

Introducción, Principios de Diseño y Estructuras de los Sistemas Operativos

Sistemas Operativos Avanzados

Prof. David A. Pérez A.

david.perez@ciens.ucv.ve

Introducción SOA

- Definición SO.
 - Administrador de recursos.
 - Capa de software.
 - Nuestra propia definición.
- Motivación para utilizar un SO.
 - Máquina al desnudo.
 - Conjunto de instrucciones reducido.
 - Ampliar el conjunto de instrucciones.

Introducción SOA

- Motivación para utilizar un SO.
 - Soporte multiusuario.
 - Liberar al usuario
 - ¿De qué?
- ¿Dónde consigo un SO?

Introducción SOA

- Funciones de un SO.
 - Administración de recursos.
 - Administración del tiempo.
 - Administración del espacio.
 - Manejo de sincronización de procesos.
 - Deadlock.
 - Manejo de estadísticas y estado del sistema.

Introducción SOA

- Funciones de un SO.
 - Interfaz amigable
 - ¿Qué esconde la interfaz?
 - Ambiente de ejecución.
 - Administración de procesos-creación, control y finalización.
 - Manipulación de archivos.
 - Manejo de interrupciones.
 - Soporte de E/S.
 - Detección y manejo de errores.

Introducción SOA

- Funciones de un SO.
 - Interfaz amigable.
 - Protección y seguridad.
 - Tolerancia a fallas y recuperación de las mismas.

Introducción SOA

- Conceptos básicos.
 - ¿Qué debo proveer?
 - Abstracción de proceso.
 - Abstracción de hilo.
 - Sincronización.
 - Manejo de dispositivos.
 - Discos, etc.
 - Manejo de memoria.
 - Principal y secundaria.
 - Sistema de archivos.

Enfoques de Diseño

- ¿Cuál es la idea natural?
 - La forma “burrera”, en verdad macro.
 - Gran colección de procesos sin estructura.
 - ¿Qué podría suceder al ejecutar una instrucción?
 - Efecto domino.

Enfoques de Diseño

- ¿Cuál es la idea natural?
 - La forma “burrera”, en verdad macro.
 - Gran colección de procesos sin estructura.
 - ¿Qué podría suceder al ejecutar una instrucción?
 - Efecto domino.

Enfoques de Diseño

- Ventajas del enfoque macro.
 - Propósito específico.
 - Sistemas pequeños.
- Desventajas del enfoque macro.
 - Poca factibilidad de expansión.
 - Fuertes complicaciones.
 - Código, pruebas y depuración.

Políticas vs. Mecanismos

- Políticas.
 - Que hacer.
- Mecanismos
 - Como hacer.
- No confundir
- Ejemplos
 - Planificación de procesos.
 - Funcionamiento de un restaurant.

Políticas vs. Mecanismos

- ¿Quién realiza dicha separación?
- Ventajas.
 - Flexibilidad.
 - Independencia.
- ¿Qué cambia?
 - Las políticas.
 - Alto nivel.
 - En ciertas ocasiones los mecanismos.
 - Bajo nivel → ¿Implicaciones?

Refinando Conceptos

- ¿Por qué surge el hilo?
 - Planificación.
- ¿Ambientes idóneos para usar hilos?
 - Procesos orientados a E/S.
 - Multiprocesadores.
 - Niveles.
 - Usuario/Librerías.
 - Kernel.

Refinando Conceptos

- Granularidad.
 - Frecuencia de sincronización.
 - Categorías de paralelismo:
 - Independiente.
 - Grano grueso.
 - Grano muy grueso.
 - Grano medio.
 - Grano fino.

Refinando Conceptos

Tamaño del Grano	Intervalo de Sincronización
Fino	<20
Medio	20 – 200
Grueso	200 – 2000
Muy Grueso	2000 – 1M
Independientes	N/A

Hyperthreading

- Tecnología propietaria de Intel.
- Por cada procesador físico presente, el SO direcciona o mapea dos procesadores virtuales.
 - Compartición de la carga de trabajo.
- Sólo se requiere que el SO posea soporte para múltiples procesadores.

Hyperthreading

- Mejora el desempeño de aplicaciones con hilos.
- Desempeño de un 155 a un 30 % mejor que sus predecesores.
- Su funcionamiento esta basado en duplicar ciertas secciones del procesador.
 - Registros de control (EFLAGS, IMR, MMU registers, etc.)
 - Registros de propósito general (AX, BX, CX, DX, etc.)

Hyperthreading

- No duplica los recursos de ejecución principales.
 - Ideas.
- Procesadores lógicos.
 - Planificación de hilos de manera simultanea.
- Punto débil.
 - Consumo de energía.
 - ARM con SMT usaba 46% menos de energía.

Multi-core

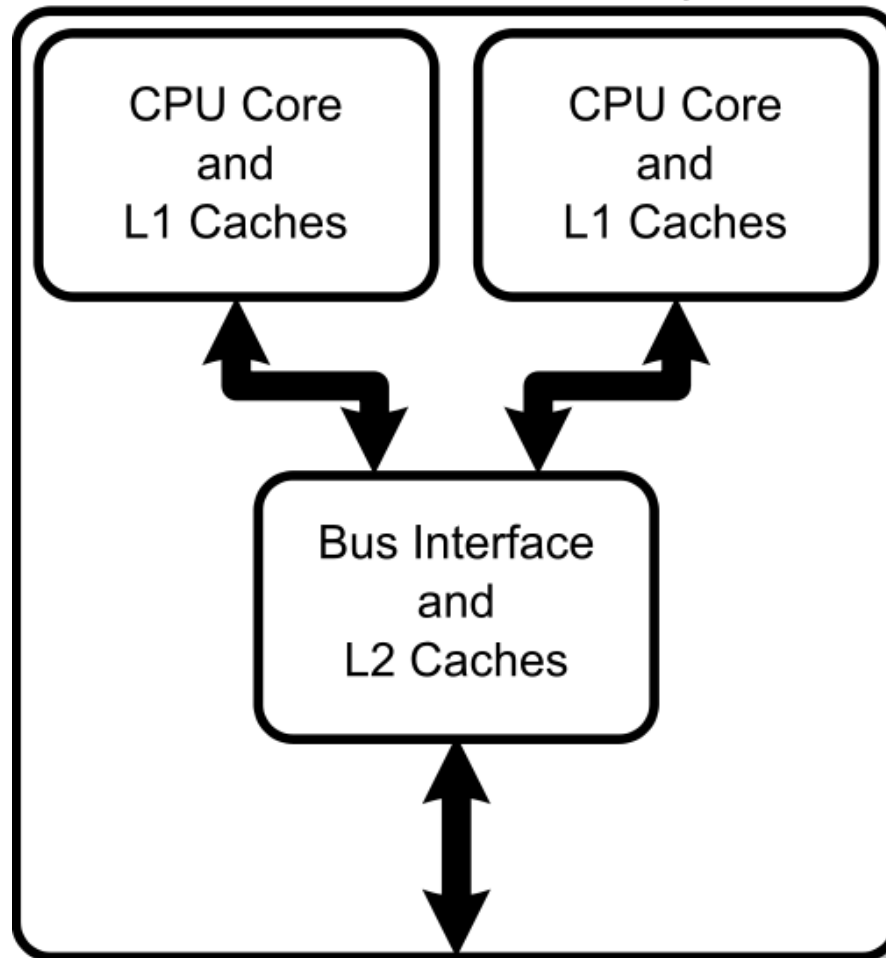
- Sistema de procesamiento compuesto por dos o más cores independientes.
- Circuito integrado que posee dos o más procesadores individuales incrustados.
- La mejora en cuanto al desempeño depende fuertemente del diseño e implementación de software.
 - Nuevo paradigma para los desarrolladores.

Multi-core

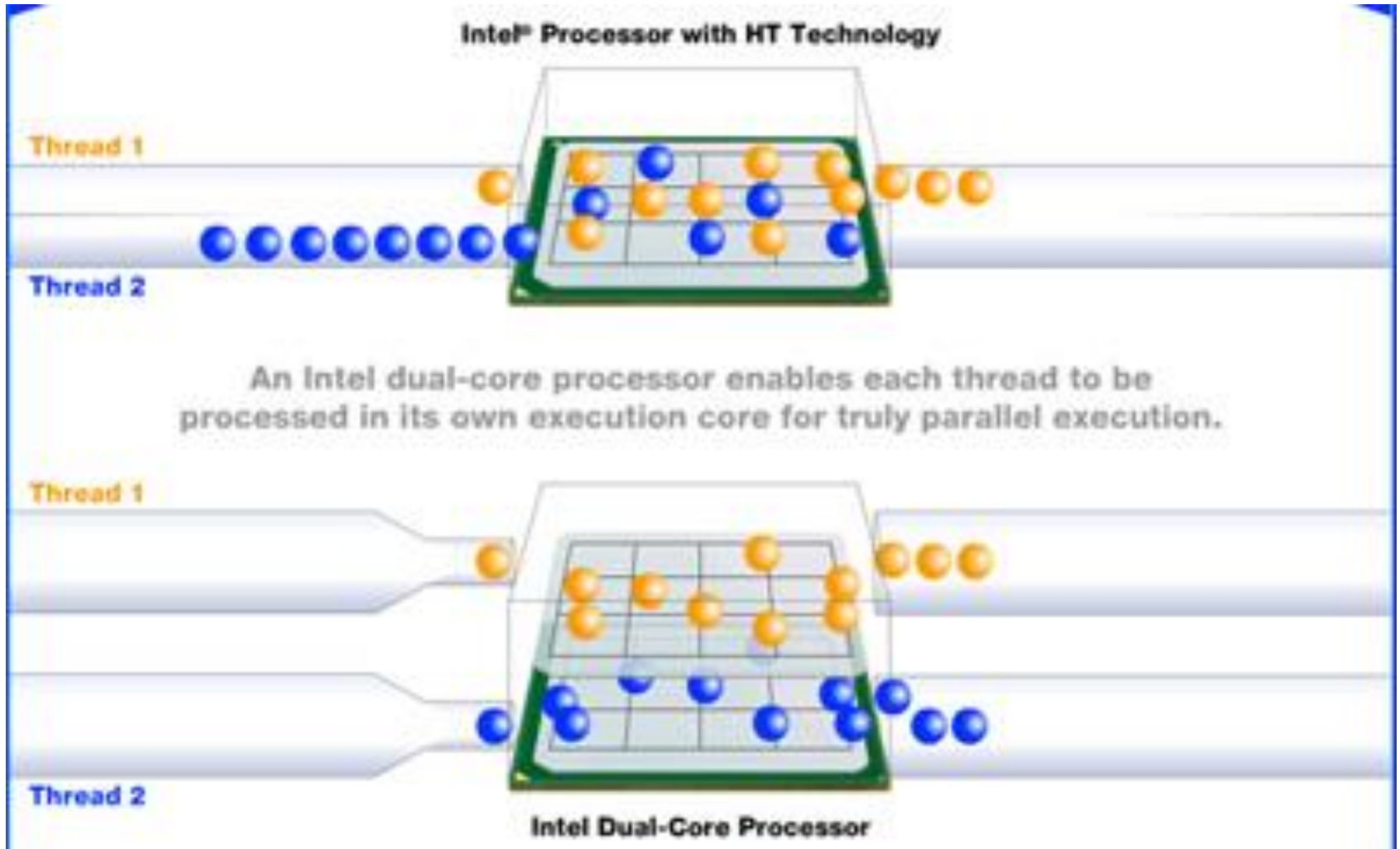
- Mejoras en el consumo de energía.
- ¿Se desecho hyperthreading?
 - Ideas.

Multi-core

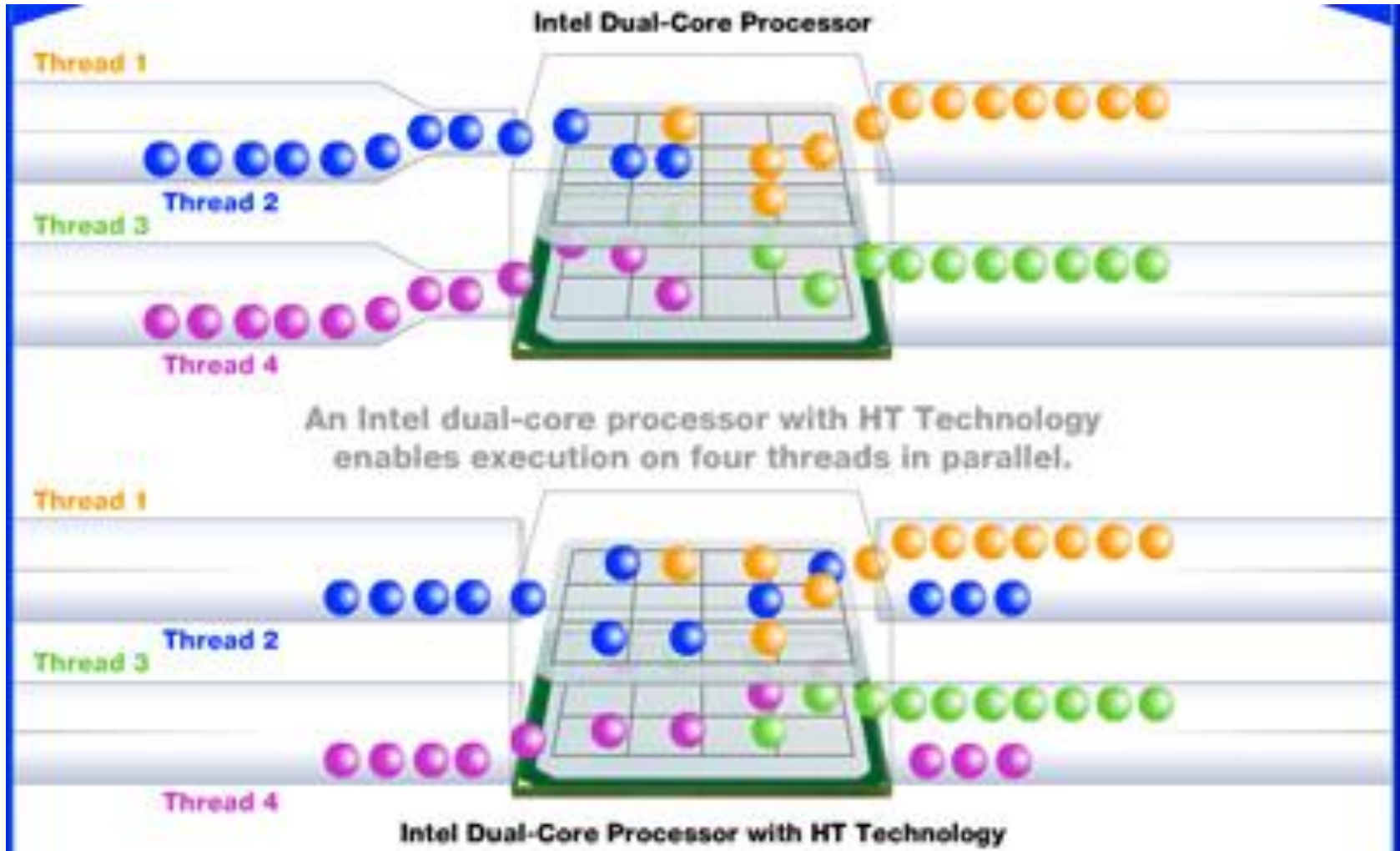
Dual CPU Core Chip



Hyperthreading vs. Multi-core



Multi-core + Hyperthreading



Refinando Conceptos

- Reentrancia.
 - Programas reentrantes.
 - No se modifica a si mismo.
 - Programas no reentrantes.
 - Puede modificarse a si mismo.
 - Lo que parece no es...
 - Utilidad.
 - Tarea.
 - Ejemplos de programas reentrantes y no reentrantes.

Diseño – Enfoque en Capas

- Dijkstra.
- ¿Qué pretende?
 - Menor complejidad de diseño.
 - Menor complejidad de implementación.
- ¿Cómo planteamos o diseñamos un SO en capas?
 - Funcionalidades.
 - Balance.
 - Interfaz de comunicación.

Diseño – Enfoque en Capas

- ¿A qué se parece este enfoque?
 - OSI.
- ¿Será el resultado un SO modular?
 - Separación y sus ventajas.
- ¿Qué ofrece este enfoque?
 - Simplicidad en:
 - Diseño.
 - Especificaciones.
 - Implementación.

Diseño – Enfoque en Capas

- Consideraciones que se deben tener en cuenta para las capas.
 - Anécdota.
- Ejemplo:
 - THE
- Tarea
 - Leer el artículo de THE para discutirlo en la próxima clase.

Diseño – Enfoque en Capas

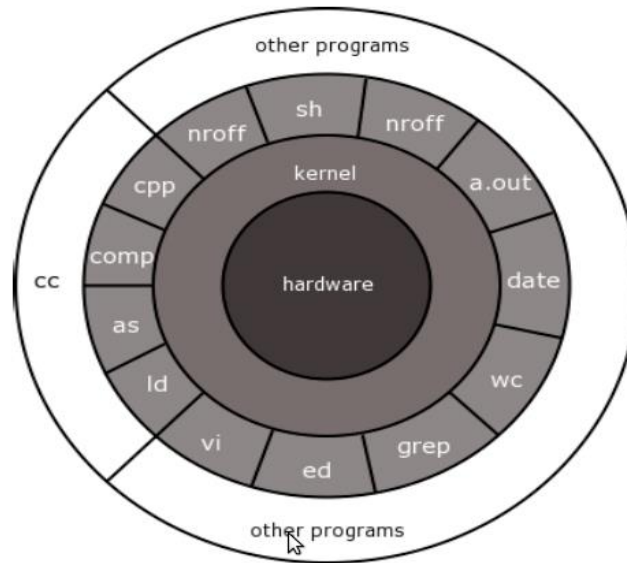
Nivel

5	Operador (Consola con Teclado)
4	Programas de Usuario
3	Flujo de E/S (Buffering)
2	Manejador de Mensajes – (Pase de Mensajes)
1	Manejador de Memoria
0	Asignación de CPU – Sincronización de Procesos

HARDWARE

Diseño – Enfoque en Capas

- Ejemplo:
 - Sistema MULTICS.
 - Capas concéntricas.



Refinando Conceptos

- Protección.
 - Visión interna.
 - El sistema se protege a si mismo.
- Seguridad.
 - Visión Externa.
 - El sistema se protege de extraños.

Diseño – Enfoque Núcleo

- Brinch Hansen.
- Núcleo.
 - Definición.
 - Colección de primitivas.
 - ¿Qué provee?
 - Ambiente.
 - Flexibilidad
 - ¿Por qué?
 - Efecto cebolla.

Diseño – Enfoque Núcleo

- Según Brinch Hansen.
 - Sólo visión de procesos.
 - Creación y comunicación.
 - Sin concepto de recurso.
 - Núcleo grande.
 - Poca flexibilidad a alto nivel.
 - Núcleo pequeño.
 - Bajo soporte a alto nivel.

Diseño – Enfoque Núcleo

- Ejemplos:
 - HYDRA.
 - C.mmp (Carnegie Mellon).
 - Nociones.
 - Recursos.
 - Procesos.

Diseño – Máquinas Virtuales

- Definición.
 - Capa de software que emula la estructura del hardware.
- ¿Para qué?
 - Acceso único.
- ¿Qué se puede correr sobre este enfoque?
 - Monousuario.
 - Multiprogramación.

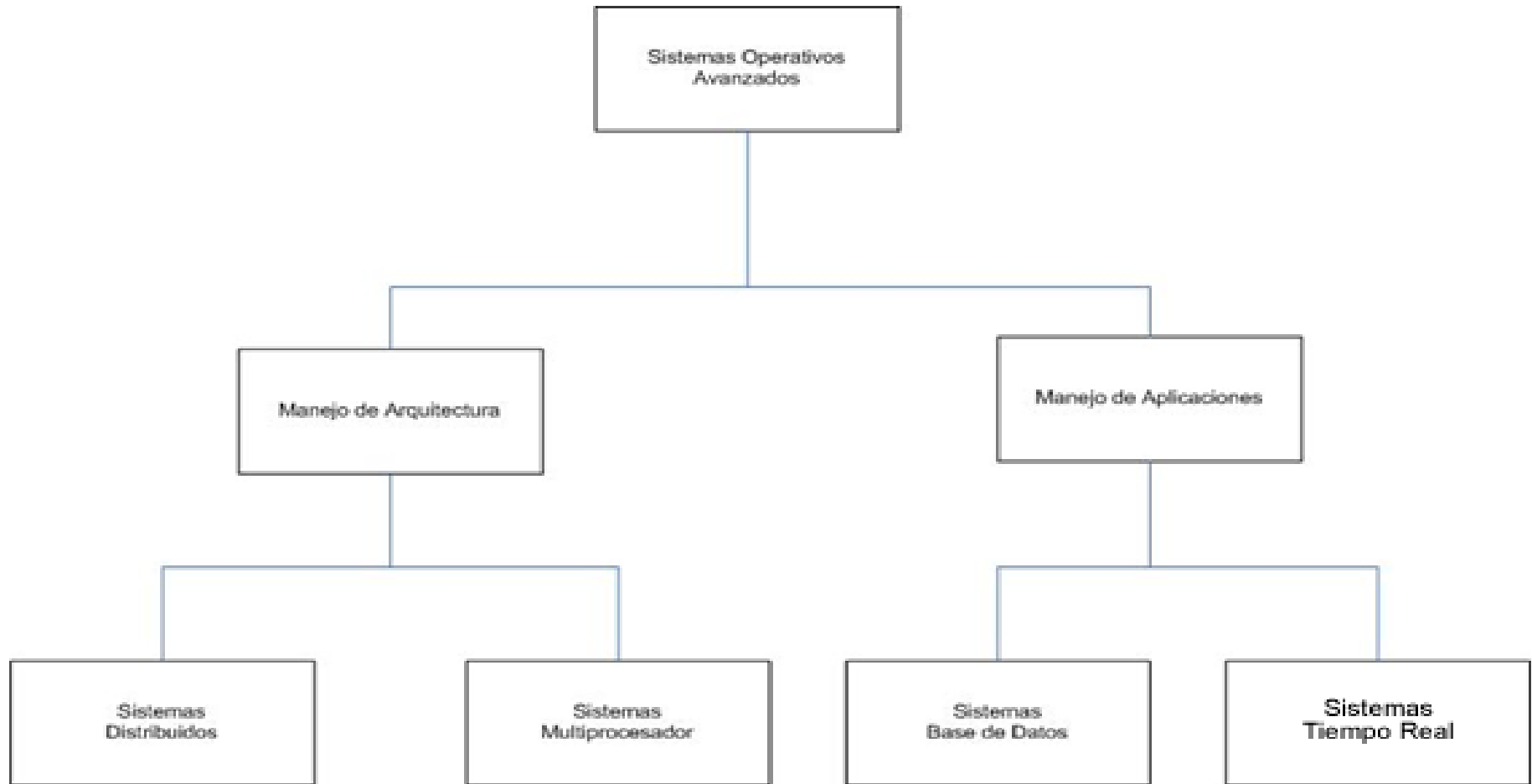
Diseño – Máquinas Virtuales

- Flexibilidad.
 - ¿De que punto de vista?
- Ventajas.
 - Test-Bed.
 - No interfiere con el resto de los usuarios.
- Desventajas.
 - Complejidad.
 - Desempeño.
- Ejemplos.

¿Por qué avanzan los Sistemas Operativos?

- Esfuerzos -- ¿Hacia donde?
 - Surgen nuevas arquitecturas.
 - Ley de Moore.
- ¿Qué paso?
 - Nos quedamos cortos.
- Nacen los Sistemas Operativos Modernos.
 - ¿Qué es un Sistema Operativo Moderno?

Sistemas Operativos Avanzados



SO Distribuidos

- Definición.
 - SO.
 - Conjunto de máquinas.
 - Autónomas.
- Función principal.
 - Transparencia.

SO Distribuidos

- Diseño.
 - Sincronización.
 - Abrazo mortal.
 - Planificación.
 - Sistema de archivos.
 - Memoria.
 - Comunicación.
 - Nombramiento.
 - Manejo de reloj.
 - Retardo en las comunicaciones.

SO Multiprocesadores

- Definición.
 - SO.
 - Varios procesadores.
- Diferencia con un SOD.

SO Multiprocesadores

- Diseño.
 - Consideraciones básicas.
 - Sincronización de procesos.
 - Planificación de tareas.
 - Manejo de memoria.
 - Protección y seguridad.

SO Base de Datos

- Definición.
 - SO.
 - Enfoque.
- ¿Qué debe incluir?
 - Transacciones.
 - Recuperación.
 - Manejo eficiente de buffer.

SO Tiempo Real

- Definición.
 - SO.
 - Enfoque.
- ¿Qué debe incluir?
 - Manejo paranoico del tiempo.
 - Tiempo real suave.
 - Tiempo real duro.