Los trabajos se procesan en el orden de admisión.  
First Come First Served
  - Interrumpir procesos muy largos, implementar protección.
  - Largo tiempo de carga y puesta a punto de una tarea
- Análisis estadísticos, Nominas
- **Alternativa: Sistemas en lotes con dispositivos de carga (spool)**



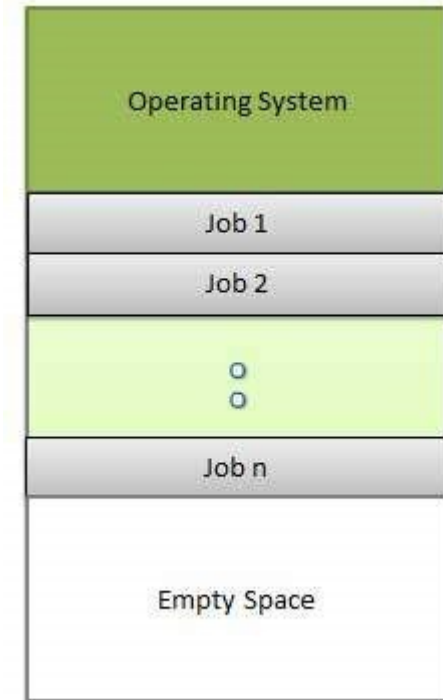
# SISTEMAS OPERATIVOS

## MULTIPROGRAMACIÓN

- Limitado por CPU y Limitado por Entrada-Salida



- Multiprogramación o Multitareas → Soportan dos o mas procesos concurrentes múltiples.
- **Busca maximizar el tiempo de uso efectivo del procesador**
- En la memoria primaria pueden residir al mismo tiempo las instrucciones y datos de dos o mas procesos
- Necesidad de infraestructura de proteccion de recursos
  - Un proceso no debe sobrescribir el espacio de memoria de otro (ni código, ni datos)--> Solución MMU (unidad de manejo de memoria)
  - Acceso exclusivo/unico a dispositivos



# SISTEMAS OPERATIVOS MULTIUSUARIO

- Aparición de las terminales (primero teletipos seriales, posteriormente equipos con una pantalla completa como se conocen hasta hoy)
- Permiten acceder simultáneamente a un sistema de computadoras a través de dos o mas terminales
- Fundamental en el manejo de redes de computadoras actualmente.



# SISTEMAS OPERATIVOS TIEMPO COMPARTIDO

- Multiusuarios e Interactivos
- Pretenden proporcionar un reparto equitativo de recursos comunes para que usuarios tengan la percepción de que poseen una computadora independiente.
- El administrador de memoria debe proporcionar aislamiento y protección de los programas, generalmente los programas no tienen necesidad de comunicarse entre ellos.
- Nacen de forma paulatina abstracciones como el concepto de archivos y directorios, el código para emplearlos se agregó a las bibliotecas de sistema y al núcleo del SO



# TIPO DE CONTROL SOBRE LA MULTITAREA

- **Multitarea cooperativa o no apropiativa** (Cooperative multitasking). La implementaron los sistemas multiprogramación: cada proceso tenía control del CPU hasta que el mismo proceso hacía una llamada al sistema (o indicara su disposición a cooperar por medio de la llamada yield: ceder el paso).
  - **Voluntariamente libera el control del procesador**
- Desventaja: Un cálculo largo no era interrumpido por el sistema operativo, en consecuencia un error de programador podía congelar la computadora completa.
- **Multitarea preventiva o apropiativa** (Preemptive multitasking). En los sistemas de tiempo compartido, el reloj del sistema interrumpe periódicamente a los diversos procesos, transfiriendo forzosamente el control nuevamente al sistema operativo. Éste puede entonces elegir otro proceso para continuar la ejecución.





# OTRA DIFERENCIA: VELOCIDAD DE CAMBIO ENTRE UNA TAREA Y OTRA

- En un **sistema multiprogramado** un cambio de contexto podía producirse sólo cuando la tarea cambiaba de un modo de ejecución a otro
- En un **sistema interactivo**, para dar la “ilusión de uso exclusivo de la computadora: interacción”, el hardware emitía periódicamente al sistema operativo interrupciones (señales) que le indicaban que cambie el proceso activo (como ahora se le denomina a una instancia de un programa en ejecución).
- ...También surgieron las Prioridades!!!



# Y EN COMPUTADORAS PERSONALES...

- Década de los 70
  - Reducidas en prestaciones, precios accesible para cualquiera.
- Altair 8800 (1975) Sin SOs o lenguajes de programación.
- Commodore Pet 2001 (1977) Proto-SO, BASIC
- IBM, PC 5150 (1981). SO administración de archivos a través de CLI, lanzar aplicaciones, PC-DOS
  - MS-DOS en clones



# SISTEMAS OPERATIVOS TIEMPO REAL

- Proporcionar tiempo mas rápido de respuesta, procesar la información sin tiempos muertos.
  - Suave
  - Duro.
- Muchos procesos residen permanentemente en la memoria, el administrador de memoria es poco solicitado.
- Control trafico aéreo, milicia, bolsa valores, control industrial, telecomunicaciones



# CLASIFICACIÓN POR LA FORMA DEL USUARIO DE ACCEDER A LOS SERVICIOS

- Sistemas Operativos en Red
- Sistemas Operativos Distribuidos



# SISTEMAS OPERATIVOS EN RED

- Tienen la capacidad de interactuar con otros SOs en otras computadoras a través de un medio de transmisión
- Su objetivo es intercambiar información, transferir archivos, ejecutar comandos remotos, etc.
- El usuario debe conocer la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desea acceder.
- El usuario puede acceder y compartir recursos e información
- Servidor → Brindar Servicios, Compartir Recursos.



# SISTEMAS OPERATIVOS DISTRIBUIDOS

- Abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, CPUs) en una sola maquina virtual que el usuario accede en forma transparente.
- El usuario no requiere saber la ubicación de los recursos, lo conoce por nombre y los utiliza como si todos fuesen locales a su lugar de trabajo habitual.
- Un SO distribuido ideal debería:
  - Distribuir procesos en las varias unidades de procesamiento
  - Reintegrar sub-resultados
  - Resolver problemas de concurrencia y paralelismo
  - Recuperarse de fallas de algunos recursos distribuidos
  - Consolidar la protección y seguridad entre los diferentes componentes del sistemas y los usuarios.
- Fuertemente acoplados (comparten memoria, reloj global)





# MAS . . .

- Embebidos
- Móviles



# SISTEMAS OPERATIVOS MÓVILES

- Carecen de Memoria Virtual
- Numero limitado de procesos interactivos en ejecución
- Multitarea pero mono contexto (utilizados uno por uno)
  - Si caben en memoria se mantienen abiertos para evitar demoras al inicializar.
- SO debe definir políticas para finalizar programas y evacuarlos de la memoria al llegar a ciertos umbrales.
- Enérgico en cambios de perfil de energía, respondiendo a un entorno cambiante
  - Luminosidad en pantalla, Niveles críticos de carga, conexiones de red, detener procesos congelados, rotación pantalla.



# SISTEMAS OPERATIVOS EMBEBIDOS

- Operan en computadoras que controlan dispositivos que no aceptan software instalado por el usuario.
- Hornos de microondas, televisiones, autos, grabadores de DVDs, teléfonos celulares, reproductores de MP3
- A diferencia de los SOs de dispositivos de bolsillo no se puede ejecutar software que no es confiable
- No hay necesidad de proteger las aplicaciones, lo cual simplifica su estructura.

➤ QNX y VxWorks



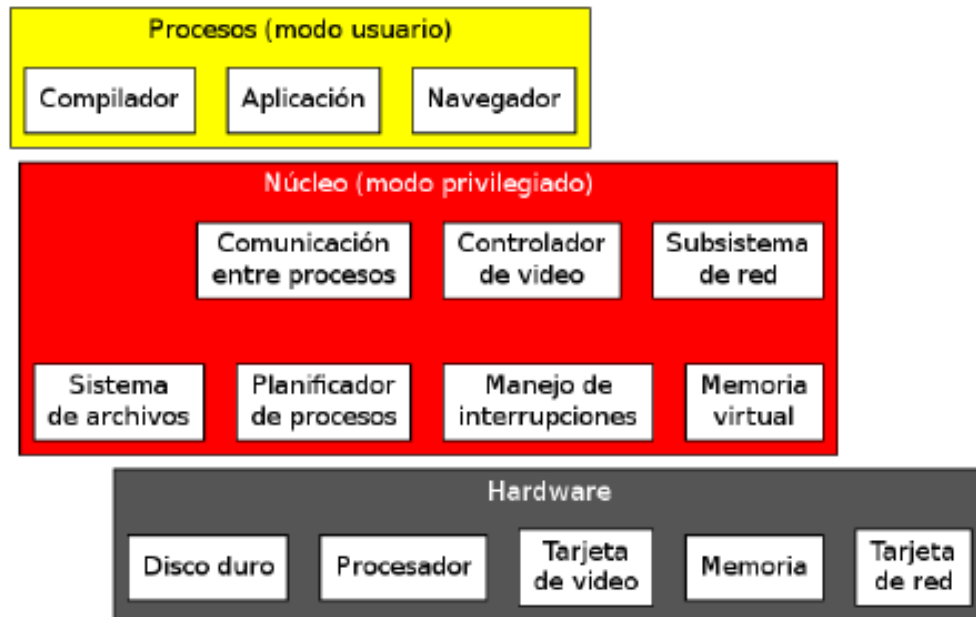
# 1.1.CLASIFICACIÓN Y ESTRUCTURAS GENÉRICAS DE LOS SISTEMAS OPERATIVOS VIGENTES

- Estructuras genéricas de los SOs “Organización Interna”
  - Modular o Monolítica
  - Jerárquica: Anillo o capas
  - MicroKernel
  - Maquinas Virtuales



# MODULAR O MONOLÍTICA

Cuentan con un solo proceso privilegiado que opera en modo supervisor, en el cual se encuentran todas las rutinas para las diversas tareas que realiza el SO



SO se ejecuta como un solo programa en modo kernel.

- Carecen de estructura
- Muchas líneas de código
- Simplifica los mecanismos de comunicación
  - Muy eficiente, las rutinas se pueden llamar entre ellas cada vez que lo requieran
- No confiable
- Pesadilla en mantenimiento

Linux, FreeBSD, Derivados UNIX, MS-DOS, Solaris



# JERÁRQUICA: ANILLO O CAPAS

- Jerarquía de capas, una construida sobre la inmediata anterior.
- Cada capa usa los servicios solo de la capa que esta por debajo
- Sistema THE (Technische Hogeschool, Eindhoven, 1968) de Dijkstra, formado por 6 capas

Capa	Función
5	El operador
4	Programas de usuario
3	Administración de la entrada/salida
2	Comunicación operador-proceso
1	Administración de memoria y tambor
0	Asignación del procesador y multiprogramación

- MS-DOS (Microsoft, 82), OS/2 (IBM, 87), NT 3.51 (Microsoft, 95)
- En la estructura de anillos cada uno tiene una apertura (puerta) por donde pueden entrar las llamadas de las capas inferiores. Las zonas mas internas están mas protegida de accesos indeseados.



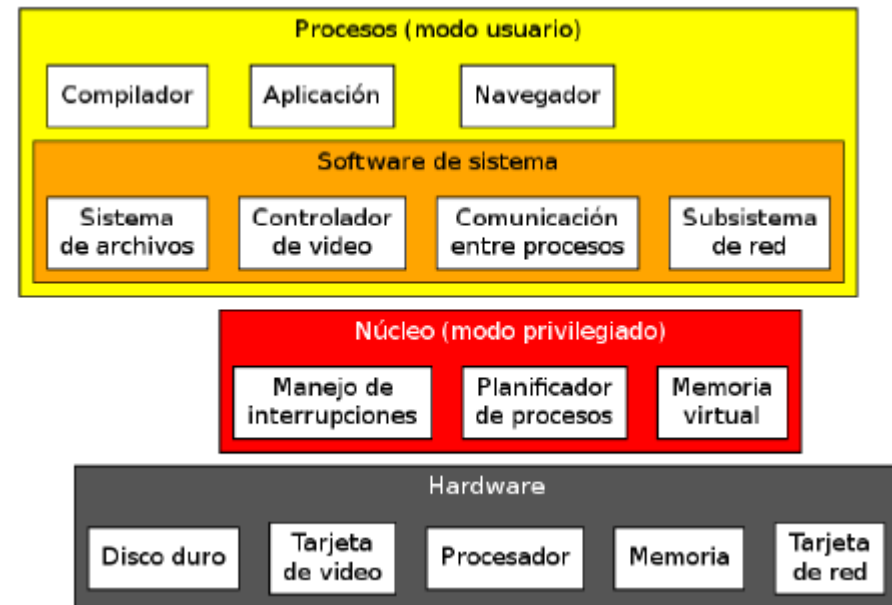


# MICROKERNEL

- Las funciones centrales se controlan por el núcleo-kernel

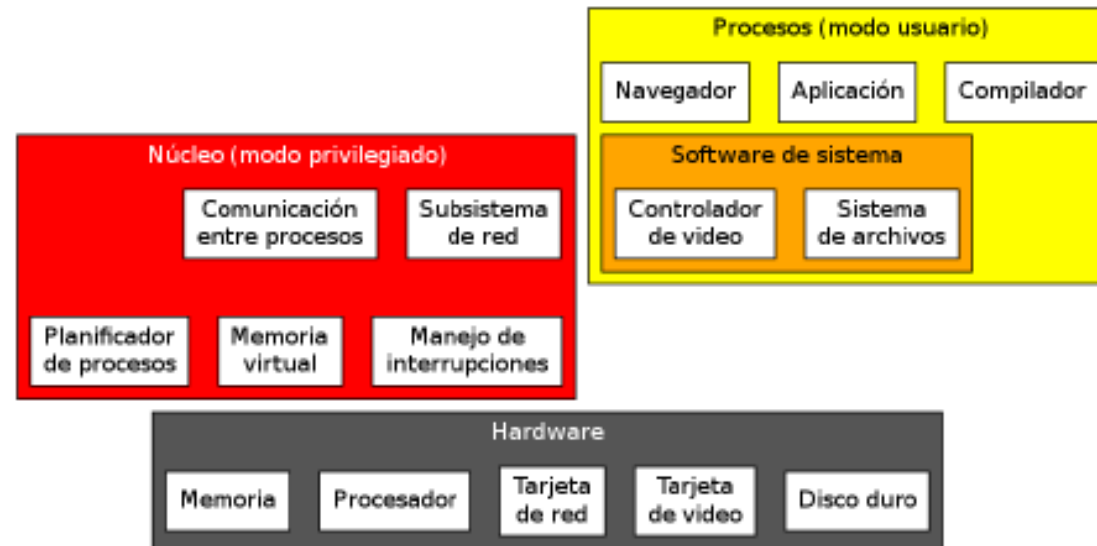
Su núcleo se mantiene en el mínimo posible de funcionalidad, descarga en procesos especiales sin privilegios algunas tareas específicas como: acceso a dispositivos y las diversas políticas de uso del sistema

- Diseño Modular
  - Facilita su comprensión
- Pocas líneas de código en el núcleo
- Una falla en un servicio no afecta a los demás
- Si falla un componente se puede autoreparar
  - El núcleo puede reiniciarlo o reemplazarlo



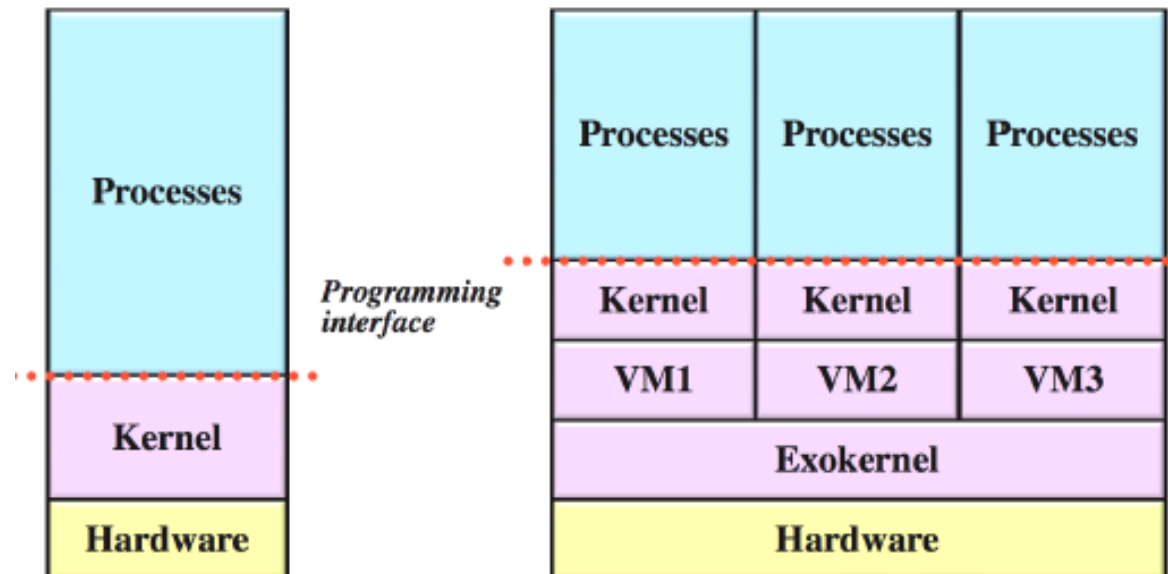
# HIBRIDOS

- Algunos sistemas son mayormente monolíticos, pero manejan algunos procesos de nivel usuario como microkernel.
  - Ej. Sistema de archivos en espacio de usuario.

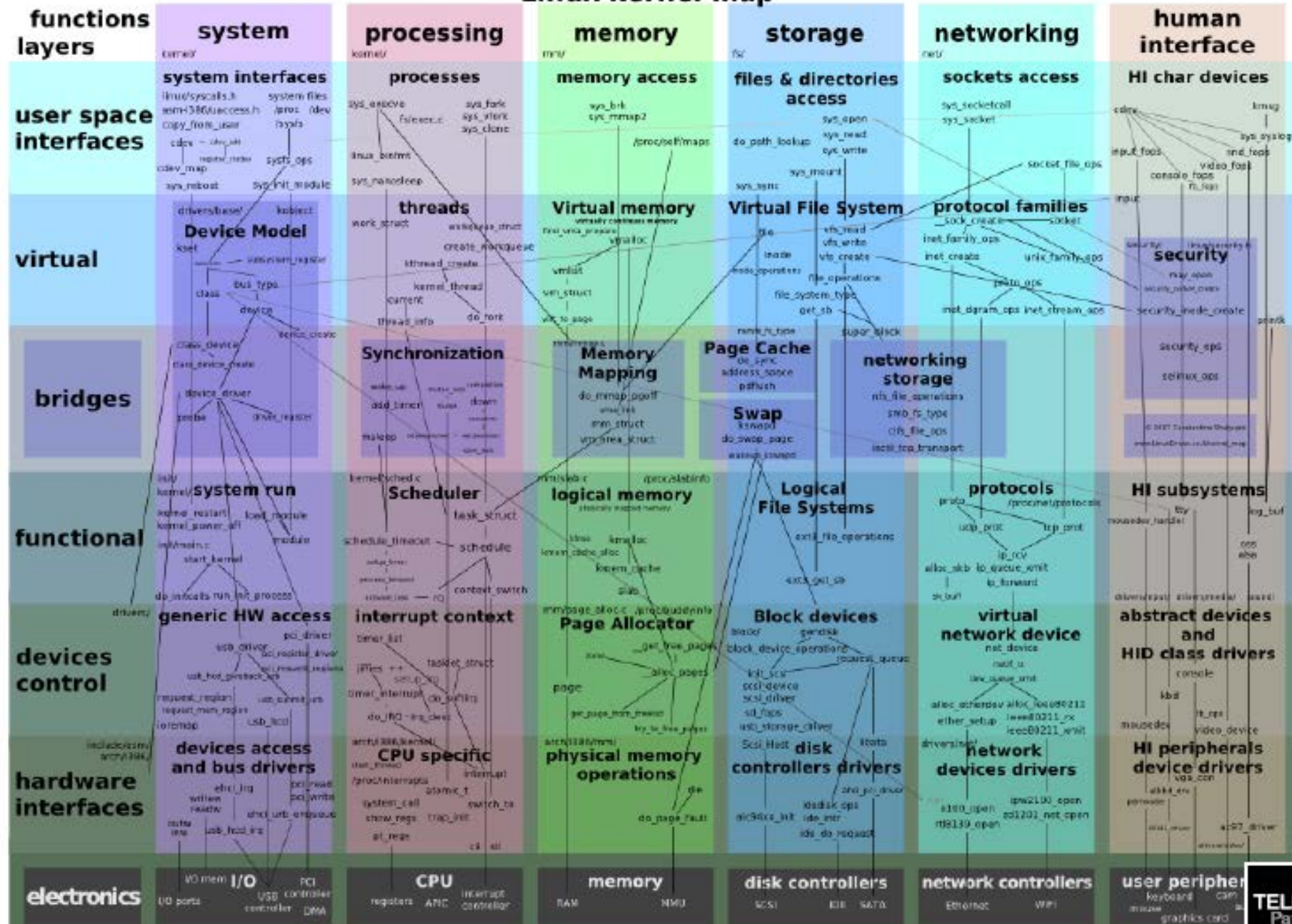


# MAQUINAS VIRTUALES

- MV es una copia exacta de la maquina real, generada por software.
- Los SOs proporcionan al usuario una copia exacta del hardware incluyendo sus modos usuario y núcleo, dispositivos E-S, interrupciones, almacenamiento, etc.

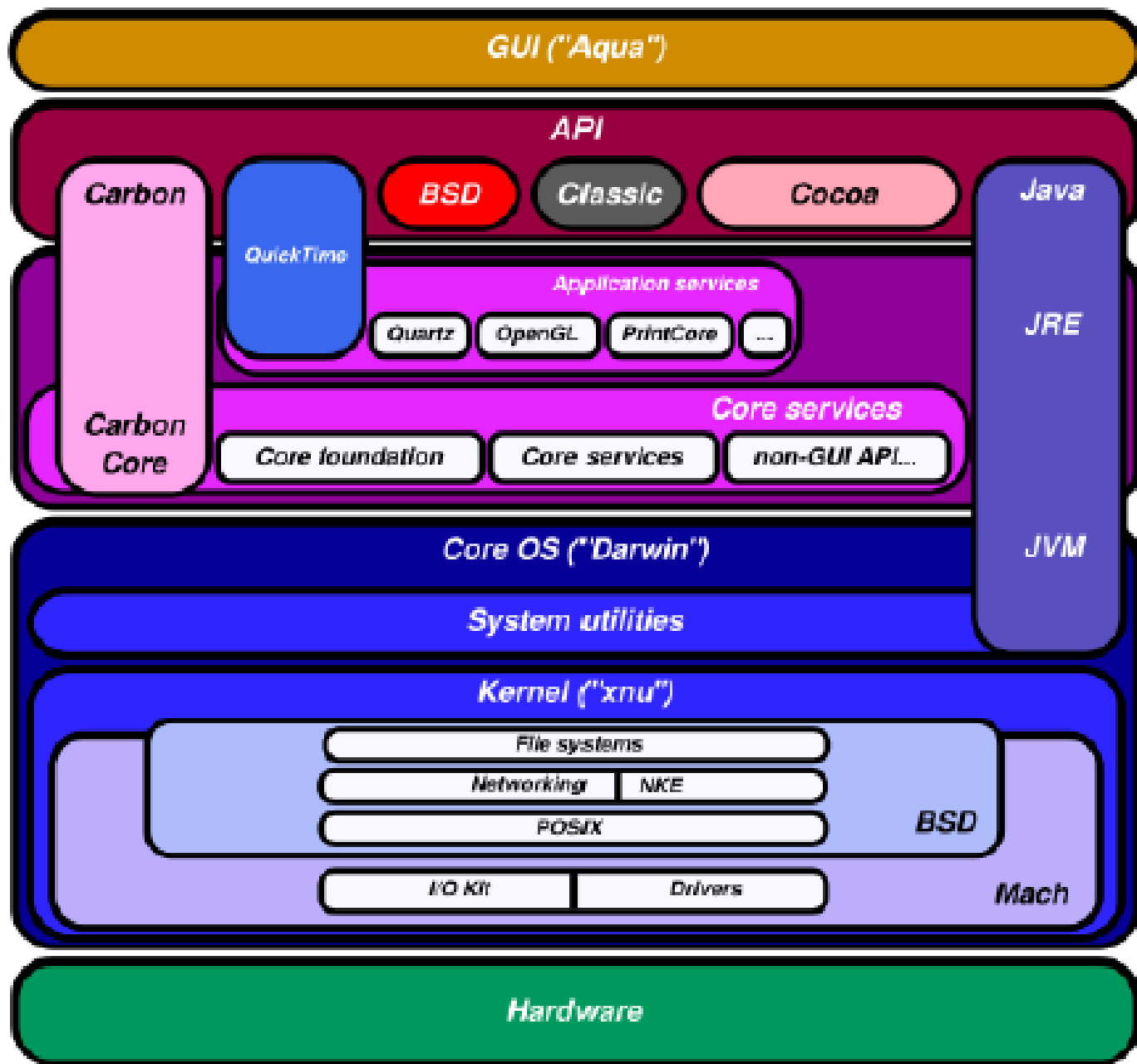


## Linux kernel map



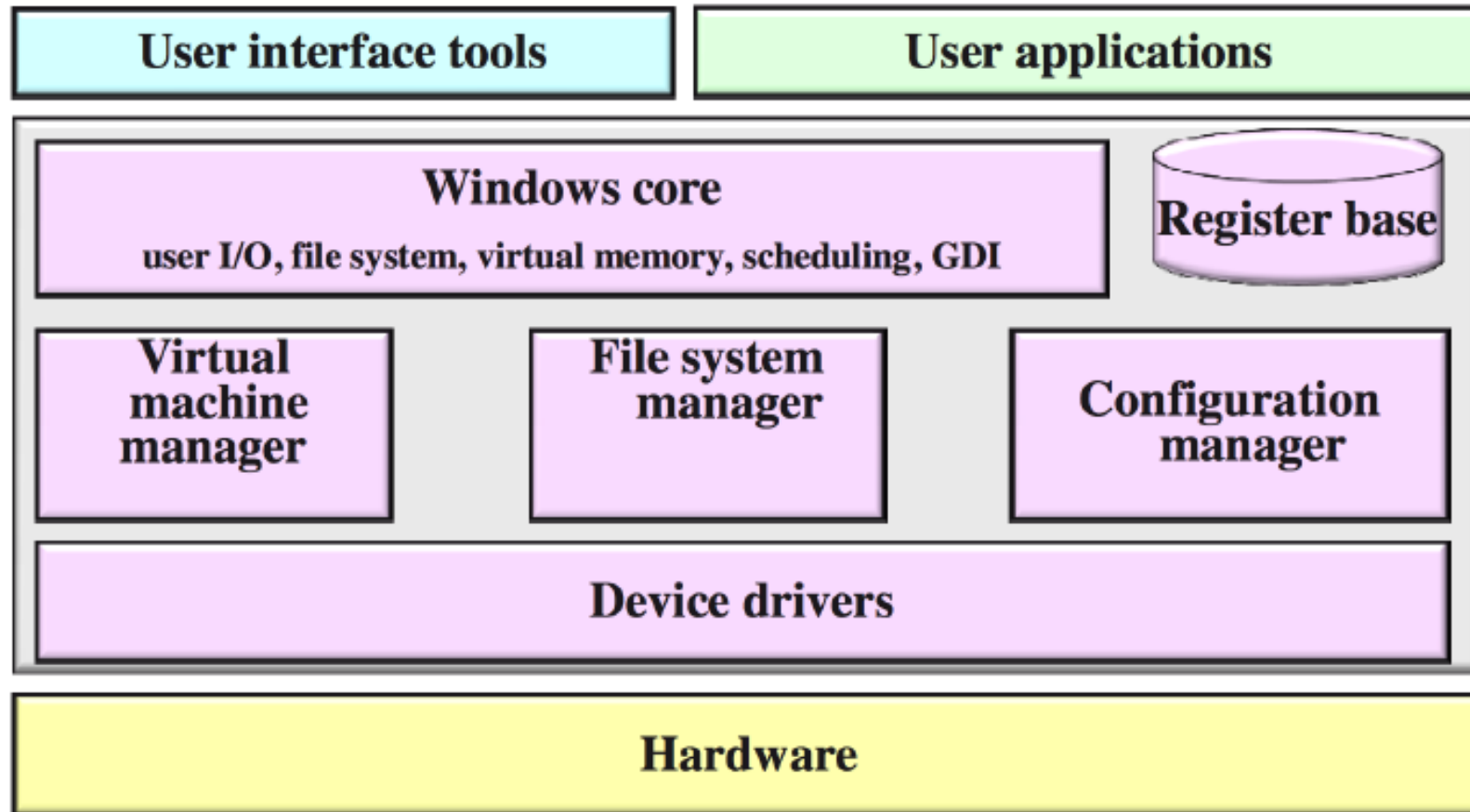




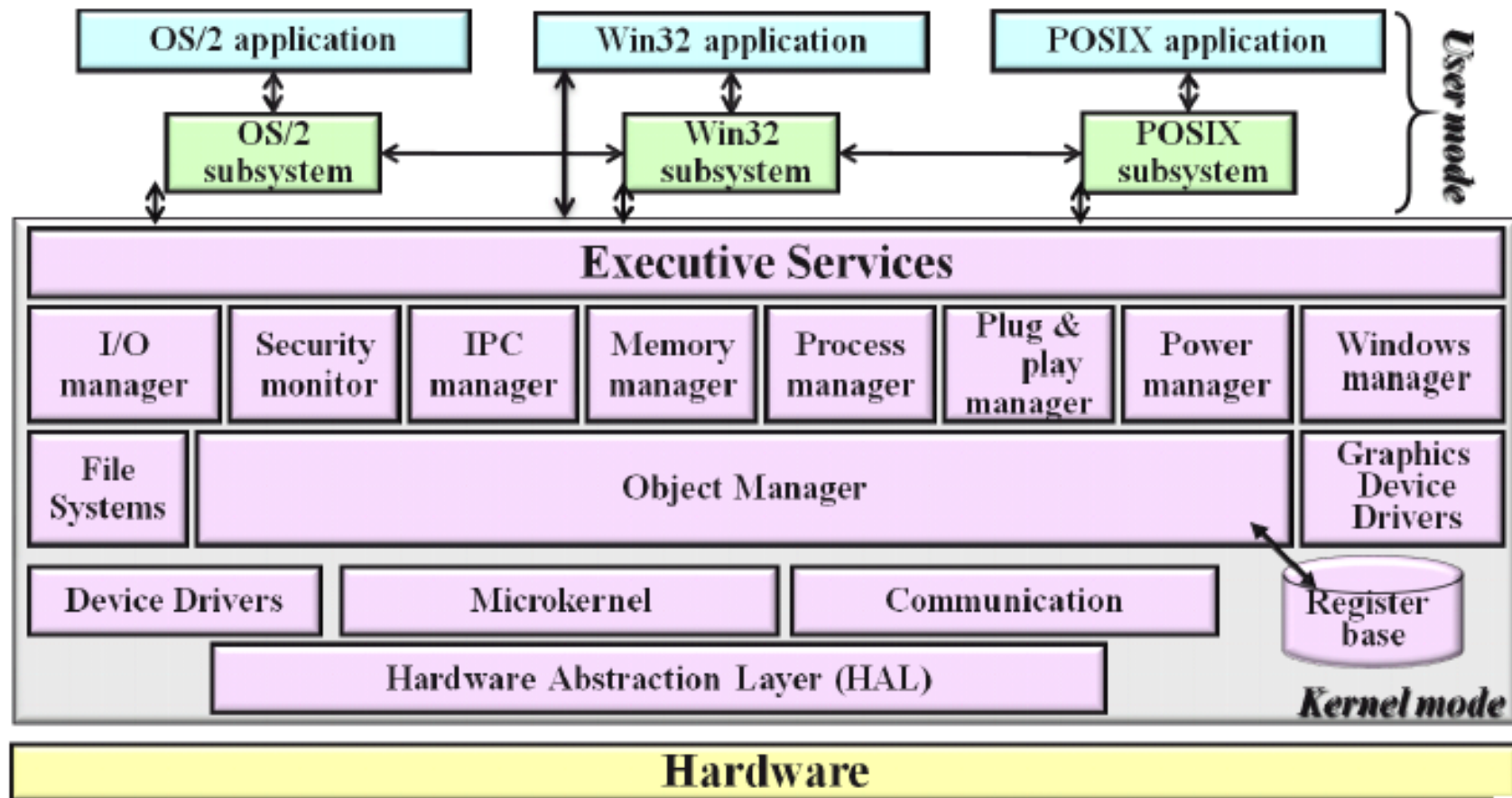




# WINDOWS 95



# WINDOWS NT ARCHITECTURE (2000/XP/VISTA/7/8)



# 1.2. PROCESOS Y MULTIPROGRAMACIÓN

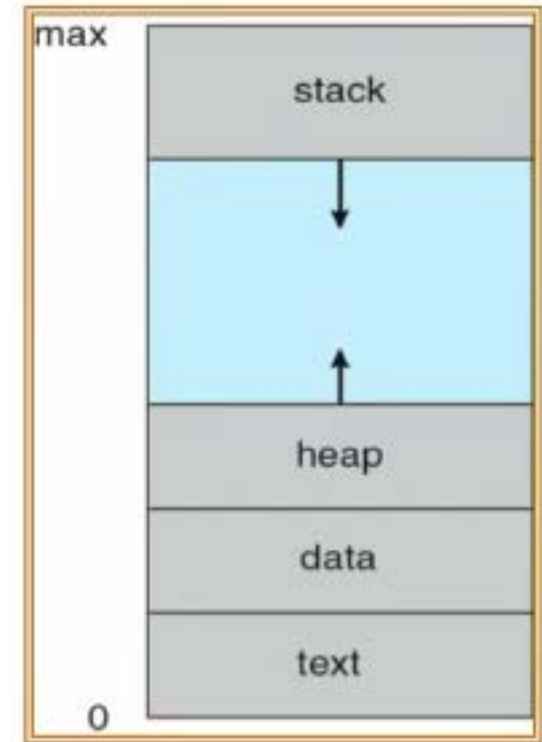
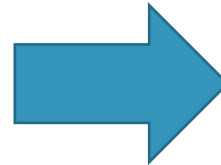
Programa	Proceso
Es estático	Es dinámico
No tiene contador de programa	Tiene contador de programa
Existe desde que se edita hasta que se borra	Ciclo de vida comprende desde que se dispara hasta que termina

- Programa
  - Archivo con comando o instrucciones (inactivo ) almacenado en disco.
- Proceso o Tarea
  - Instancia o abstracción de un programa en ejecución
  - Entidad activa que requiere un conjunto de recursos
  - Necesita espacio en memoria principal donde permanece durante su ejecución, puede leer y escribir información, además puede requerir de otros recursos como archivos de datos o dispositivos de E/S



# 1.2. PROCESOS Y MULTIPROGRAMACIÓN

- Asociado a cada proceso un conjunto de recursos
  - Registros (Contador de programa, apuntador de pila...)
- Lista archivos abiertos, alarmas pendientes, listas de procesos relacionados
- Espacio de direcciones
  - Lista de ubicaciones de memoria que va desde algún mínimo (0) hasta cierto valor máximo, donde el proceso puede leer y escribir información.
  - Código, Variables globales, Llamada a funciones (stack), asignación de memoria (heap)



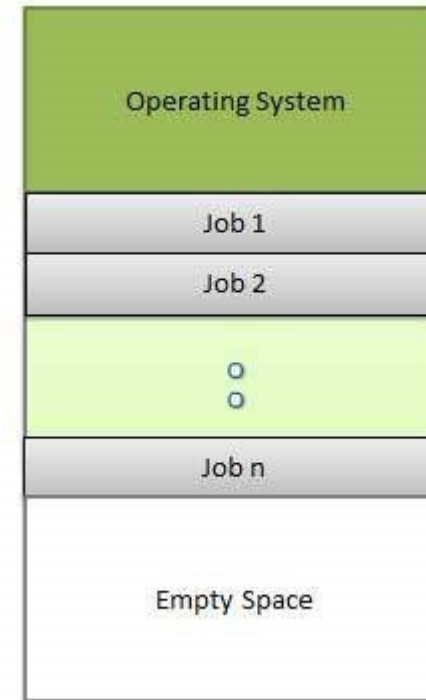
# 1.2. PROCESOS Y MULTIPROGRAMACIÓN

## Multiprogramación

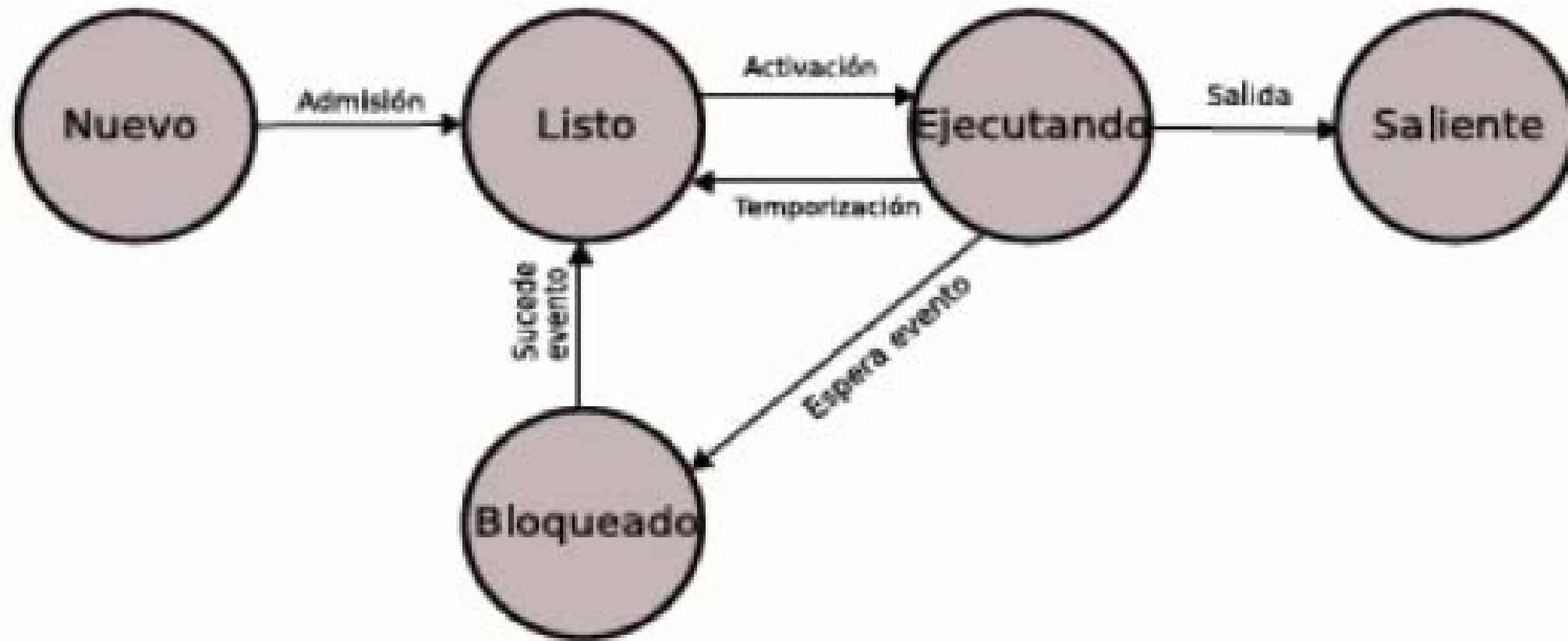
Permite que un solo procesador procese varios procesos que residen simultáneamente en la memoria principal

Un proceso puede pasar por varios estados (listo, ejecución, bloqueado)

- Planificación
  - Algoritmos de planificación
- Comunicación interprocesos
  - Intercambio de datos, sincronización



# ESTADOS Y TRANSICIONES DE LOS PROCESOS



# PCB PROCESS CONTROL BLOCK

## BLOQUE DE CONTROL DE PROCESOS

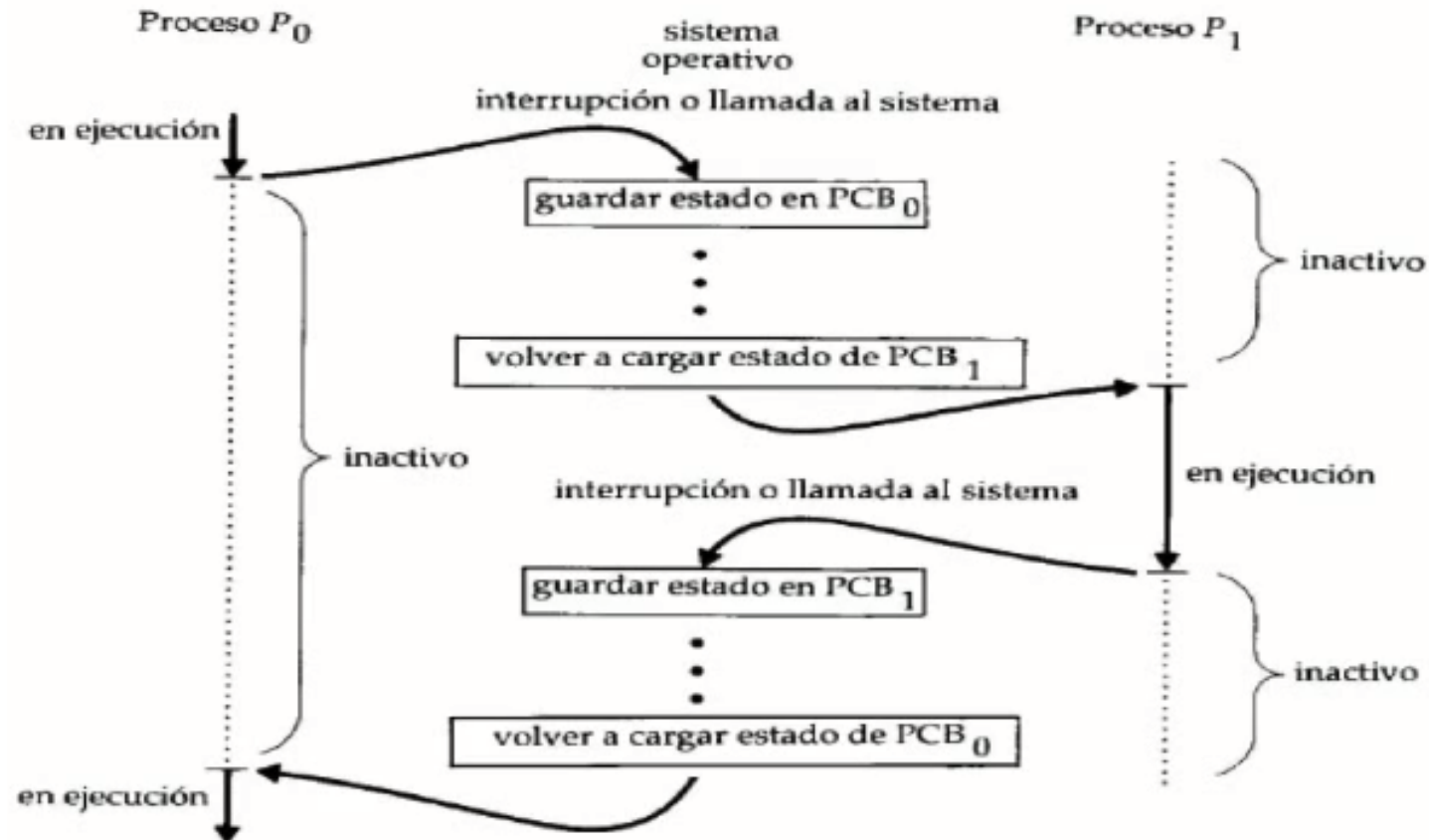
Estructura de datos PCB: Cada bloque contiene elementos de información asociados a un proceso específico.

- Información de Gestión de Memoria
  - Registros base y límite, tablas de páginas o segmentos
- Estado del proceso
- Contador de programa
- Registros de CPU
- Información de planificación de CPU
  - Prioridad, punteros a colas de planificación
- Información contable
  - Tiempo CPU, límites, etc
- Información de estado de E/S
  - Dispositivos E/S asociados, archivos abiertos



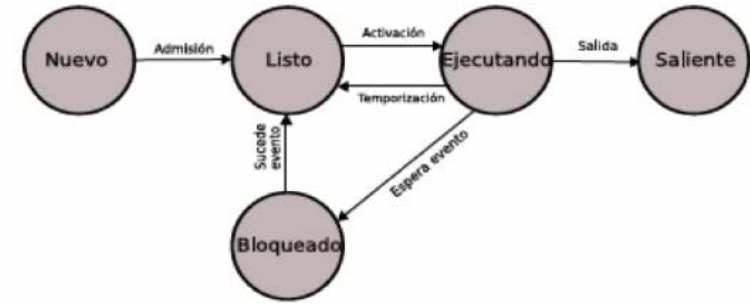
# CAMBIO DE CONTEXTO

Guarda información del estado de procesamiento de un proceso al momento de retirarlo de la CPU y carga la información de otro proceso para que la CPU pueda procesarlo





# CONTROL DE LA MULTITAREA



Planificador a **corto plazo** puede ser invocado cuando:




1. Pasa de estar ejecutando a estar bloqueado (por ejemplo, por solicitar una operación de E/S, esperar a la sincronización con otro proceso, etcétera).
2. Pasa de estar ejecutando a estar listo (por ejemplo, al ocurrir la interrupción del temporizador, o de algún evento externo).
3. Deja de estar bloqueado a estar listo (por ejemplo, al finalizar la operación de E/S que solicitó).
4. Finaliza su ejecución, y pasa de ejecutando a terminado.

**1 y 4 el SO  
siempre toma  
el control**

- **Multitarea preventiva o apropiativa** (Preemptive multitasking)
  - el SO implementa el 2 y 3
- **Multitarea cooperativa o no apropiativa** (Cooperative multitasking)
  - El SO no necesariamente implementa 2 y 3



# CARACTERIZACIÓN DE MECANISMOS DE PLANIFICACIÓN A CORTO PLAZO

	NO CONSIDERA INTRÍNSECA	CONSIDERA INTRÍNSECA	
COOPERATIVA	Primero llegado primero servido (FCFS)	Proceso más corto (SPN), Proceso más penalizado (HPRN)	<div>            P=1, si aumenta tiempo de espera aumenta P         </div>
PREVENTIVA	Ronda (RR) Lotería (Boletos)	Proceso más corto apropiativo (PSPN), Retroalimentación (FB), Ronda egoísta (SRR)	<div>  Colas de prioridad   Favorece procesos aceptados sobre procesos nuevos         </div>

Considera intrínseca: Si un proceso es tratado de distinta forma dependiendo de su historial de ejecución

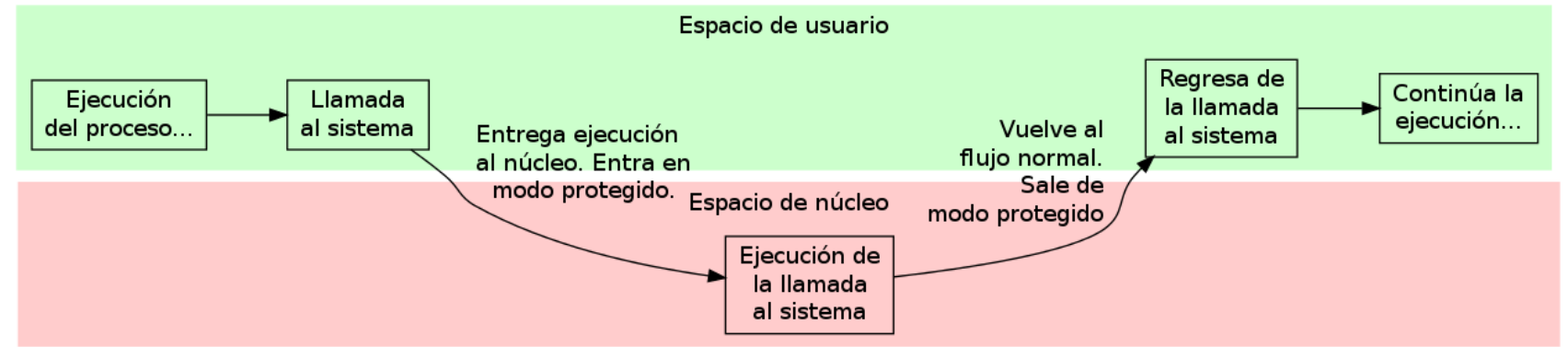


# NÚCLEO

- Código diseñado para que sea ejecutado mientras el procesador esta en modo núcleo
- Rutina de biblioteca, en la que uno o mas procedimientos en ella ejecutan siguiendo una excepción (trampa), una interrupción o una llamada al sistema.
- En algún punto el procesador conmuta a modo usuario y el control retorna a un programa de usuario (puede ser o no el mismo que estaba ejecutando cuando se ingreso a modo núcleo)
- Las 4 áreas de administración del SO están implementadas en el núcleo: Procesador, Memoria, Dispositivos, Archivos.



# LLAMADA AL SISTEMA



- Se realiza cuando un proceso requiere acceso a alguna acción privilegiada : Interfaz entre un programa en ejecución (procesos) y el SO
- Una llamada al sistema la invoca un proceso de usuario (un proceso en modo usuario) y es servida por el núcleo
- Implica pasar o saltar del código del usuario al código del núcleo → cambio en el modo de funcionamiento del procesador
- Un SO expone una serie de llamadas al sistema a través de una API
- Ensamblador (MS-DOS), Alto Nivel como C (UNIX, Minix, Linux y Windows NT)
- Modos de operación del microprocesador (usuario y núcleo)
  - Un único bit en el PSW(Program Status Word) registra el modo de ejecución del sistema.



# LLAMADAS AL SISTEMA

- **Control de procesos** Crear o finalizar un proceso, obtener atributos del proceso, esperar la finalización de un proceso o cierto tiempo, asignar o liberar memoria, etcétera.
- **Manipulación de archivos** Crear, borrar o renombrar un archivo; abrir o cerrar un archivo existente; modificar sus metadatos; leer o escribir de un descriptor de archivo abierto, etcétera.
- **Manipulación de dispositivos** Solicitar o liberar un dispositivo; leer, escribir o reposicionarlo, y otras varias. Muchas de estas llamadas son análogas a las de manipulación de archivos, y varios sistemas operativos las ofrecen como una sola.
- **Mantenimiento de la información** Obtener o modificar la hora del sistema; pedir detalles acerca de procesos o archivos, etcétera.
- **Comunicaciones** Establecer una comunicación con determinado proceso (local o remoto), aceptar una solicitud de comunicación de otro proceso, intercambiar información sobre un canal establecido.
- **Protección** Consultar o modificar la información relativa al acceso de objetos en el disco, otros procesos, o la misma sesión de usuario.



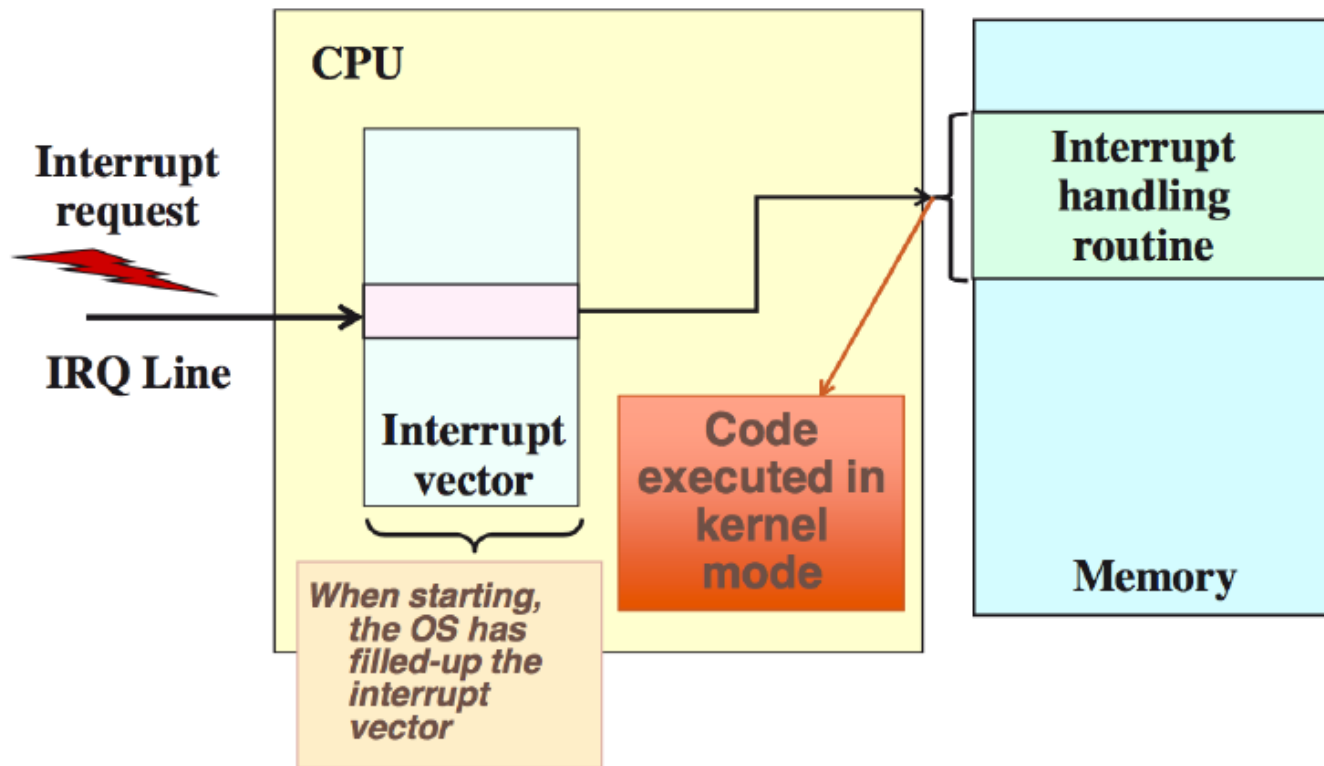
# INTERRUPCIÓN

- Hardware, generada por causas externas al sistema
  - Dispositivo que requiere atención
- Señal asincrónica que indica la necesidad de atención (mediata o inmediata), no depende del programa en ejecución, es externa a el.
  - Ejecución de funciones de administración del dispositivo en modo núcleo



# INTERRUPCIÓN

- Polling
- Mecanismo de interrupciones (IRQ, Control de interrupciones (HW), Vector de interrupciones (rutinas))



# EXCEPCION O TRAMPA

- Software, internas al proceso
- Tipo de interrupción sincrónica provocada típicamente por una condición excepcional (división por cero, acceso a memoria invalido) en procesos hilo o en modo usuario.
- Provoca un cambio a modo núcleo, el SO debe realizar alguna acción antes de devolver el control al proceso que la origino





# 1.3. VIRTUALIZACIÓN

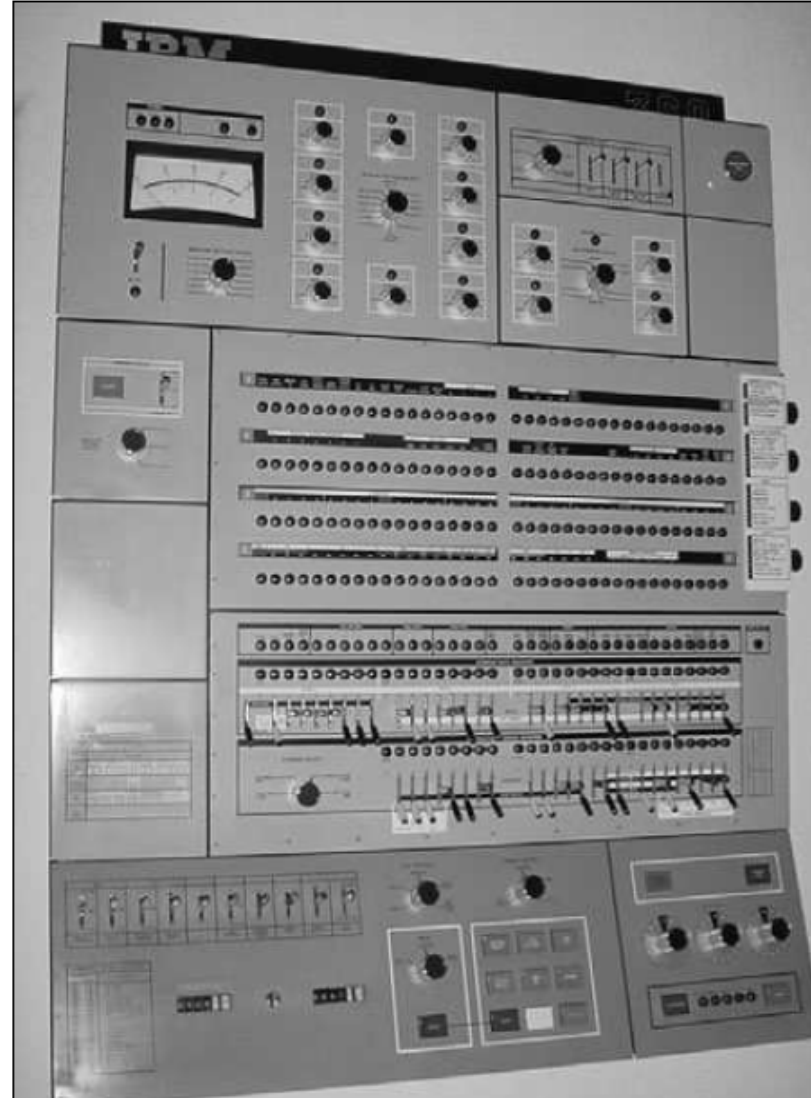
## 1.3.1. Componentes y Niveles de Virtualización

Virtualizar: Proveer algo que no esta ahí, aunque parezca estarlo



Servidor con un aplicación de facturación en un mainframe

- Actualizar datos con cierta frecuencia (10%) y a fin de mes un uso completo (+100%)
- ❖ Hardware desaprovechado!



# VIRTUALIZACIÓN

## VM/370

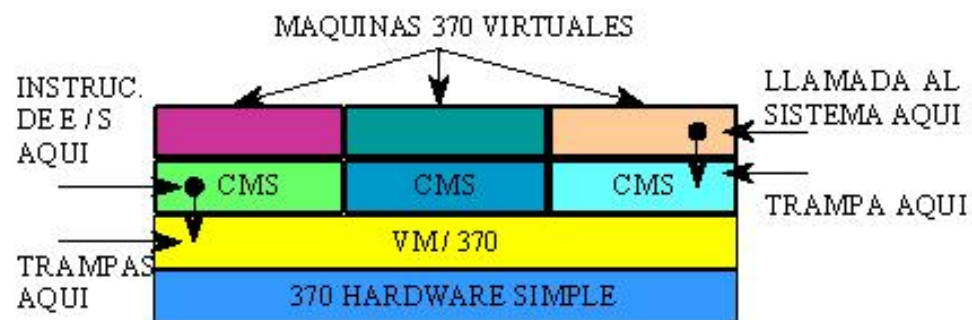


Figura 1.4: La estructura de VM/370 con CMS.

- Surgió en 1967 IBM
- IBM S/360-67 Mainframe para dar soporte a virtualización
- Sistema Operativo CP/CMS (Programa de Control), primer sistema de máquina virtual, Software Libre
  - Permitía crear y gestionar MVs en el hardware, dentro de cada una (CMS, Sistema de Monitoreo Conversacional) se podía ejecutar un SO estándar de la serie 360 sencillo, interactivo, monousuario OS/360
  - El resultado un SO multiusuario de tiempo compartido, con protección entre procesos y compatible con modelos de la serie 360
- CP Multiplexador de hardware . Se ejecuta en el hardware y realiza la multiprogramación, proporciona varias máquinas virtuales a la capa de arriba



# Z/VM

- Actualmente: z/VM en la línea de Sistemas z
- Utilizado en los mainframes de IBM (Zseries)
- Ejecuta SOs completos
  - Unas o mas maquinas virtuales de Linux
  - SOs tradicionales de IBM
- Centros de datos de corporativos extensos
  - Comercio electrónico: transacciones, BDs



# NUEVAS NECESIDADES

- Múltiples servidores con diferentes SOs ejecutando en la misma maquina, evitando que la falla de uno dañe al resto
- Hospedaje Web: Compartido (no control sobre el software), Dedicado (maquina propia, \$)
- Dos o mas SOs al mismo tiempo, para ejecutar aplicaciones exclusivas



# VIRTUALIZACIÓN

- 2005, Intel procesadores con extensión Vanderpool Technology (VT/X), AMD-V extensión Pacífica
- Pone una capa de tecnología entre el hardware y el SO, fabrica hardware virtual para que el SO crea que esta en un servidor físico.
- Hardware virtual es independiente de cualquier plataforma, siempre utiliza los mismos drivers.
- DEFINICION: Creación de abstracciones de dispositivos o recursos físicos con el fin de ofrecerlos a uno o más entornos de ejecución
- SO es el supervisor de la actividad de los usuarios
  - **Hipervisor seria el programa que administra y virtualiza a los supervisores.**



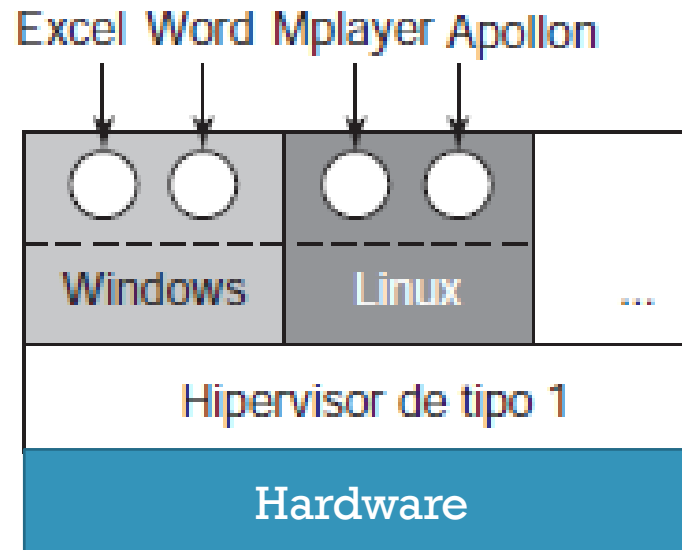
# HIPERVISOR: MAS ABAJO QUE EL NÚCLEO

- Tradicionalmente las arquitecturas virtualizables corren un micro-sistema operativo encargado de gestionar a cada uno de los sistemas operativos que corre en cada una de las máquinas virtuales
- Es un micro SO porque no cubre muchas de las áreas clásicas (sistemas de archivos, comunicación entre procesos, gestión de memoria virtual, ...)
- Se limita a gestión básica de memoria física contigua, asignación de dispositivos, y poco mas que eso
- No tiene interfaz de usuario directo, es administrado por medio de llamadas privilegiadas desde algún SO huésped.
- Ojo: Hay hipervisores que son sistemas operativos completos (como KVM bajo Linux)
- Este micro-SO es conocido como el hipervisor



# HIPERVISOR DE TIPO 1

- Monitor de maquina Virtual → Hipervisor de Tipo 1

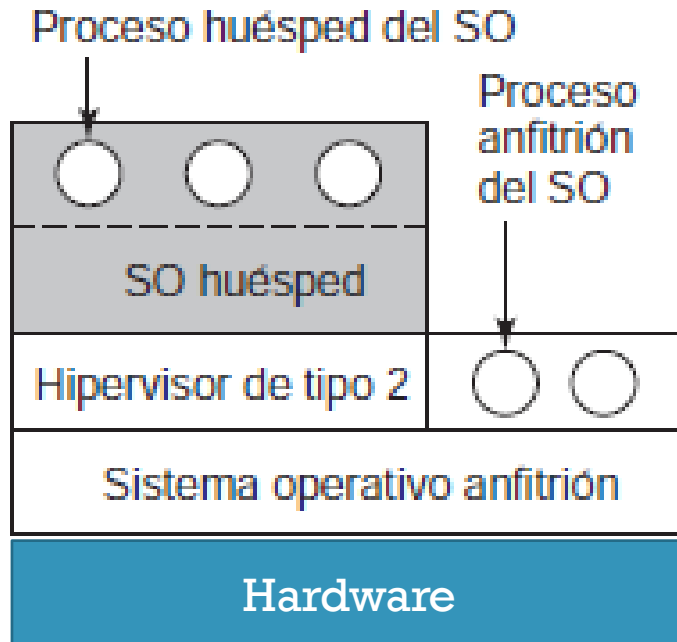


Si un SO que opera en una MV ejecuta una instrucción privilegiada (Ej. E/S), debe ser atrapada por el HW para el monitor de la MV, de manera que la instrucción se pueda emular en SW.

CPU debe ser virtualizable, algunas operaciones E/S o privilegiadas en modo usuario se ignoraban



# HIPERVISOR DE TIPO 2



- Se ejecutan como programas de aplicación encima de un SO anfitrión
- El SO huésped se instala en un disco virtual (gran archivo en el Sistema de archivos del SO anfitrión)





# COMO IMPLEMENTAR HVM

- Proyectos Libres
  - Xen (modo asistido por hardware)
  - KVM (sobre Linux)
  - Logical Domains (sobre Solaris)
- Proyectos híbridos libre/ propietario
  - Virtual Box
- Proprietarios
  - VMWare
  - VirtualPC Microsoft
  - HyperV Microsoft
  - Parallels Bare Metal



# VENTAJAS DE LA VIRTUALIZACIÓN

- Consolidación de Servidores: Reducir su numero y aumentar el porcentaje de utilización.
- Aislamiento: Si falla un SO de una MV, los demás no se afectan
- Ahorro de energía
- Ahorro de espacio
- Redundancia /alta disponibilidad (replicas)
- Balanceo dinámico de MVs, garantizando un consumo de recursos optimo y homogéneo
- Agilidad en la administración /mantenimiento
- Seguridad
- Migración (en caliente)
  - Se pueden transportar fácilmente entre distintos equipos físicos
- Compatibilidad hacia atrás
- Desarrollo (depuración) para sistemas embebidos



# VPS (VIRTUAL PRIVATE SERVER)

## SERVIDORES PRIVADOS VIRTUALES

- Hosting compartido
- Servidor dedicado    \$\$\$\$
- VPS
  - Mas barato que un dedicado
  - Instalación y configuración personalizada
  - No compartir recursos .... Mejor rendimiento
  - Menos limitantes en almacenamiento

