Sistemas Operativos I

Introducción

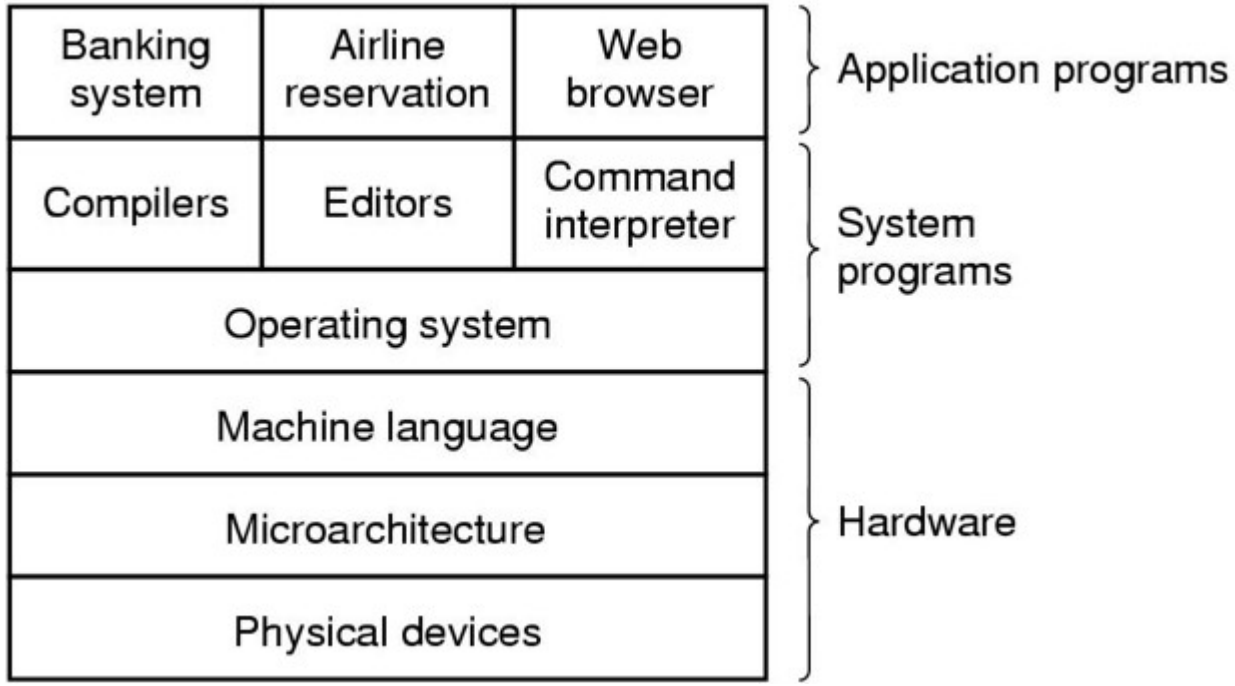
- Qué es un SO?
- Historia de los SO
- Revisión del hardware
- Conceptos de SO
- Llamadas al sistema
- Estructura de los SO

¿Qué es un Sistema Operativo?

Sistema Operativo

- Es una máquina extendida
  - Oculta los detalles tediosos y complejos que se deben realizar
  - Presenta al usuario una “máquina virtual” que es mucho más simple de usar
- Es un administrador de recursos
  - Cada programa obtiene tiempo de uso de recursos
  - Cada programa obtiene espacio de utilización de un recurso-

Componentes de un Sistema

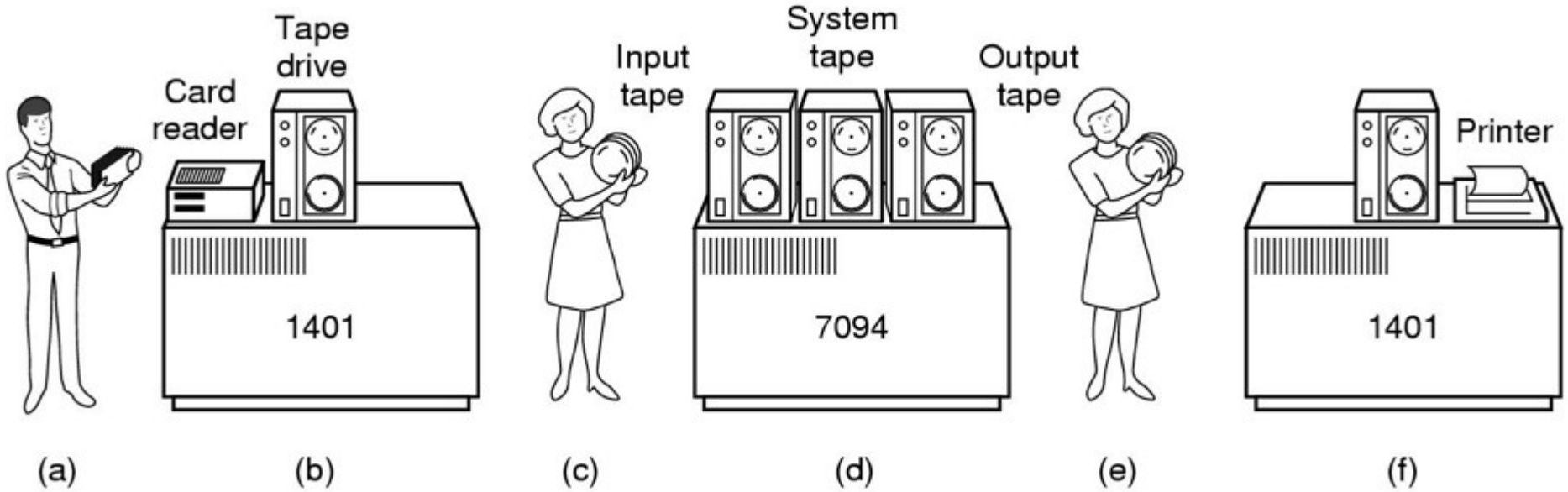


Historia

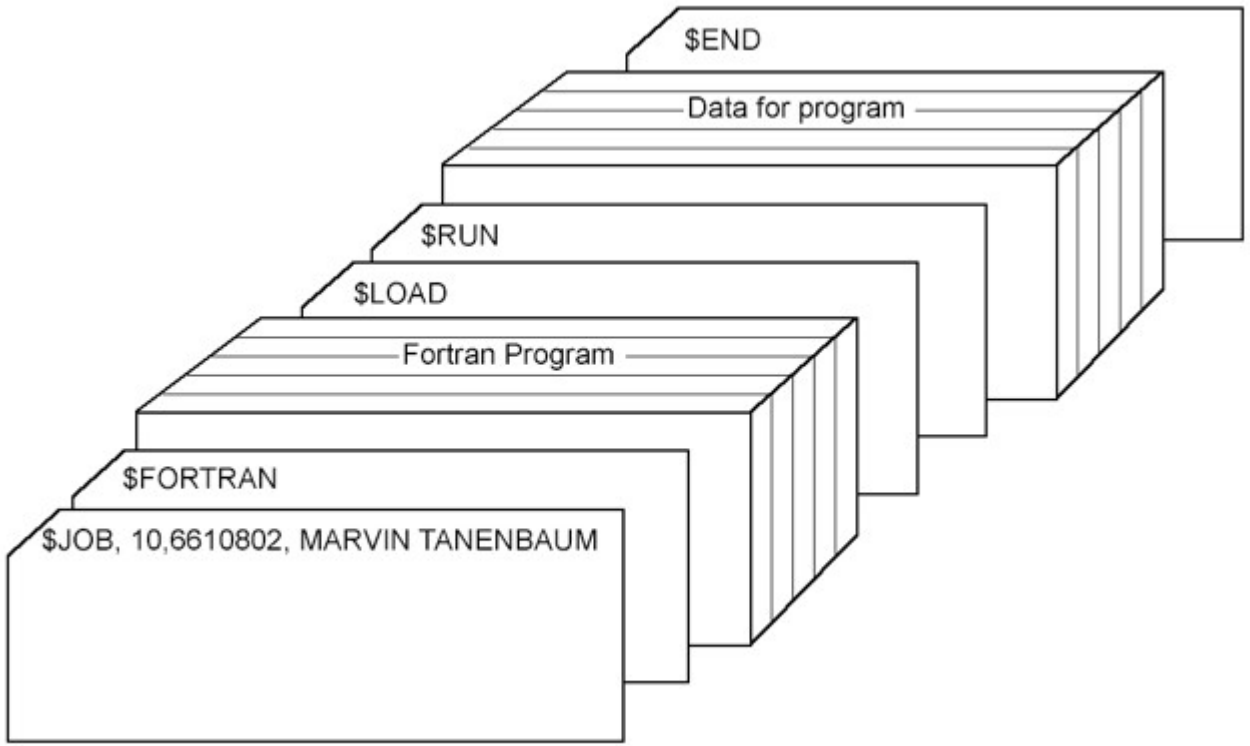
Generaciones

- Primera generación 1945 - 1955: válvulas, tableros
- Segunda generación 1955 - 1965: transistores, sistemas batch
- Tercera generación 1965 – 1980: circuitos integrados, multiprogramación
- Cuarta generación 1980 – presente: microprocesadores, computadoras personales
- Quinta generación 1990 – presente: computadoras de bolsillo o móviles

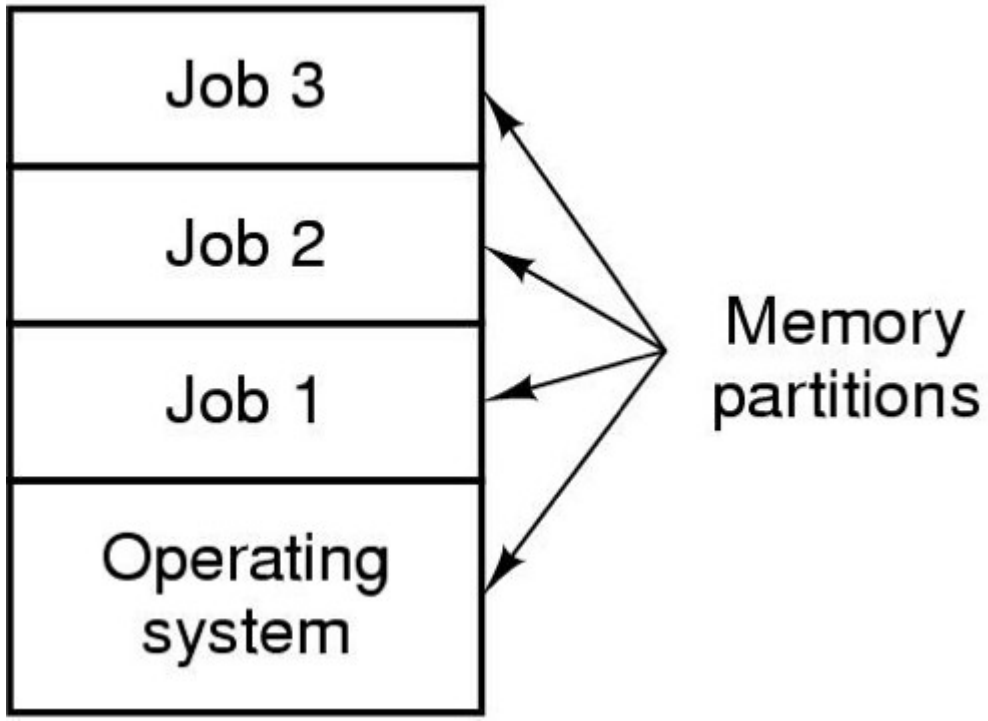
Sistemas Batch



Sistemas Batch

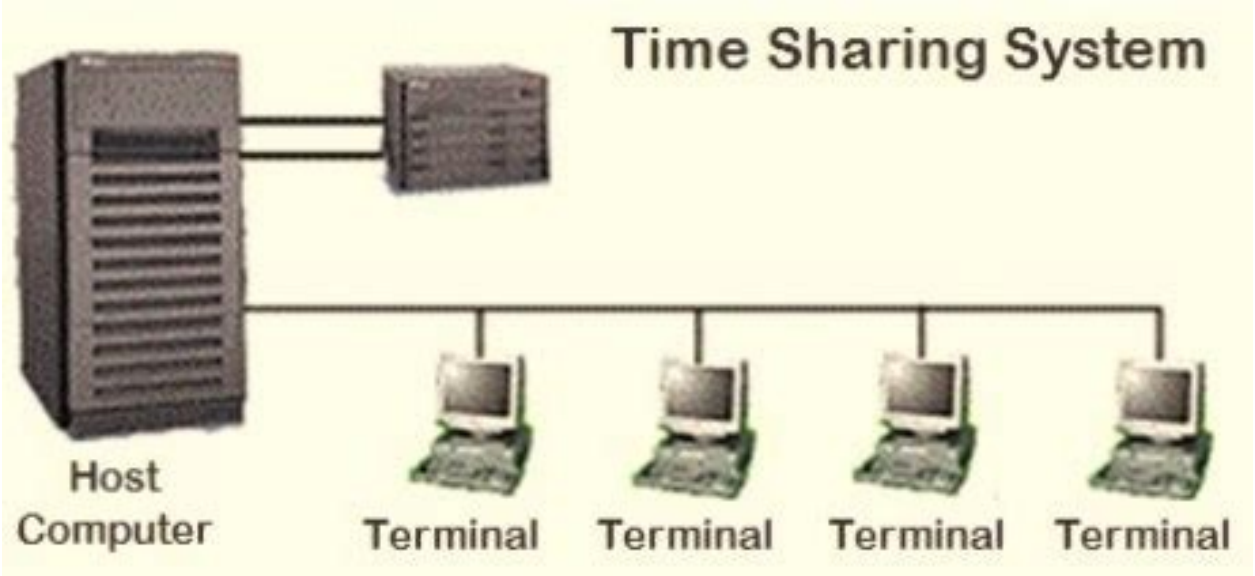


Multiprogramación

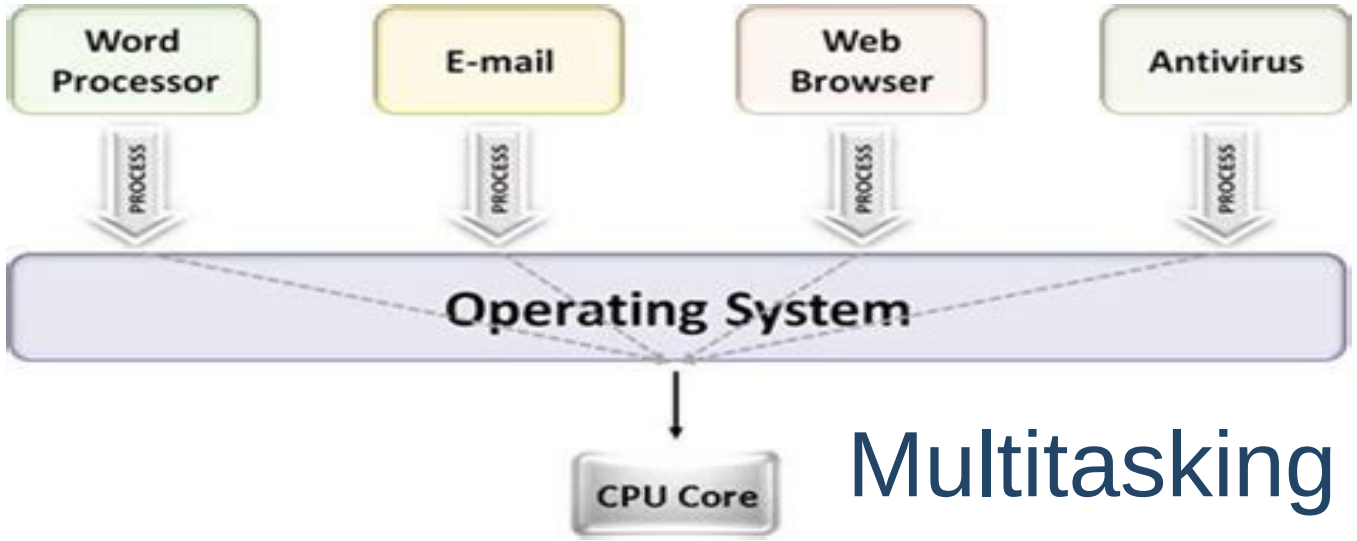


- INCREMENTA USO de CPU
- JOB POOL en disco
- CPU NO IDLE
- NO INTERACCION con el USUARIO

Time Sharing - Multitasking



- INTERACCION con el USUARIO - input device
- ACCIONES CORTAS
- POCO TIEMPO de CPU.
- sensación SISTEMA DEDICADO
- asegura tiempo de respuesta
- require PROGRAMAS cargados EN MEMORIA
- ORDEN directa al SO
- **process:** programa en memoria en ejecución



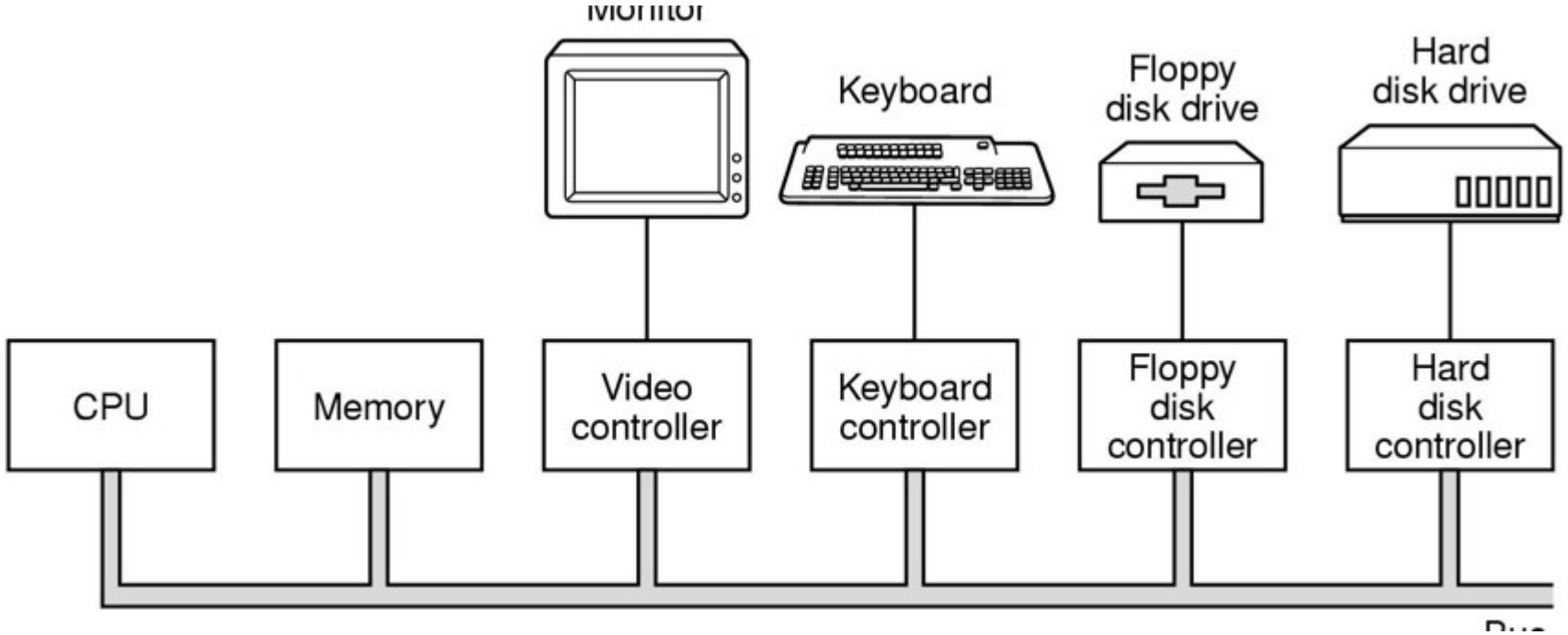
Multitasking

Diversidad de SO

- Mainframes
- Servidores
- Multiprocesador
- Computadora Personal
- Tiempo Real
- Embebidos

Revisión del Hardware

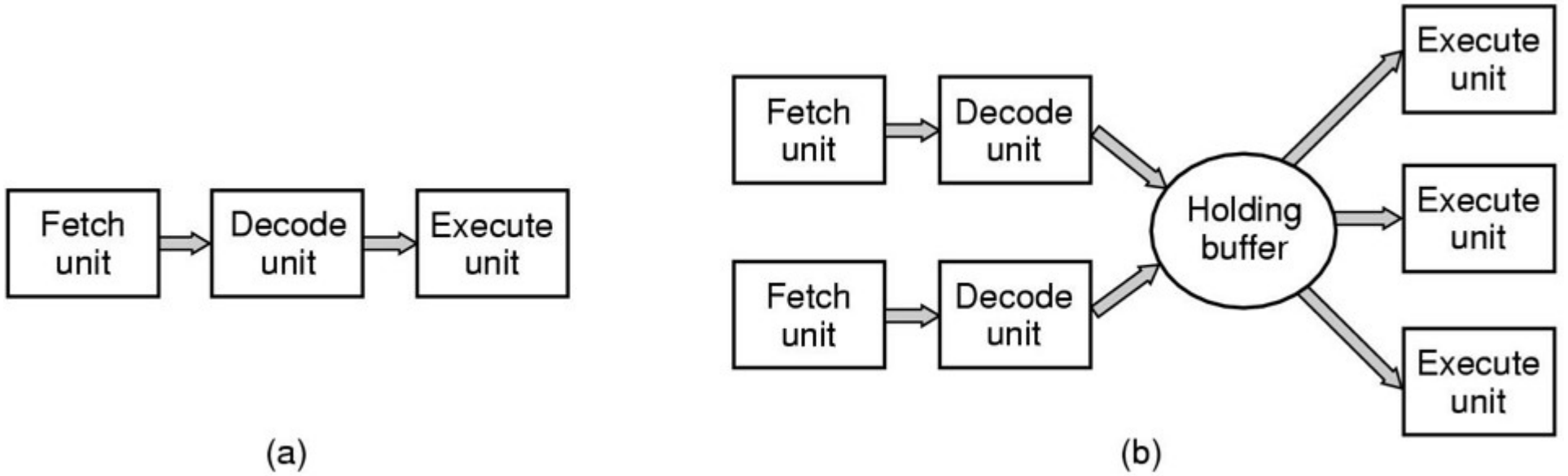
Componentes de una Computadora



CPU

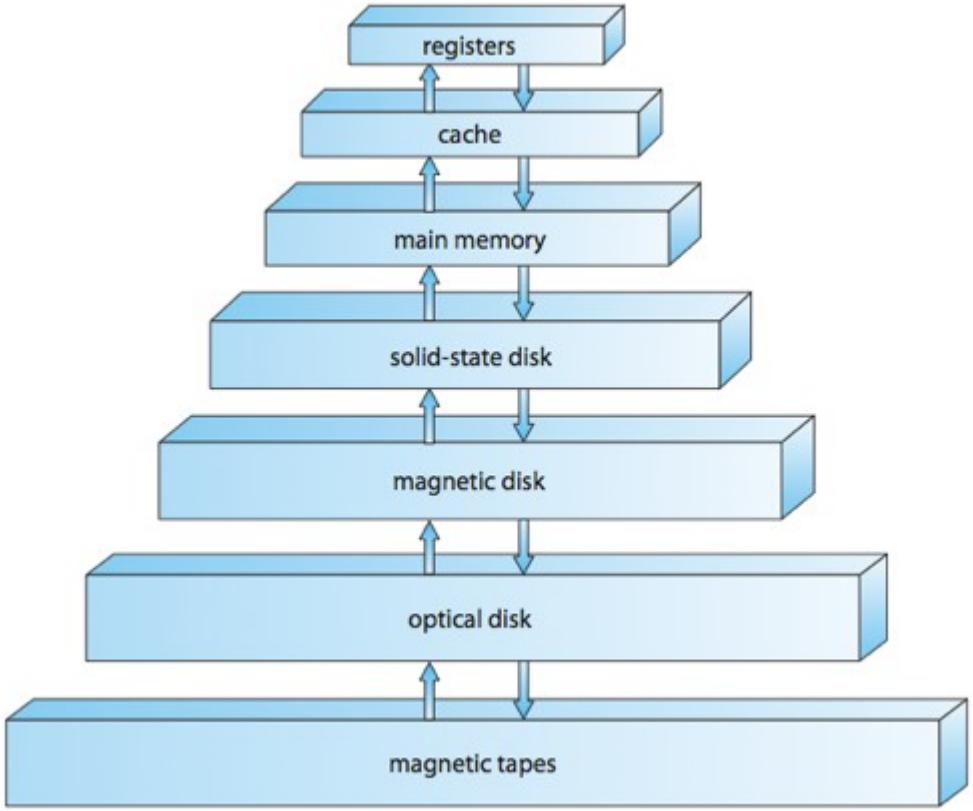
- Tomar instrucción de memoria, decodificarla y ejecutar
- Conjunto de instrucciones específico
- Registros de propósito general, contador de programa, puntero a la pila, PSW

Tipos de CPU



- (a) Pipeline 3 etapas
- (b) Superescalar

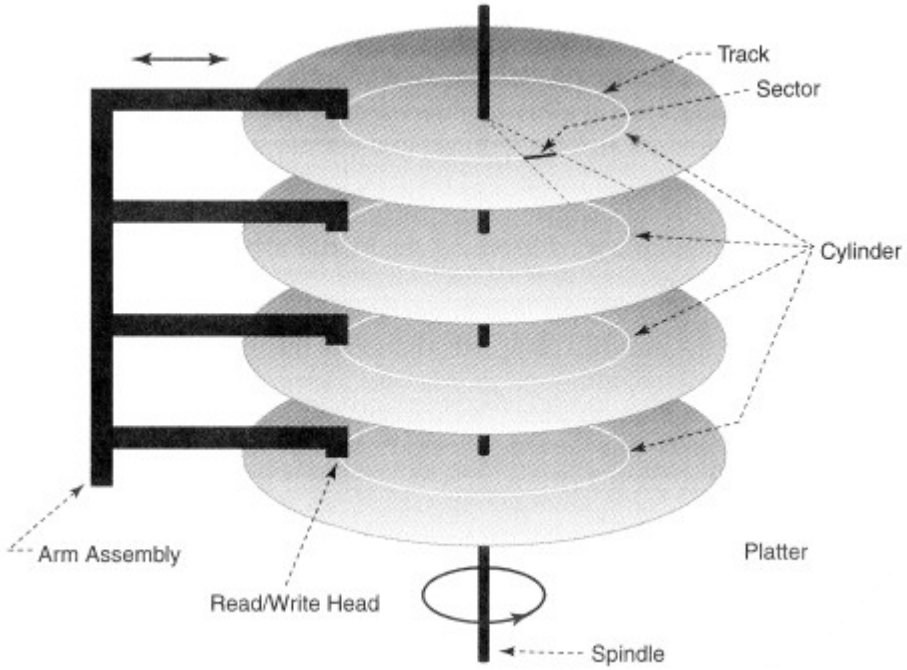
Almacenamiento



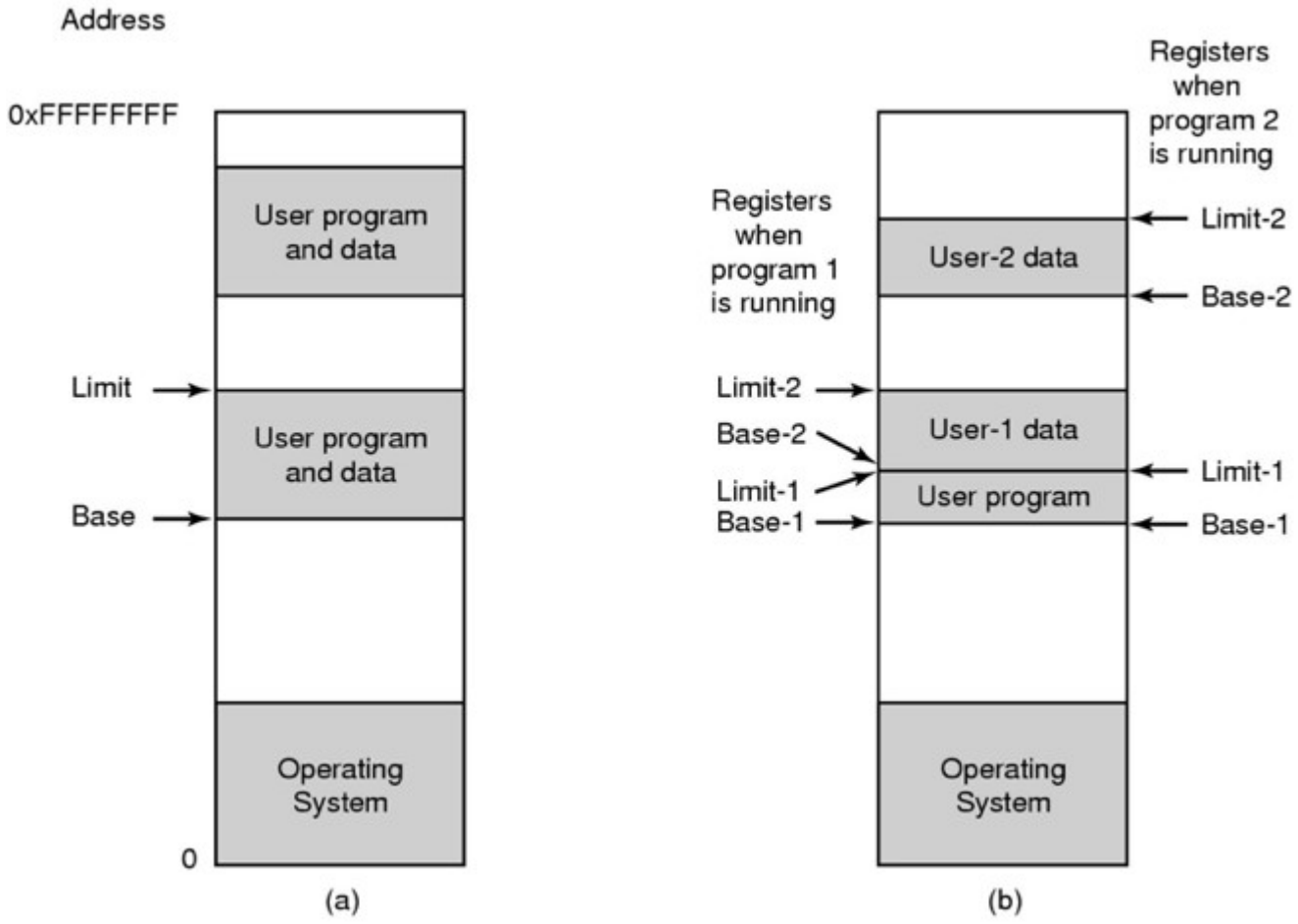
Almacenamiento

Nivel	1	2	3	4
Tipo	reg	cache	mem ppal	disco
Tamaño	< 1Kb	> 16Mb	16Gb	128Gb
Tecnología	CMOS	CMOS/SRAM	CMOS/DRAM	SSD
T Acceso (ns)	0.25	0.5	100	35000
Bandwith (Mb/s)	100000	10000	5000	250
Manejado	compilador	hardware	SO	SO

Discos Magnéticos



Memoria Principal

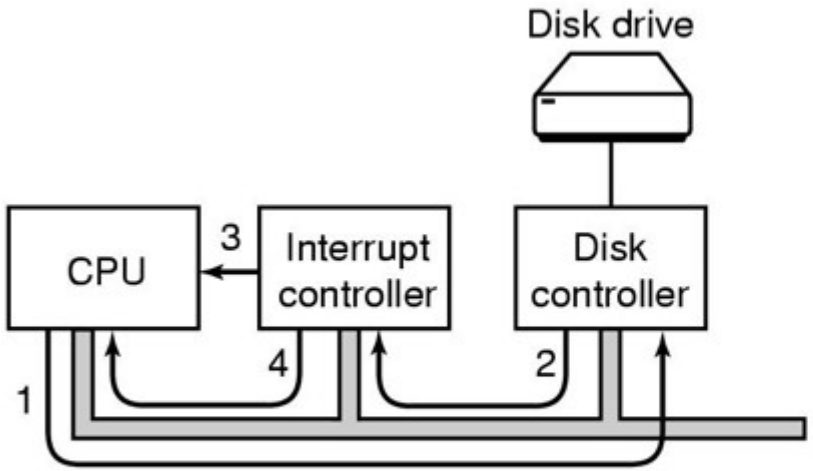


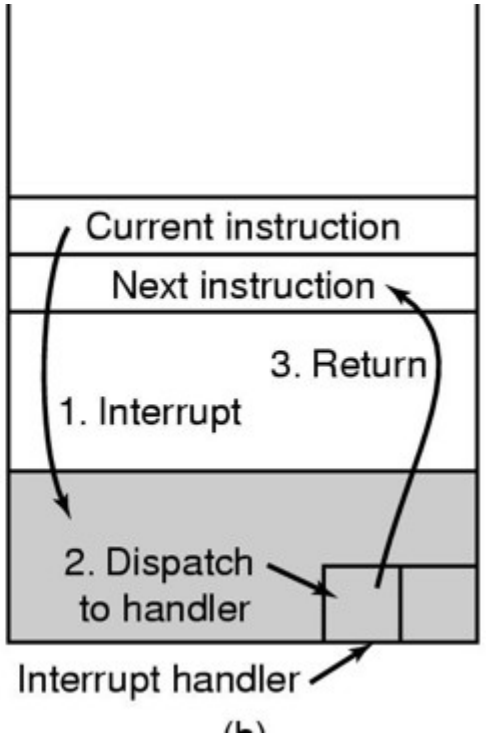
Entrada/Salida

Solicitudes de E/S

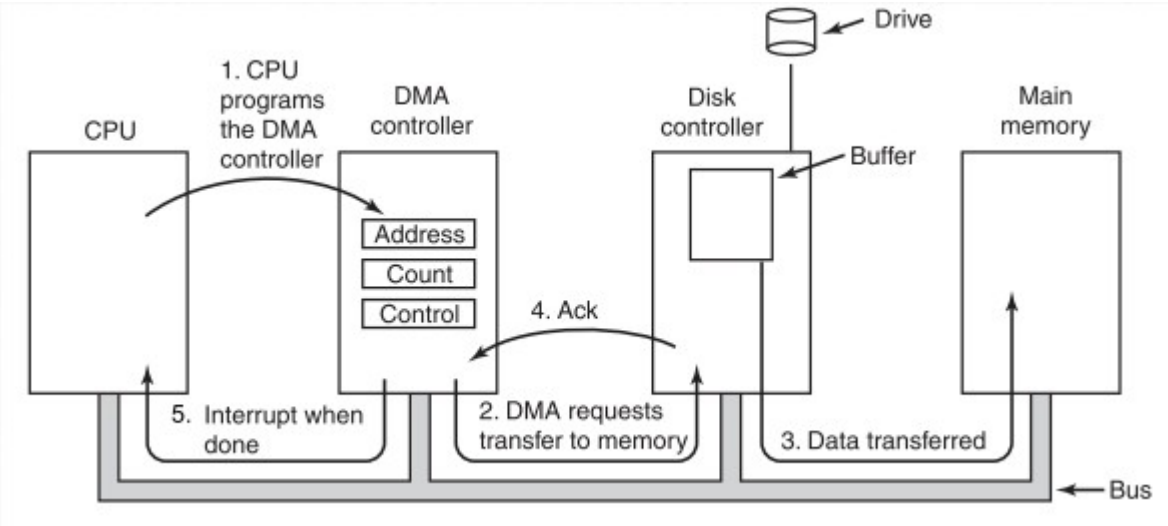
- Busy waiting
- Interrupciones
- DMA

Interrupciones

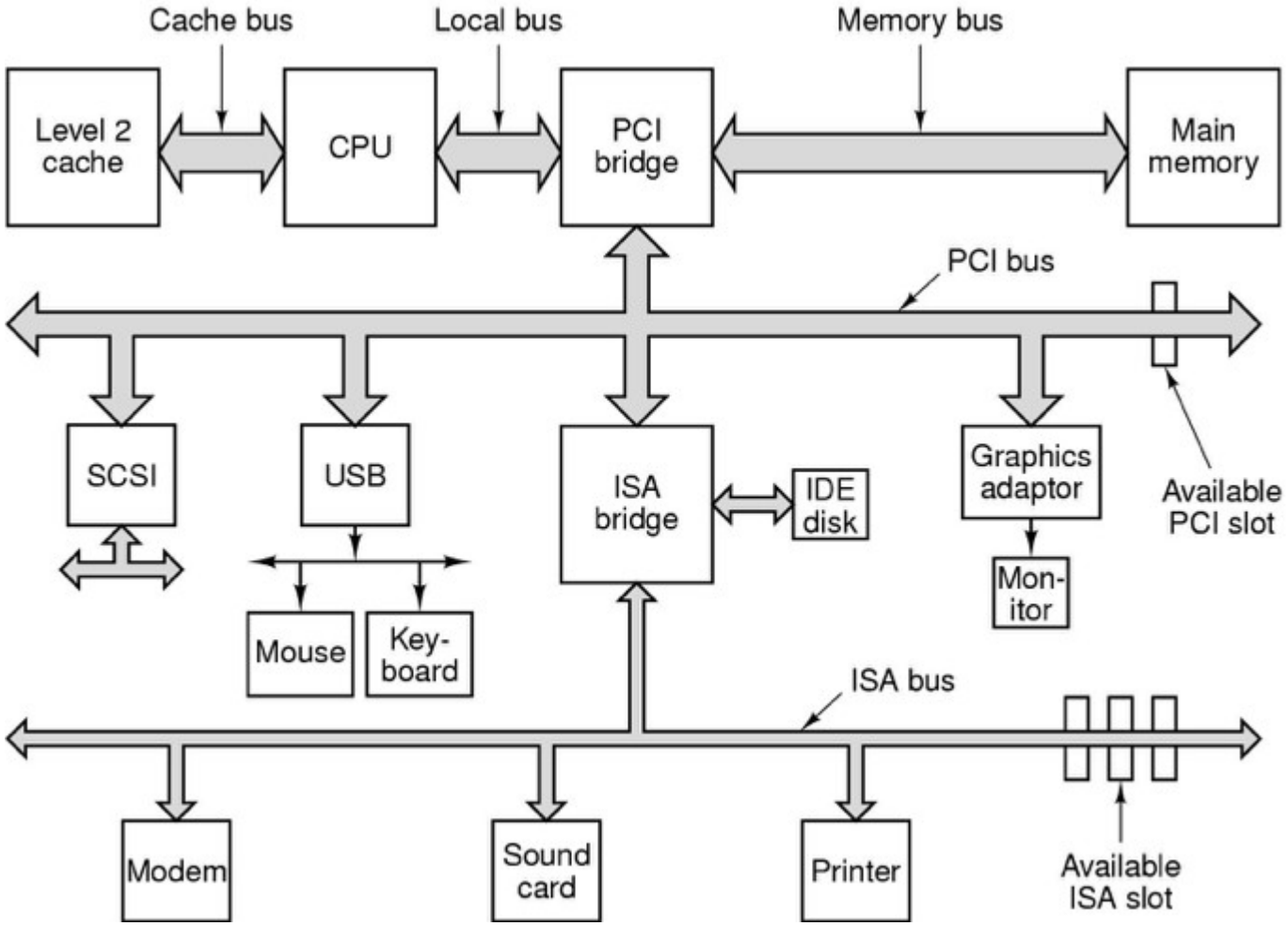




DMA

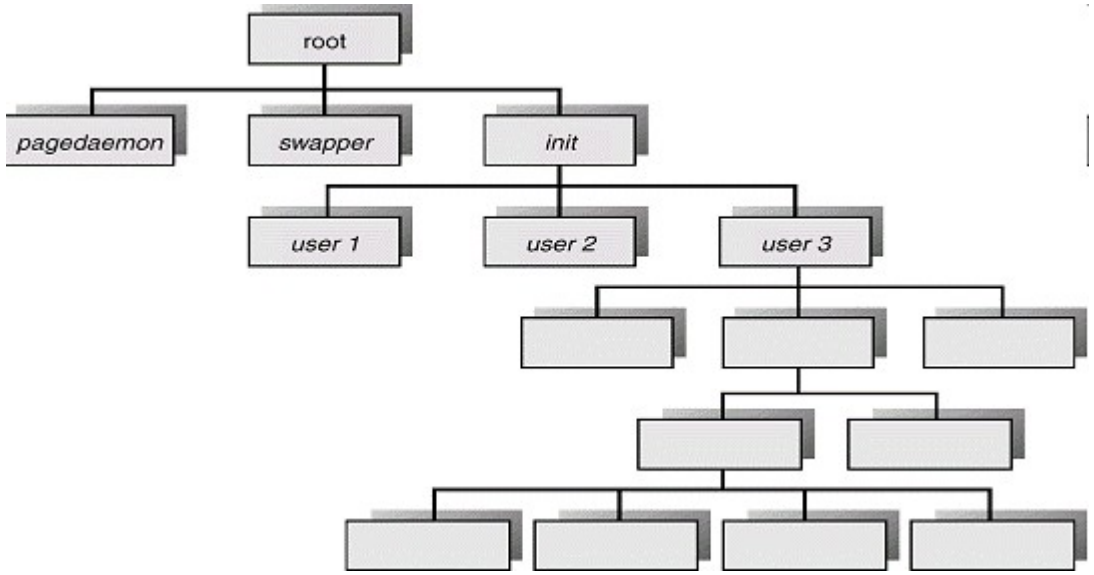

Drive
1. CPU programs the DMA controller
DMA controller
Disk controller
Main memory
CPU
Buffer
Address
Count
Control
4. Ack
5. Interrupt when done
2. DMA requests transfer to memory
3. Data transferred
Bus

Complejidad del Hardware



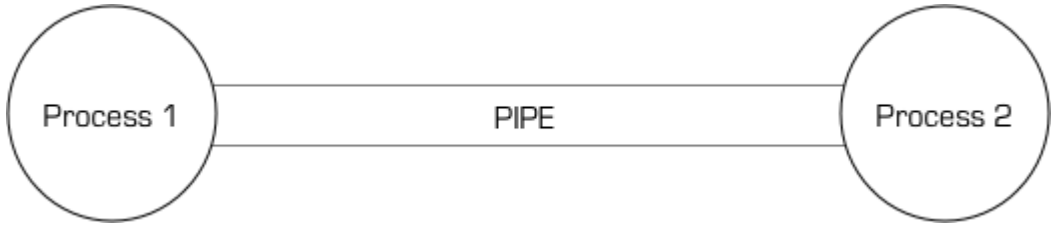
Conceptos

Procesos

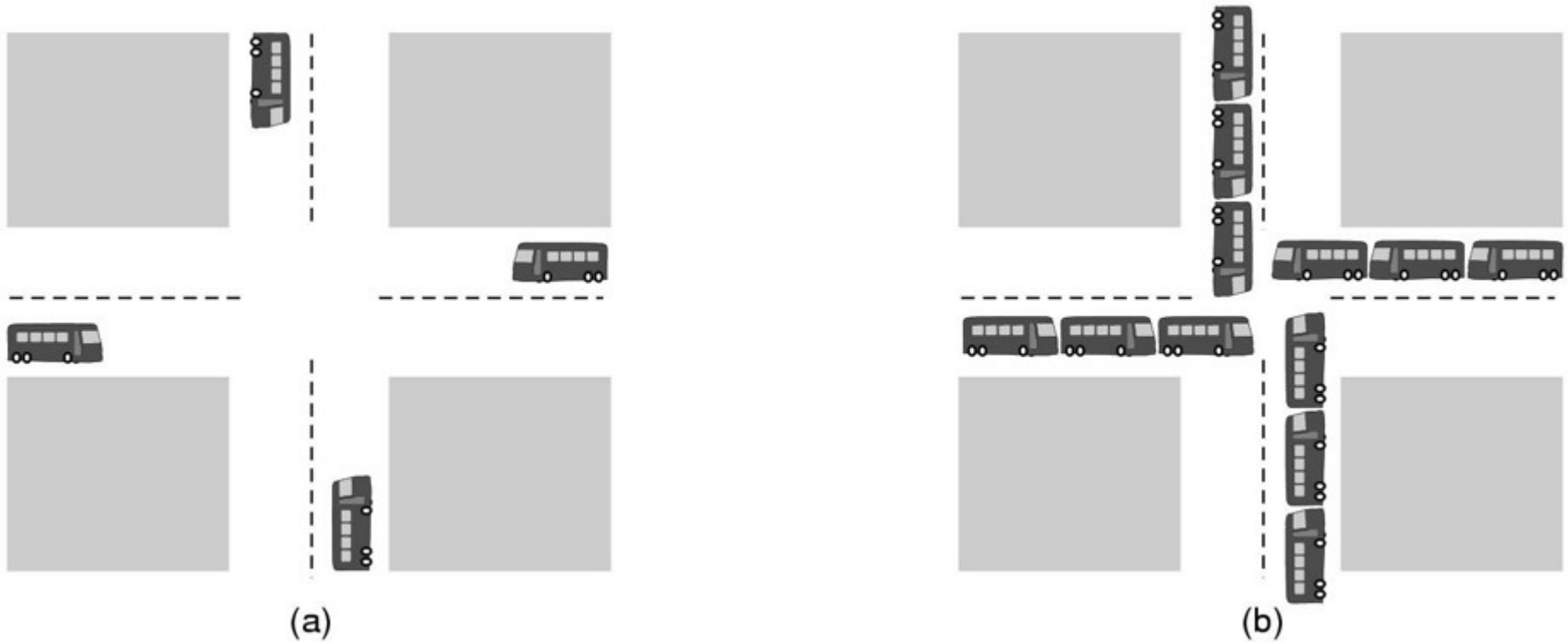


> Arbol de procesos

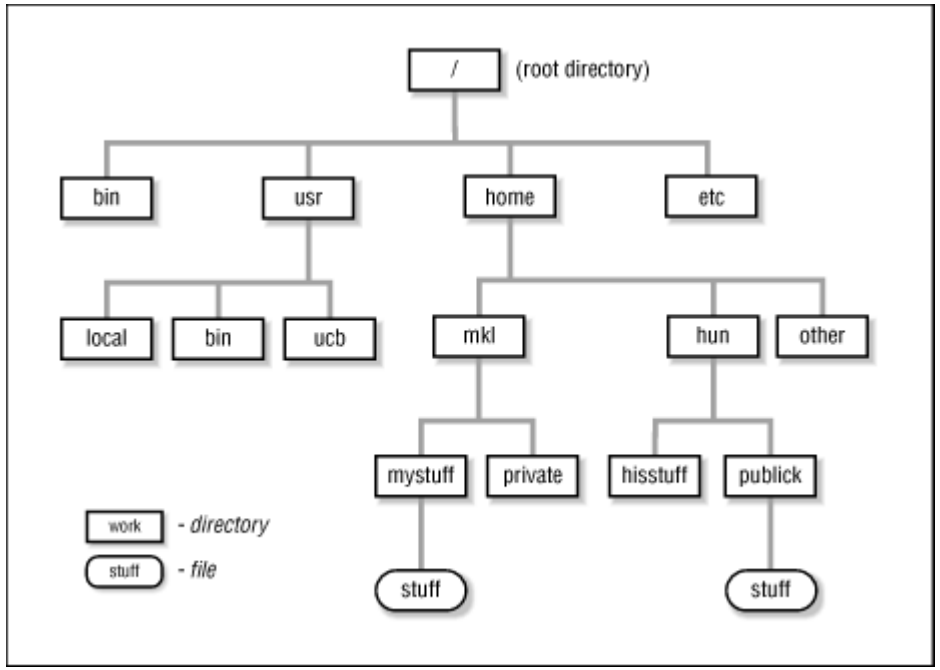
Comunicación Entre Procesos



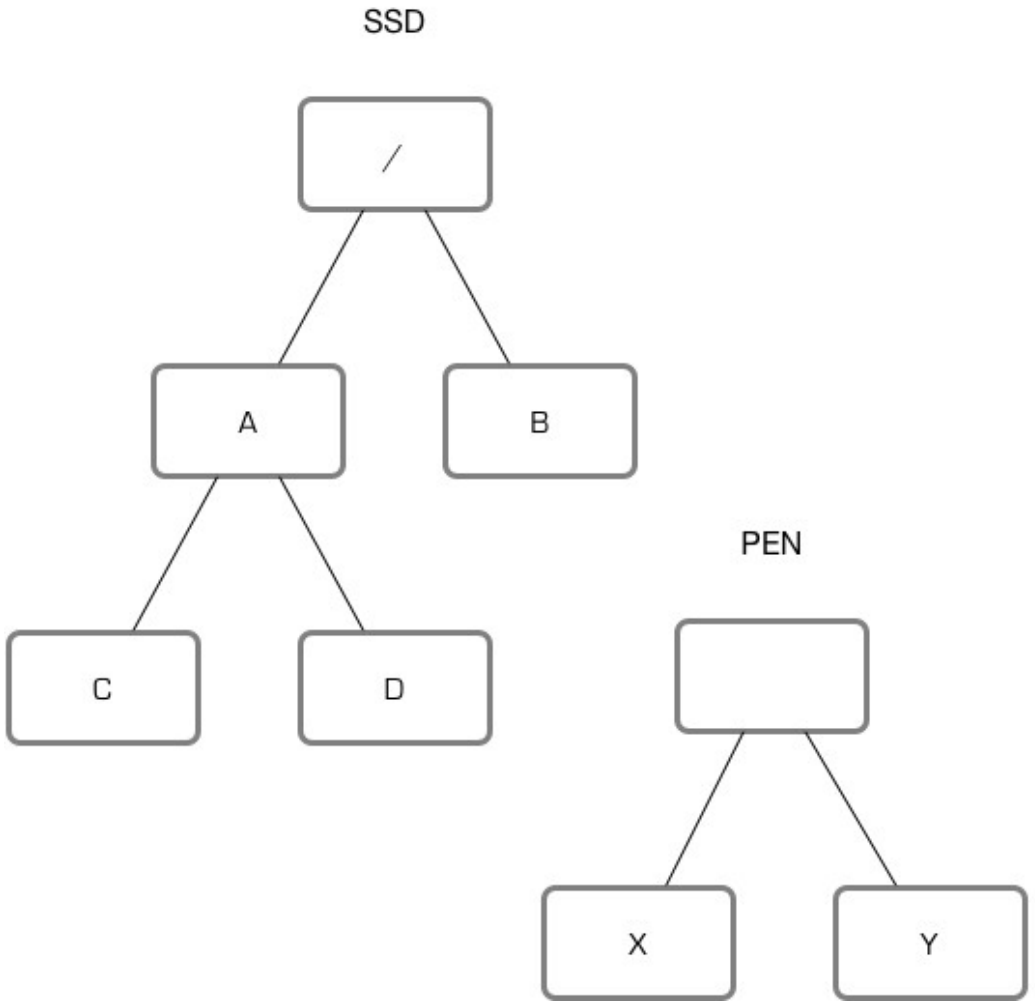
Deadlocks

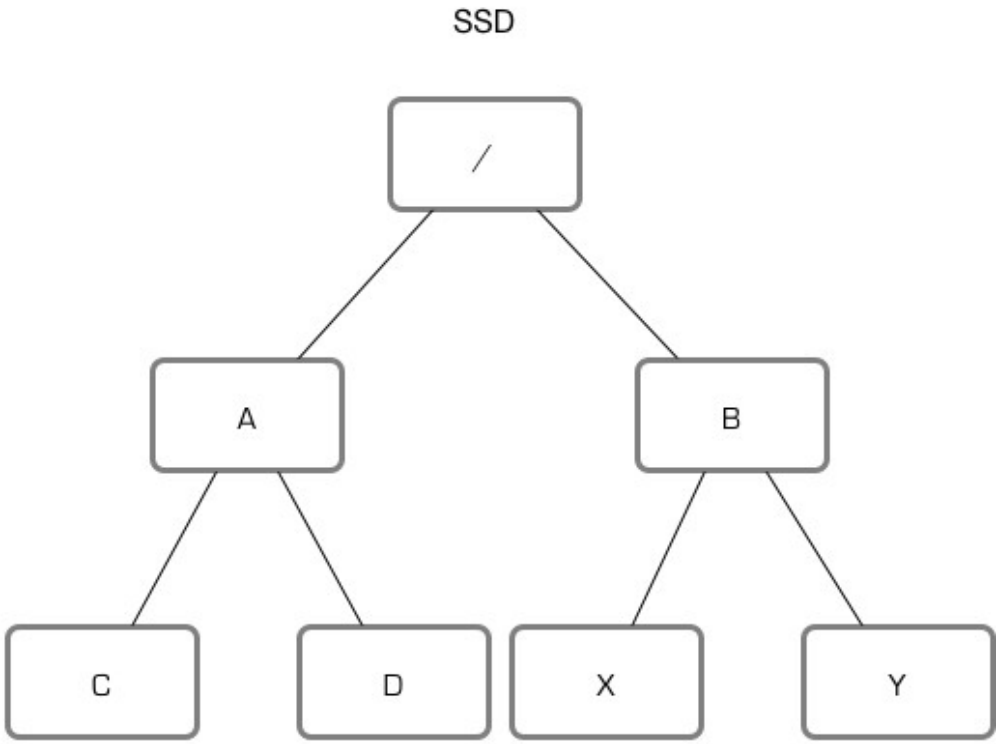


Sistemas de Archivos



Volumenes y Puntos de Montado



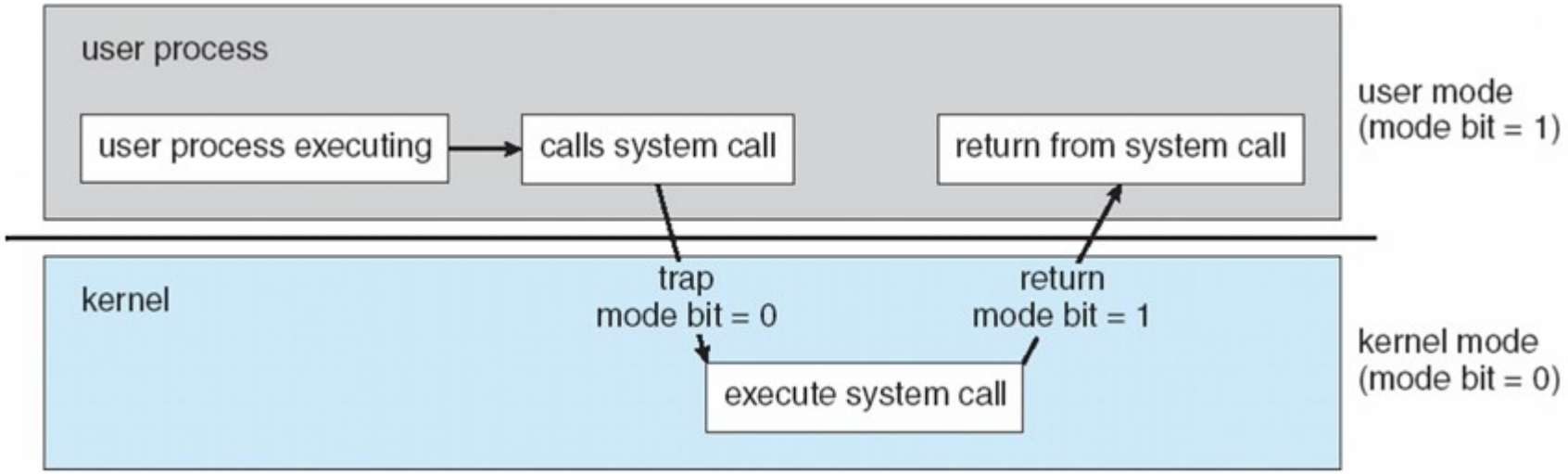


Modo Protegido

> El modo dual de ejecución permite al SO protejerse así mismo y otros componentes del sistema

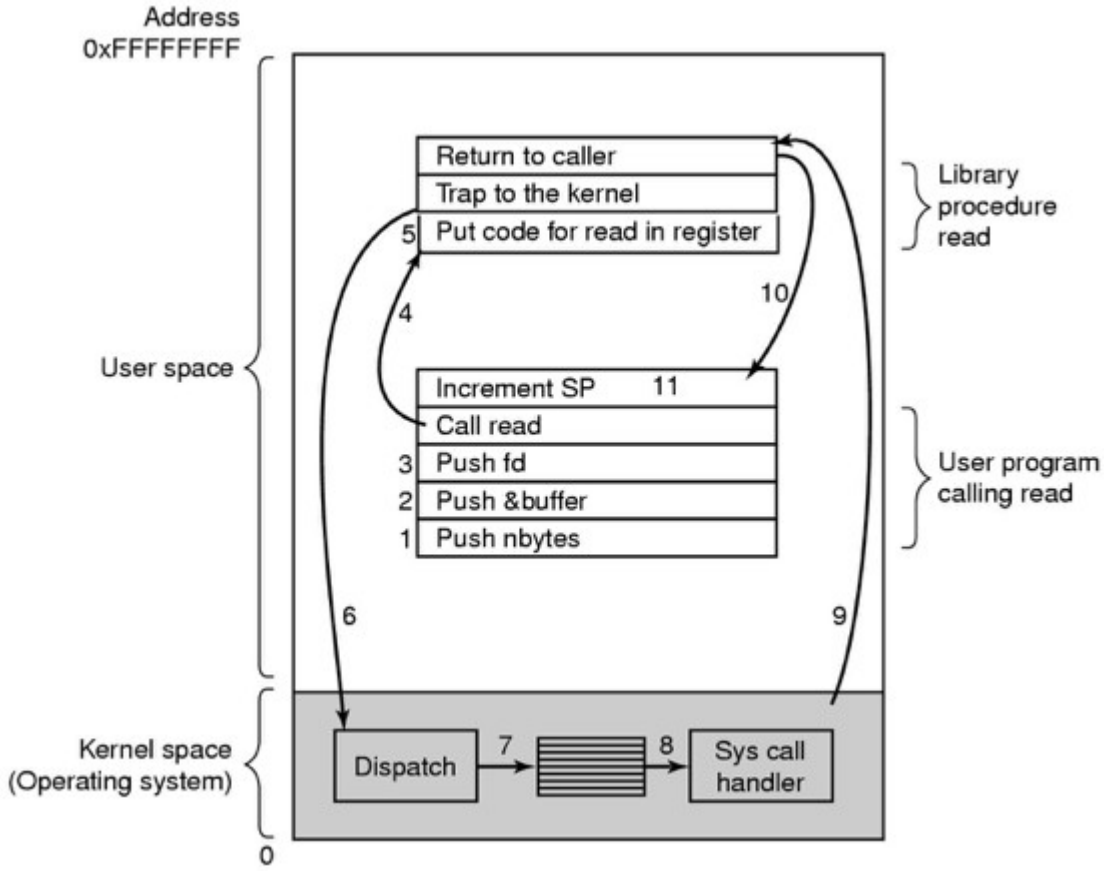
- 2 modos: usuario y supervisor (kernel)
- El bit de modo permite:
  - Distinguir cuando el sistema está ejecutando código del sistema o usuario
  - Algunas instrucciones designadas como privilegiadas, solo ejecutar en modo kernel
  - Las llamadas al sistema cambia a modo kernel, cuando retornan vuelven al modo usuario

Ejecución en Modo Protegido



Un ERROR en un programa SOLO debe afectar a dicho programa

Llamadas al Sistema



Aministración de Procesos

Llamada	Descripción
`pid = fork()`	crea un nuevo proceso idéntico al hijo
`pid = waitpid(pid, ...)`	espera que un proceso termine
`s = execv(name, argv, env)`	reemplaza la imagen del proceso
`exit(status)`	termina la ejecución de un proceso

Manejo de Archivos

Llamada	Descripción
`fd = open(file, ...)`	abre un archivo
`s = close(fd, ...)`	cierra un archivo abierto
`n = read(fd, buffer, nbytes)`	lee de un archivo a un buffer
`n = write(fd, buffer, nbytes)`	escribe de un buffer a un archivo
`pos = lseek(fd, offset, whence)`	mueve el puntero
`s = stat(fd, &buf)`	retorna información de estado de un archivo

Manejo de Archivos

Llamada	Descripción
`s = mkdir(name, mode)`	crea un nuevo directorio
`s = rmdir(name)`	borra un directorio vacío
`s = link(name1, name2)`	crea una nueva entrada name2 apuntando a name1
`s = unlink(name)`	remueve una entrada
`s = mount(special, name, flag)`	monta un sistema de archivos
`s = umount(special)`	desmomta un sistema de archivos

Un Shell Simplificado

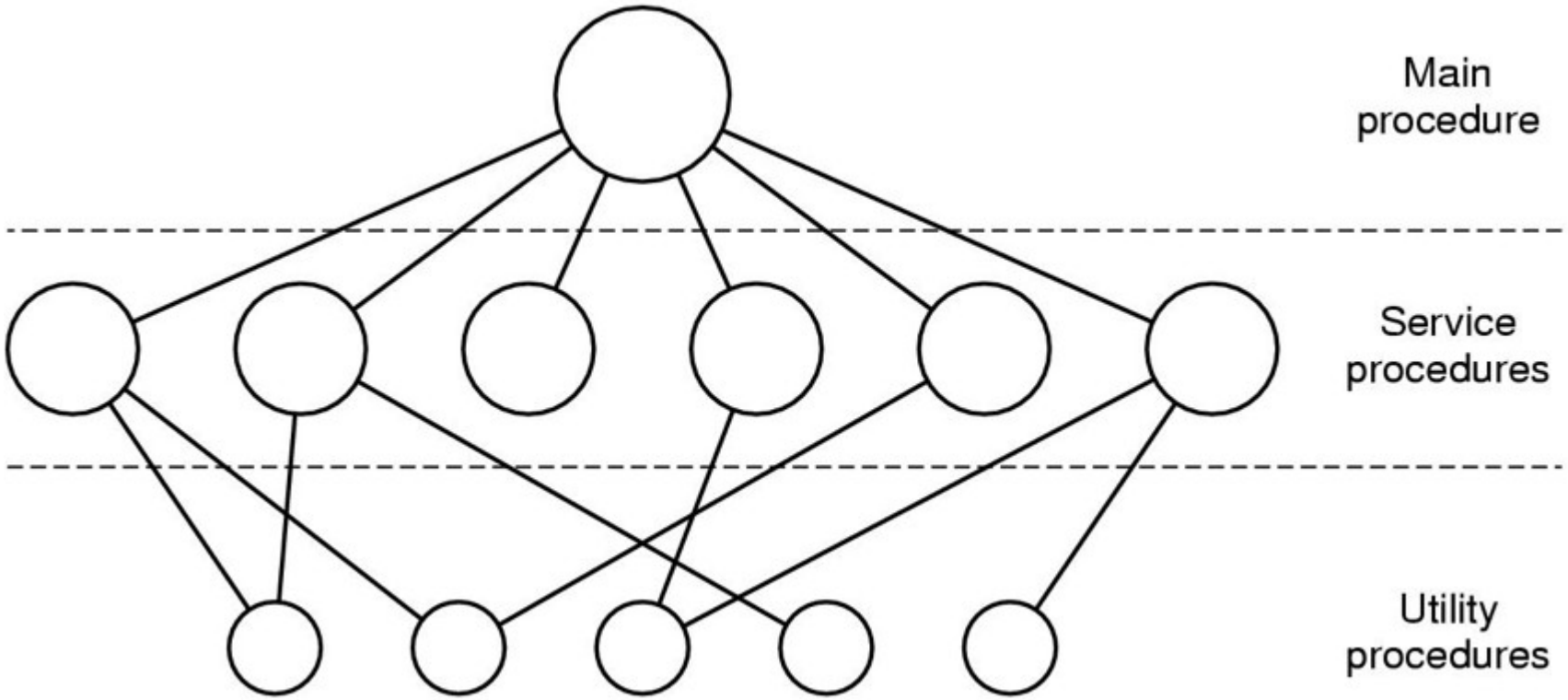
```
    while (TRUE) {                                          /* repite por siempre */
       type_prompt( );                                   /* mostrar el prompt */
       read_command(command, parameters)              /* ingreso desde la terminal */
       if (fork() != 0) {                                      /* fork de un proceso */
          /* Parent code */
          waitpid( -1, &status, 0);                     /* esperar finalización del hijo */
       } else {
         /* Child code */
         execve(command, parameters, 0);              /* ejecutar comando */
      }
   }
```

Llamadas al Sistema WIN32

UNIX	WIN32	Descripción
`fork()`	`CreateProcess`	crea un nuevo proceso idéntico al hijo
`waitpid`	`WaitForsingleObject`	espera que un proceso termine
`execv`	-	reemplaza la imagen del proceso
`exit`	`ExitProcess`	termina la ejecución de un proceso
`open`	`CreateFile`	abre un archivo
`close`	`CloseHandle`	cierra un archivo abierto
`read`	`ReadFile`	lee datos de un archivo
`i`	`i il`	ib d t hi

Estructura de los Sistemas Operativos

Sistema Monolítico



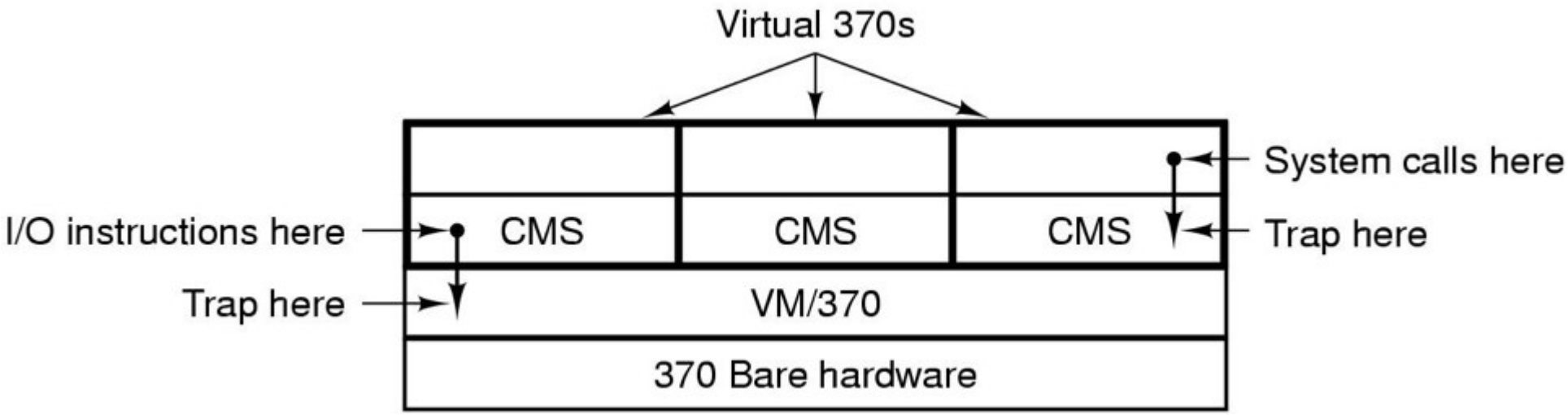
- Los programas de aplicación invocan los servicios del sistema
- Tienen un conjunto de servicios del sistema implementan las llamadas al sistema
- Un conjunto de procedimientos de utilidad que ayudan a los servicios del sistema

Estructura en Capas

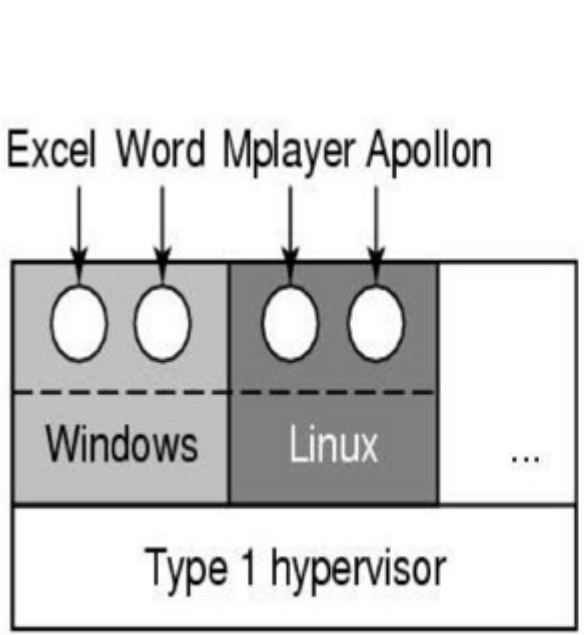
Layer	Función
5	El operador
4	Programas del usuario
3	Manejo de entrada/salida
2	Comunicación entre procesos
1	Manejo de memoria
0	Asignación de procesador y multiprogramación

- El sistema operativo está dividido en un número de capas (niveles), cada una construido sobre las capas - inferiores
- La capa inferior (capa 0) es el hardware; la capa más alta (capa N) es la interfaz del usuario
- En un sistema modular, las capas son elegidas de forma tal que cada capa usa solo las funciones y servicios de las capas inferiores

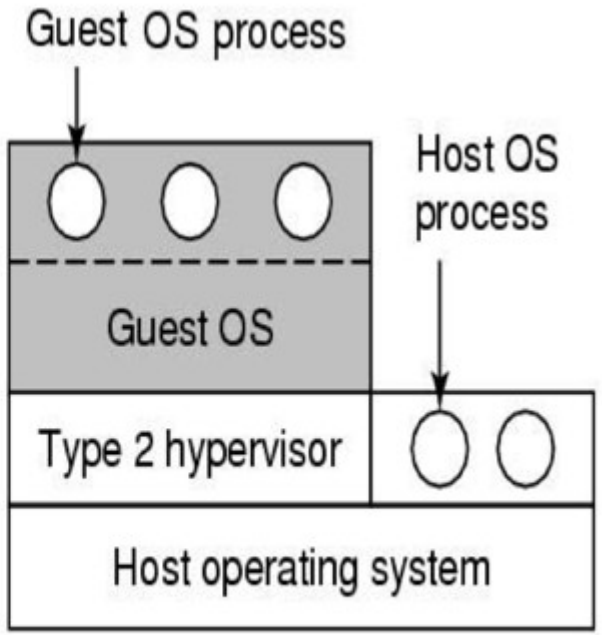
Máquinas Virtuales



- Una máquina virtual tiene una aproximación de un sistema en capas respecto de su siguiente capa lógica. Trata al sistema operativo y al hardware como si fuesen hardware
- Una máquina virtual brinda una interfaz idéntica al hardware subyacente
- El sistema operativo host crea la ilusión que cada proceso tiene su propio procesador y memoria
- A cada huésped se le brinda una copia virtual de la computadora subyacente

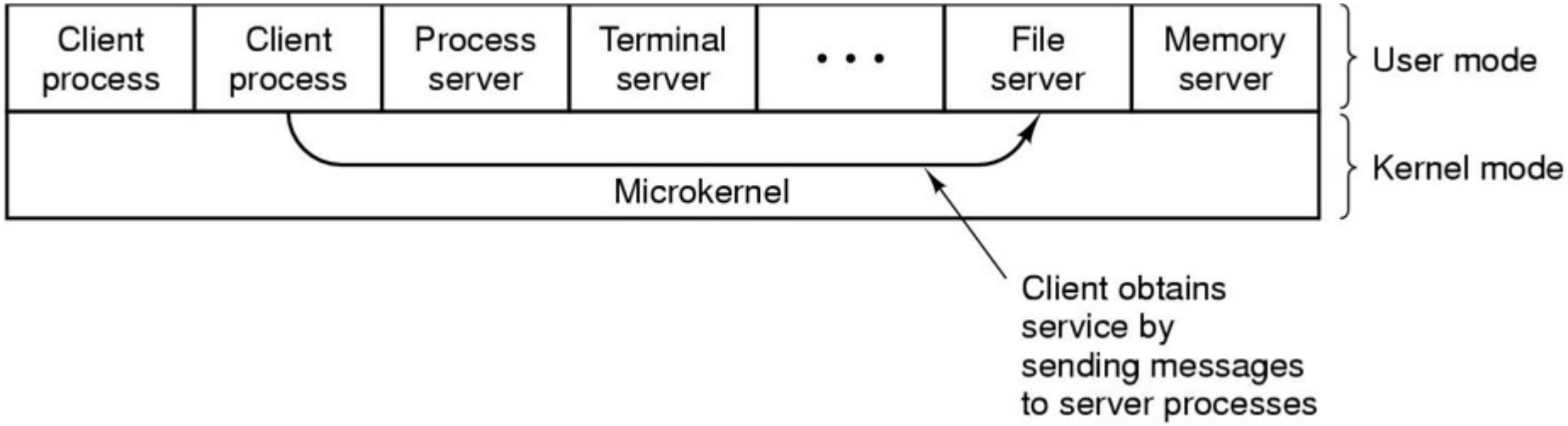

Excel
Word
Mplayer
Apollon
Windows
Linux
...
Type 1 hypervisor

(a)


Guest OS process
Host OS process
Guest OS
Type 2 hypervisor
Host operating system

(b)

Microkernel

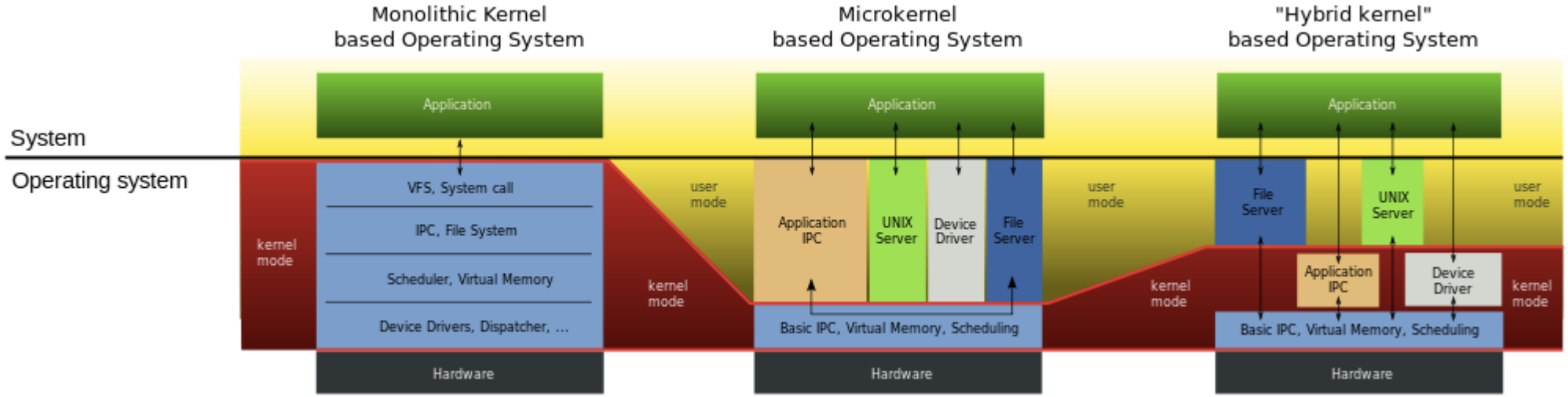


- Mueve la mayor cantidad de funcionalidad posible del kernel al espacio de “usuario”
- Solo algunas funciones esenciales quedan en el kernel: manejo primitivo de memoria (espacio de direcciones), E/S y manejo de interrupciones, Comunicación entre procesos (IPC), planificación básica
- Otros servicios del SO son provistos por procesos que ejecutan en espacio del usuario:
  drivers de dispositivos, sistema de archivos, memoria virtual

> La comunicación entre los módulos del usuario se hace mediante pasaje de mensajes

- Beneficios:
  - más fácil de extender
  - más fácil de portar
  - más sólido
  - más seguro
- Desventajas:
  - Hay un overhead de peformance por la comunicación entre el espacio de usuario y espacio de kernel

Comparación de Estructuras



Arbol Genealógico