Tema I

Consideraciones Generales de los Sistemas Operativos

UNED
Manuel Fernández Barcell
http://www.mfbarcell.es
Blog: http://prof.mfbarcell.es

Fuentes:
- Carolina Mañoso
- Gustavo Romero

Objetivos docentes

- Saber qué es un sistema operativo y cuáles son sus objetivos y servicios.
- Conocer la evolución histórica de los sistemas operativos.
- Conocer los criterios que permiten clasificar a los sistemas operativos.
- Saber distinguir los diferentes tipos de sistemas operativos.
- Saber qué son las llamadas al sistema, cómo se invocan y cómo se tratan.
- Conocer cuáles son los principales componentes del núcleo de un sistema operativo.
- Conocer las principales estructuras que puede tener el núcleo de un sistema operativo

Un sistema operativo es una capa de software que gestiona de forma eficiente todos los dispositivos hardware de un computador y además suministra a los usuarios una interfaz cómoda con el hardware

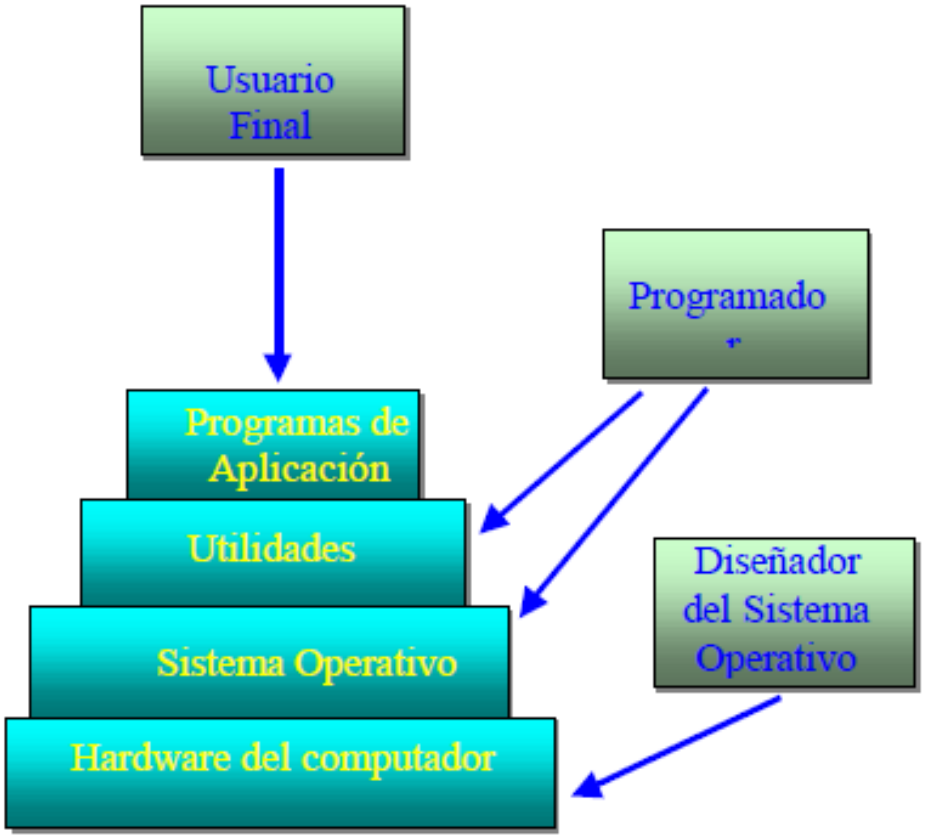
Definición (1/3)

- *El sistema operativo como **máquina virtual o extendida***: Un sistema operativo es una serie de componentes que ocultan la complejidad del hardware y proporcionan abstracciones de mayor nivel (máquina extendida) que facilitan su uso
  Proporciona servicios para:
  - Creación de programas
  - Ejecución de programas
  - Operaciones de Entrada/Salida
  - Manipulación y control del sistema de archivos
  - Detección de errores
  - Control del acceso al sistema
  - Elaboración de informes estadísticos

Definición (2/3)

- *Sistema Operativo como **gestor de recursos***: Un sistema operativo es un conjunto de políticas para gestionar un conjunto de recursos, normalmente escasos, entre un conjunto de procesos que compiten por ellos. El sistema operativo lleva la cuenta del estado de cada recurso y decide quien lo obtiene, durante cuanto tiempo y cuando. Estos recursos son:
  - El procesador
  - La memoria
  - El sistema de archivos
  - Los dispositivos de entrada y salida

Definición (3/3)



Servicios de un sistema operativo

- Ejecución de programas.
- Acceso a los dispositivos de E/S.
- Manipulación del sistema de archivos.
- *Manipulación del sistema de archivos.*
- *Comunicación y sincronización.*
- *Detección y respuesta a errores.*
- *Protección y seguridad.*
- *Contabilidad.*

Historia de los s.o.

- Generación cero: ausencia de s.o. (40-50 años)
- La primera generación: organización de trabajos en lotes (50-60)
- La segunda generación: multiprogramación, multiprocesamiento y sistemas en tiempo real (60-65)
- La tercera generación: incorporación de muchos programas de aplicación y muchas posibilidades diferentes (grandes y costosos (65-75)
- La cuarta generación: estado actual. s.o. para computadores grandes y desarrollo de s.o. para computadores personales. Redes de computadores y sistemas distribuidos

- Primera generación (1945-55)
- Segunda generación (1955-65)
- Tercera generación (1965-80)
- Cuarta generación (1980-hoy)

Primera generación (1945-55)

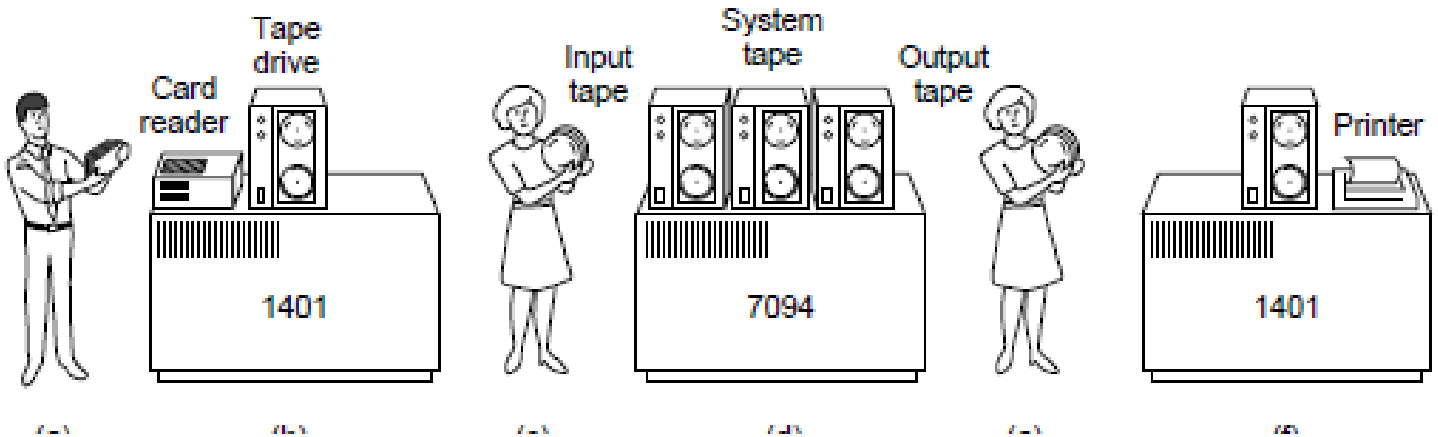
Tubos de vacio y paneles

- Utilidad: máquinas de cálculo.
- Tecnología: dispositivos mecánicos $\Rightarrow$ tubos de vacio y paneles.
- Método de programación: cables $\Rightarrow$ interruptores y tarjetas perforadas.
- Diseño/construcción/operación/programación/-mantenimiento: genios como Aiken, von Newman o Mauchley.

Segunda generación (1955-65)

Transistores y sistemas por lotes

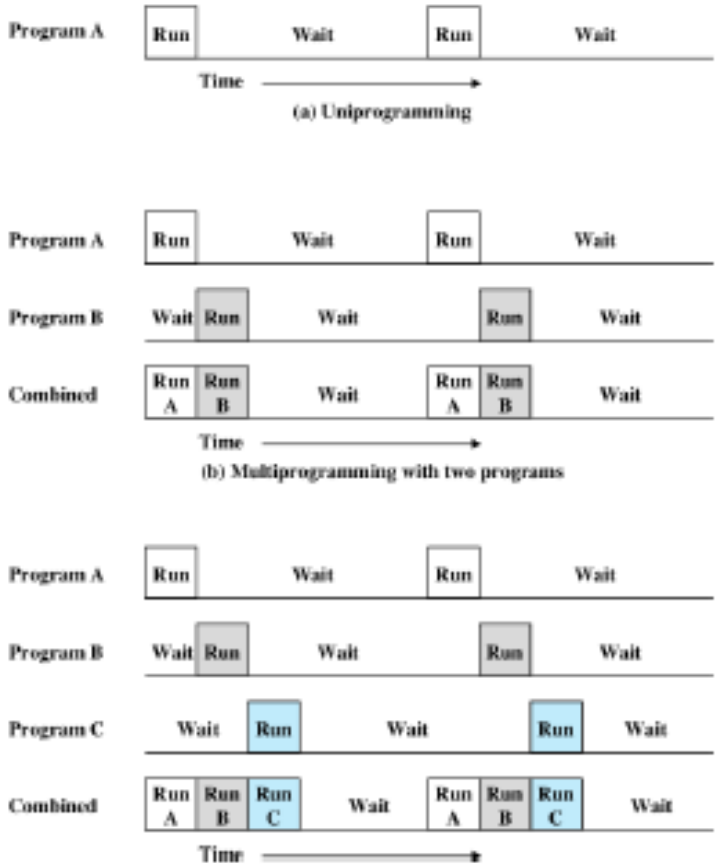
- Utilidad: cálculo científico e ingeniería.
- Tecnología: la invención del transistor redujo su tamaño y precio y los popularizó ⇒ **mainframes/IBM 1401/7094**.
- Método de programación: **ensamblador** y lenguajes de alto nivel (**FORTRAN**) sobre tarjetas perforadas.
- Paso de procesamiento **secuencial** a procesamiento **por lotes**.
- Ejemplos: **FMS** y **IBSYS**.



Tercera generación (1965-80)

Circuitos integrados y multiprogramación

- 2 usos principales:
  - cálculo científico e ingeniería.
  - procesamiento de carácteres.
- **Circuito integrado** ⇒ +barato ⇒ +popular ⇒ **IBM 360**, **GE-645**, **DEC PDP-1**.
- Logros destacables:
  - **multiprogramación**.
  - **spooling**.
  - **tiempo compartido**.
- Ejemplos: **OS/360**, **CTSS**, **MULTICS**, **UNIX**.



Cuarta generación (1980-hoy)

Ordenador personal (era μ)

- **(V)LSI** $\Rightarrow$ ++barato $\Rightarrow$ ++popular $\Rightarrow$ **IBM PC**.
- μ**P**: 8080, Z80, 8086, 286, 386, 486, Pentium, Core 2, Athlon, Alpha, Ultrasparc.
- Logros destacables:
  - **GUI**.
  - **SO de red**.
  - **SMP**.
  - **SO distribuidos**.
- Ejemplos: **UNIX**, **CP/M**, **MS-DOS**, **Linux**, **MacOS**, **XP**, **NT**, **Vista**...

Conceptos fundamentales (1/4)

- Proceso
  - La entidad que puede ser asignada a un procesador y ejecutada por él
  - “Espíritu animado del ordenador”
  - Es un programa en ejecución que consta del programa ejecutable, de los datos necesarios para el programa (variables, espacio de trabajo, buffers, etc.) y del contexto de ejecución del programa (pila del programa, el contador, los registros de datos...)

Introducción a los procesos (2/2)

- Es la unidad más pequeña de trabajo individualmente planificable por un sistema operativo
- La intención de activar un programa ejecutable (compilado y enlazado) es anunciada al sistema operativo mediante una orden especializada o por una llamada al sistema provista para este fin. El sistema operativo responde creando un proceso
- Así pues, el proceso es un concepto dinámico que se refiere a un programa en ejecución, que sufre frecuentes cambios de estado y atributos

Introducción a los procesos (1/2)

- Multitarea: la capacidad que tienen los sistemas operativos de ejecutar de forma simultánea varios procesos

- Multiprocesamiento: un computador que tiene varios procesadores

- Multiprogramación: incluye además de la posibilidad de multitarea, la capacidad de la gestión de memoria y de los ficheros

Monousuario - Multiusuario
Monoprogramados – multiprogramados
Tiempo compartido

Monoprogramado multiprogramado

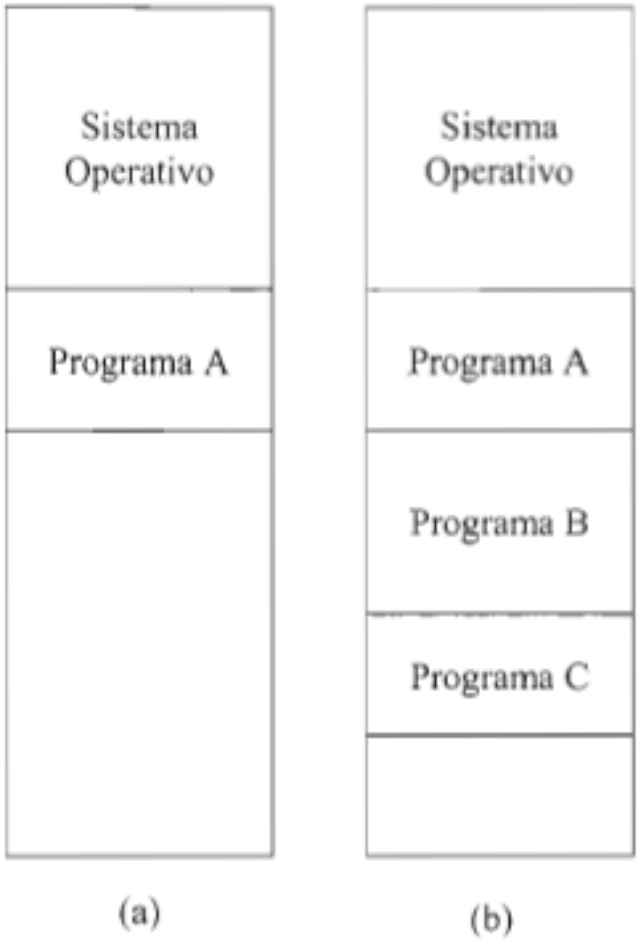


Figura 1.2 – Esquema de distribución de la memoria principal en sistemas operativos con monoprogramación (a) y con multiprogramación (b)

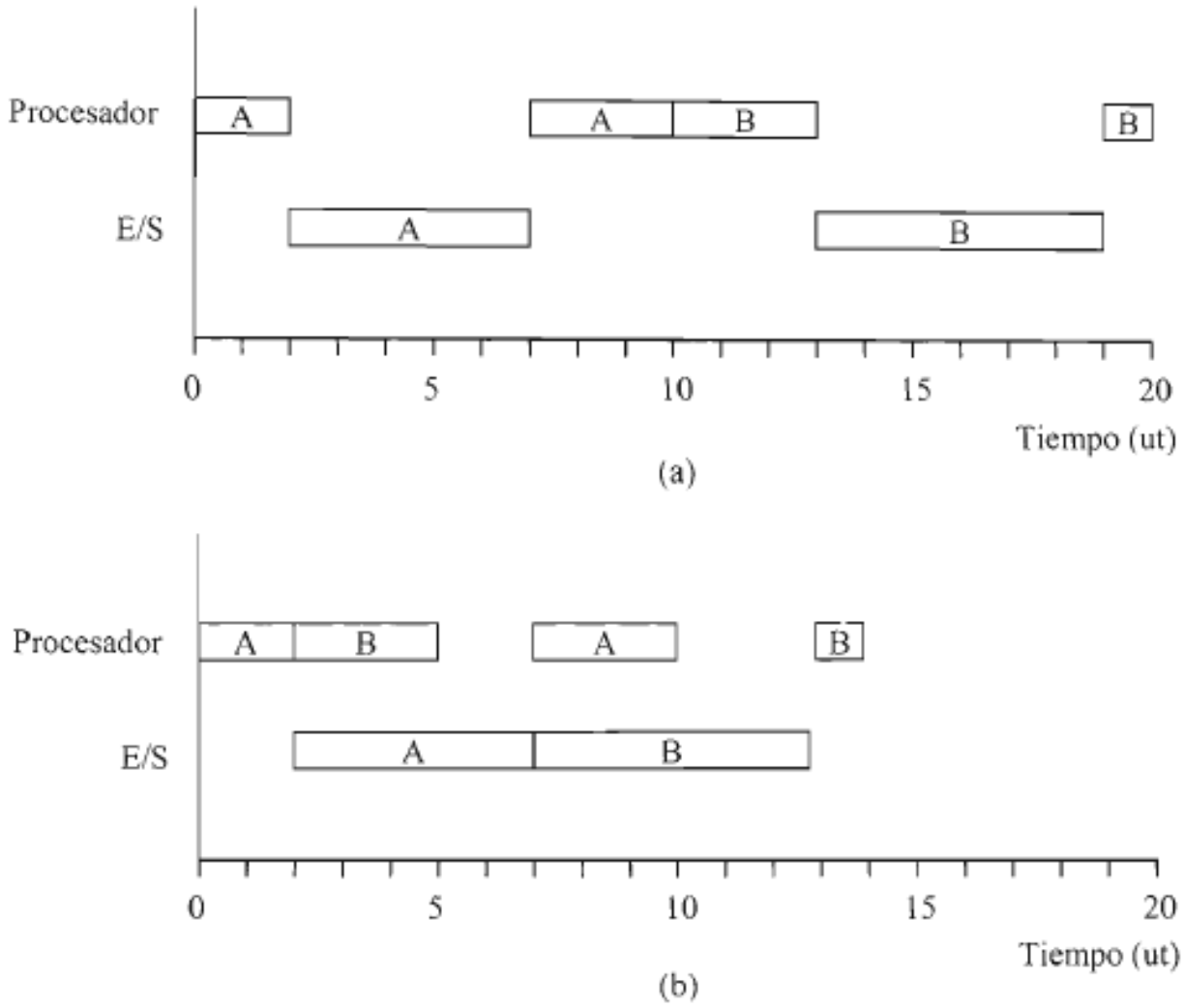


Figura 1.3 – Diagrama de uso del procesador y de los dispositivos de E/S del Ejemplo 1.1 sin usar multiprogramación (a) y usando multiprogramación (b)

Clasificación Requisitos temporales

- *Sistemas operativos por lotes* o *sistemas batch*
- *Sistemas operativos de tiempo compartido* o *sistemas interactivos.*
- *Sistemas operativos híbridos*
- *Sistemas operativos de tiempo real*

Otras clasificaciones

- *Sistemas operativos para macrocomputadores*
- *Sistemas operativos para servidores de red.*
- *Sistemas operativos para computadores personales.*
- *Sistemas operativos para computadores de mano*
- *Sistemas operativos integrados (empotrados)*
- *Sistemas operativos paralelos* o *sistemas multiprocesador*
- *Sistemas operativos distribuidos*
- *Sistemas operativos de red.*

Conceptos fundamentales (3/4)

- Llamadas al sistema:
  - Este concepto corresponde al interfaz entre el s.o. y los programas y los usuarios
    - Los programas invocan generalmente los servicios del s.o. en tiempo de ejecución por medio de llamadas al sistema operativo
    - Los usuarios pueden interactuar con el sistema operativo directamente por medio de **órdenes**. Estas órdenes suelen ser traducidas y se ejecutan como una serie de llamadas al sistema
  - Las llamadas se pueden agrupar en cinco categorías:
    - Control de Procesos
    - Manipulación de archivos
    - Manipulación de periféricos
    - Mantenimiento de la información
    - Comunicaciones

Llamadas al sistema

Descripción	UNIX	Windows
Crear un proceso nuevo	fork	CreateProcess
Esperar por la terminación de un proceso	waitpid	WaitForSingleObject
Finalizar la ejecución de un proceso	exit	ExitProcess
Crear un archivo o abrir uno ya existente	open	CreateFile
Cerrar un archivo	close	CloseHandle
Leer un archivo	read	ReadFile
Escribir un archivo	write	WriteFile
Mover el puntero de lectura/escritura de un archivo	lseek	SetFilePointer
Obtener los atributos de un archivo	stat	GetFileAttributeEx
Crear un directorio	mkdir	CreateDirectory
Borrar un directorio	rmdir	RemoveDirectory
Cambiar el directorio de trabajo	chdir	SetCurrentDirectory

Tabla 1.1 – Ejemplos de llamadas al sistema de UNIX y Windows

Conceptos fundamentales (2/4)

- Gestión de memoria y del sistema de archivos:
  - Aislar los procesos, de manera que un proceso no interfiera en los datos o en la memoria de otro
  - Ubicar y gestionar automáticamente a los procesos, de manera que sea transparente a los programadores
  - Soportar programación modular, de forma que se puedan definir módulos de programas en los que se pueda alterar dinámicamente sus tamaños
  - Controlar el acceso y proteger la memoria
  - Disponer de un medio de almacenamiento de larga duración

Conceptos fundamentales (4/4)

- Gestión y planificación de recursos:
  - Gestionar los diferentes recursos que disponga el sistema (procesadores, memoria, periféricos...)
  - Planificar la utilización de los mismos por los procesos en ejecución de forma
    - justa
    - eficiente

Estructura de los s.o.

- La estructura usual corresponde a los siguientes componentes:
  - Gestor de procesos
  - Gestor de la memoria principal
  - Gestor del almacenamiento secundario y del sistema de archivos
  - Gestor del sistema de E/S
  - Sistema de protección
  - Sistema de comunicación
  - Intérprete de órdenes

Interconectar estas partes de forma que formen un núcleo de manera: Modular + niveles jerárquicos y abstracción de información

Diseño e implementación de los s.o.

- Requisitos de usuario: El sistema debe ser cómodo, fácil, sencillo de utilizar y aprender, fiable, seguro y rápido
- Requisitos del sistema: El sistema debe ser fácil de diseñar, implementar y mantener, flexible, fiable, sin errores y eficiente
- Considerar su posible evolución
- Lenguaje de alto nivel (ciertas partes muy dependientes del hardware escritas en ensamblador)

Clasificación de SO según su estructura

¿Cómo se organiza internamente el SO?

- Clasificación:
  - Desestructurados.
  - Estructura simple:
    - monolíticos
    - capas
    - modulares
  - Estructura cliente/servidor:
    - micronúcleo
    - exonúcleo
  - Máquina virtual.
  - Híbridos.
- Tendencias:
  - Núcleos extensibles.
  - Multiservidores sobre un micronúcleo.
  - Núcleos híbridos.

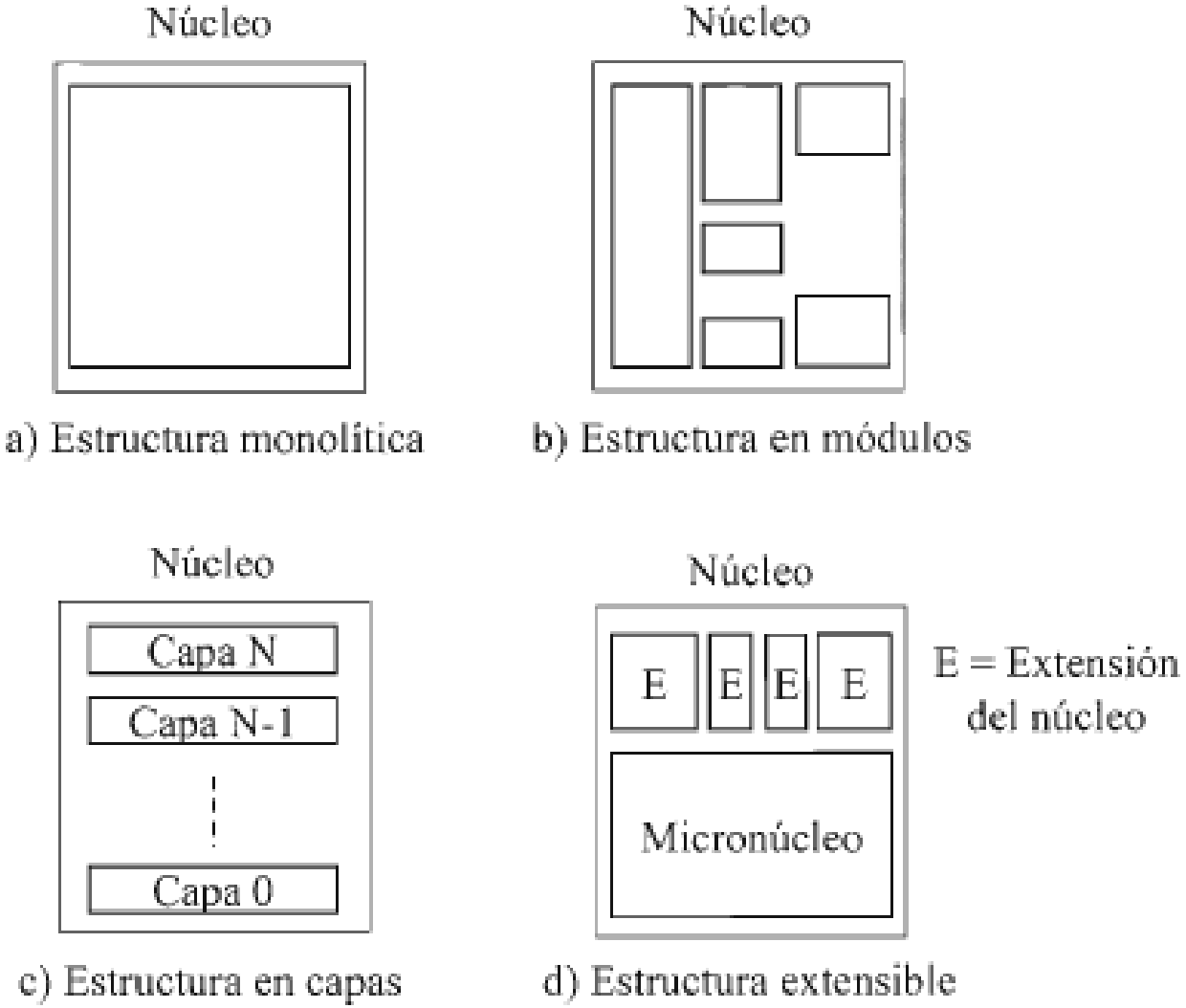


Figura 1.5 – Principales estructuras del núcleo de un sistema operativo

Cliente - servidor

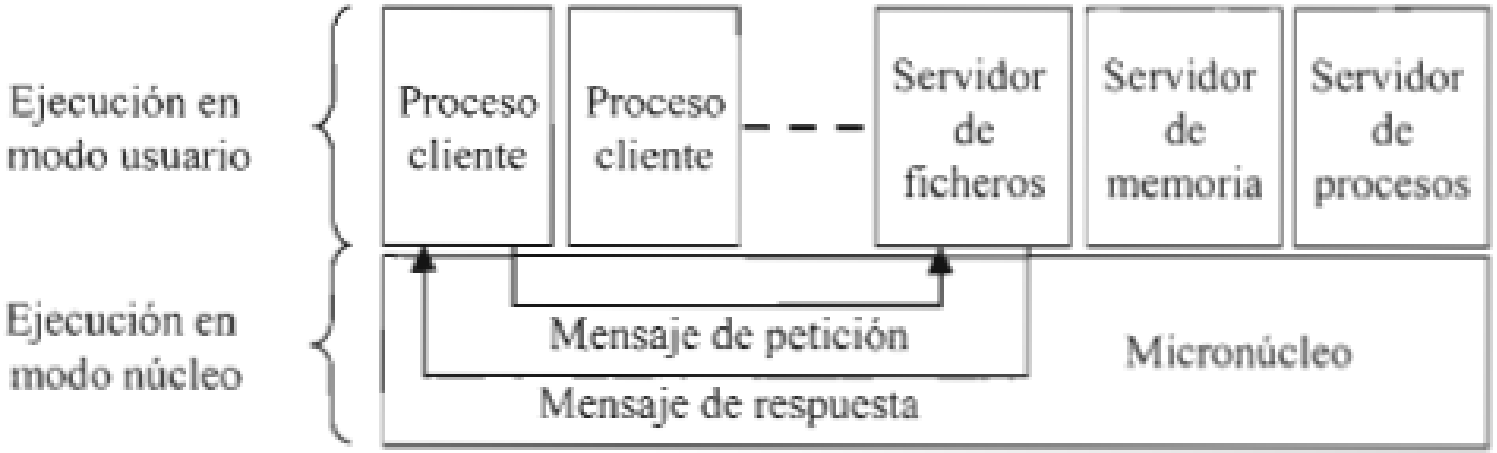


Figura 1.6 – Modelo cliente-servidor utilizado en la estructura de tipo extensible

Extensible

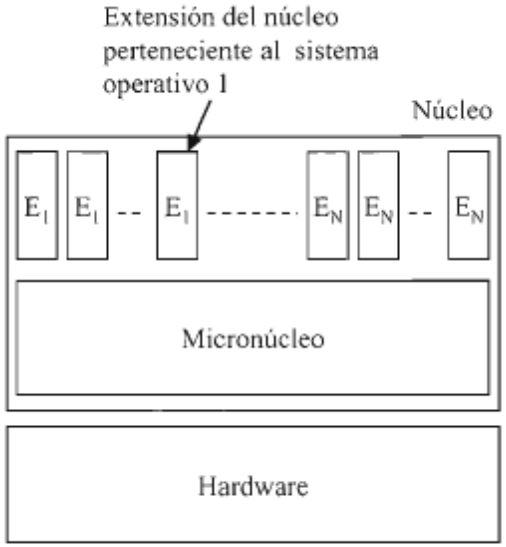
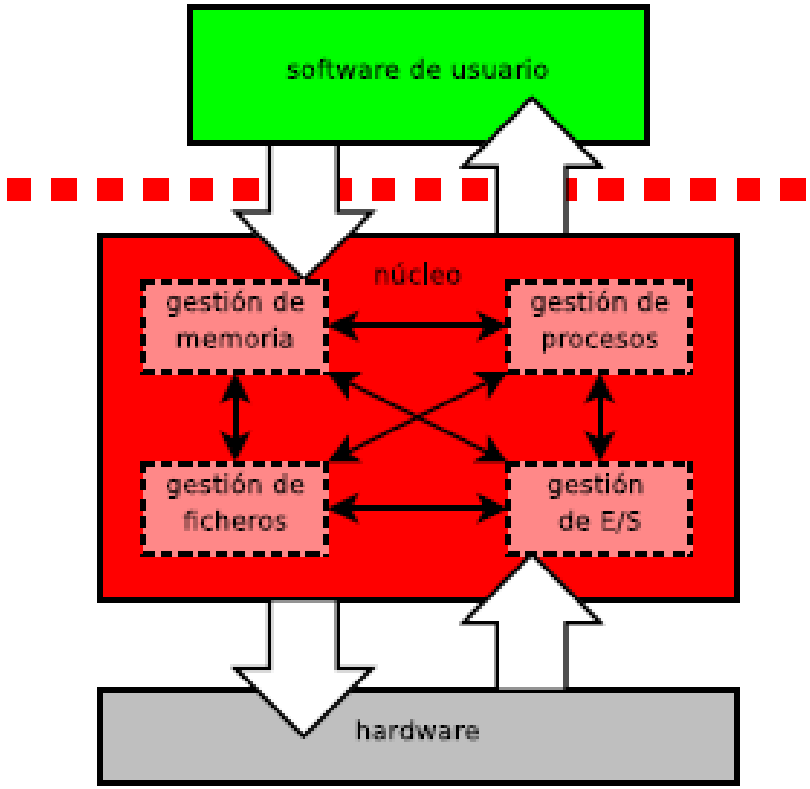


Figura 1.7 – Diagrama de una estructura extensible que soporta múltiples sistemas operativos

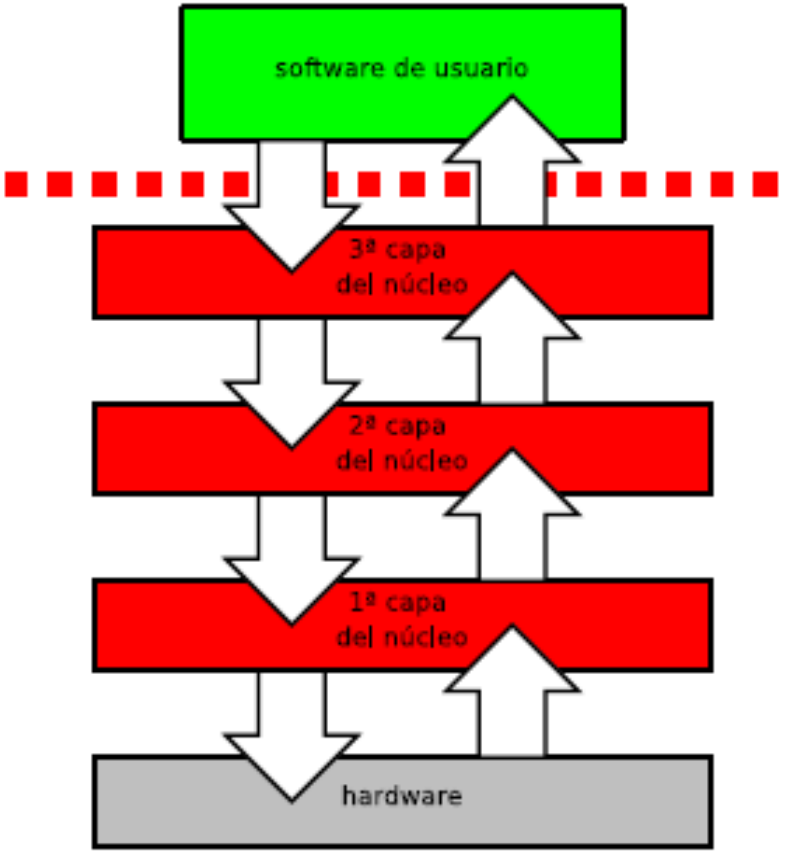
Monolítico

- El SO completo se ejecuta en modo protegido.
- Nula protección entre los componentes.
- Ventajas:
  - Economía de cambios de contexto $\Rightarrow$ **+eficiente**.
- Inconvenientes:
  - Falta de protección $\Rightarrow$ **-fiabilidad** (controladores).
  - Manejo de la complejidad: Es más sencillo escribir 10^3 programas de 10^3 líneas que uno de 10^6.



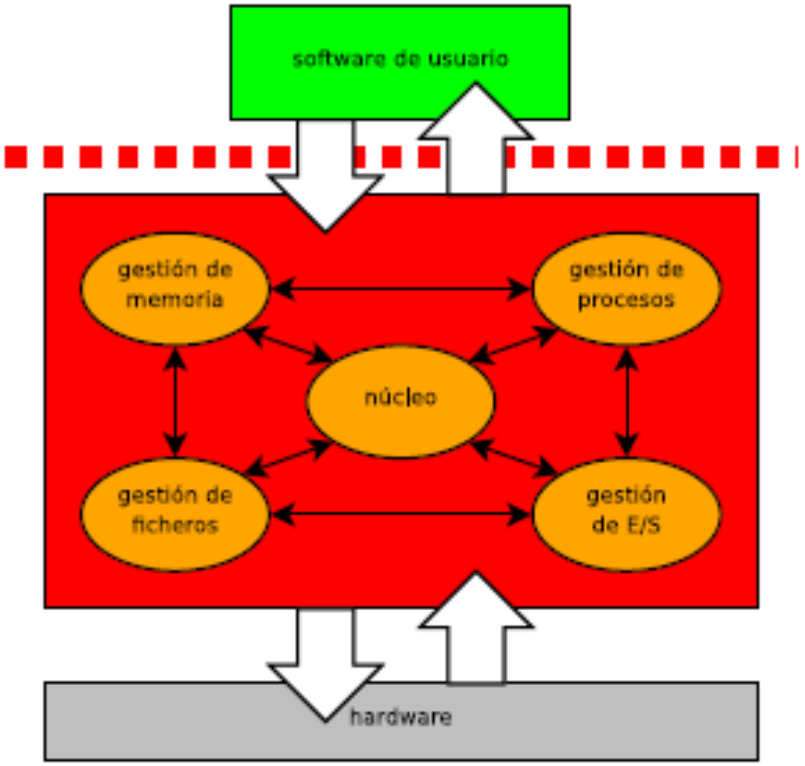
Capas/Niveles

- El SO completo se ejecuta en modo protegido.
- Escasa protección entre los componentes.
- Ventajas:
  - Economía de cambios de contexto ⇒ **+eficiente**.
  - Menor complejidad.
- Inconvenientes:
  - Falta de protección ⇒ **-fiabilidad** (controladores).
  - Menos flexible que monolítico.
- ¿Cómo subdividir en capas?



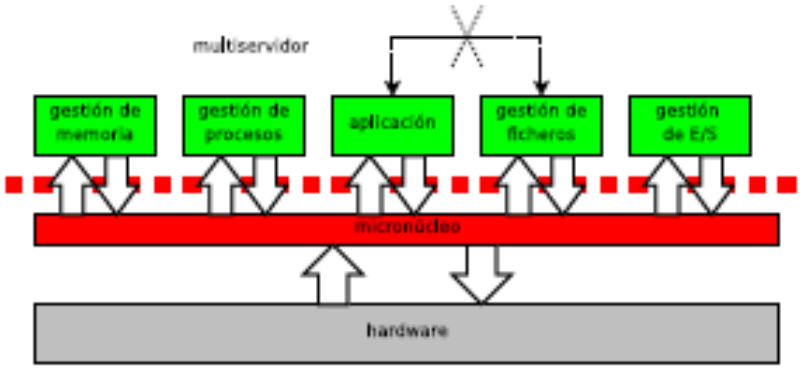
Modular

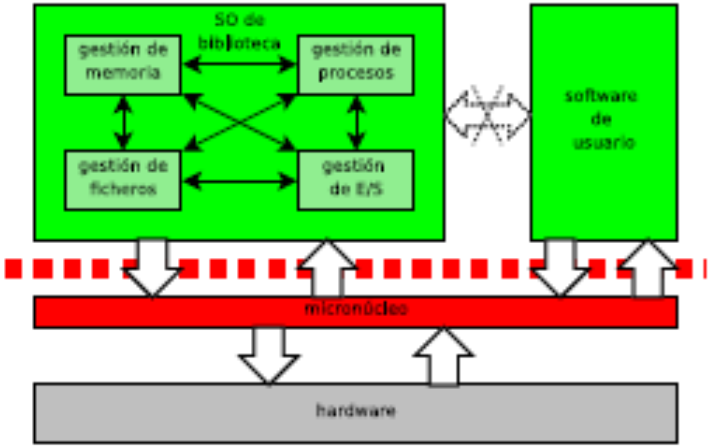
- El SO completo se ejecuta en modo protegido.
- Escasa protección entre los componentes.
- Ventajas:
  - Economía de cambios de contexto $\Rightarrow$ **+eficiente**.
  - Menor complejidad.
- Inconvenientes:
  - Falta de protección $\Rightarrow$ **-fiabilidad** (controladores).
  - Menos flexible que monolítico.
- ¿Qué colocar en el núcleo y qué en módulos?



Micronúcleo

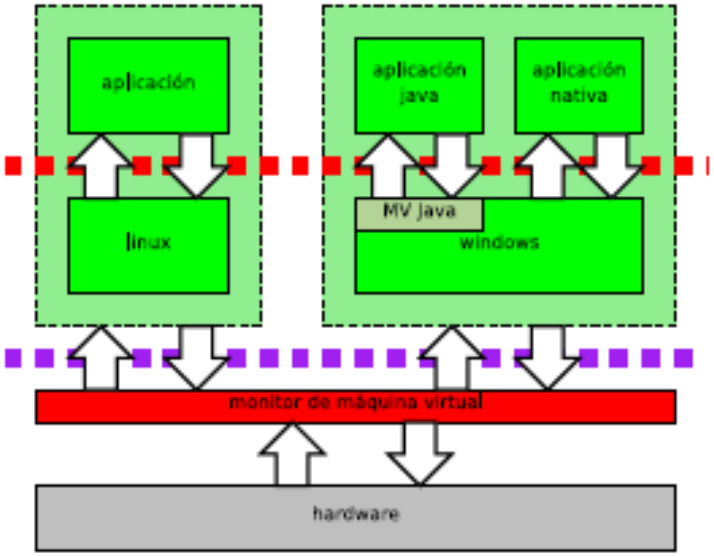
- Una mínima parte del SO se ejecuta en modo protegido.
- Ventajas:
  - Perfecta protección entre componentes ⇒ **+fiabilidad**.
  - Manejo de la complejidad.
  - Facilidad de programación.
- Inconvenientes:
  - Sobrecarga en las comunicaciones ⇒ **-eficiencia**.





Máquina virtual

- N **copias virtuales** de la máquina real:
  - Software: Bochs, Qemu, VMWare, Xen.
  - Hardware: VMWawe, Xen.
- IBM VM/370 (1972).
- Ventajas:
  - Perfecta protección entre componentes ⇒ **+fiabilidad**.
  - Mejor aprovechamiento del hardware.
  - Máxima reutilización de código.
- Inconvenientes:
  - La simulación del hardware real es costosa ⇒ **poco eficiente**



Híbrida

- Mezcla más frecuente: **micronúcleo + monolítico**.
- Ventaja: $\Longrightarrow$ ganamos velocidad respecto a micronúcleo.
- Inconveniente: $\Longrightarrow$ perdemos protección entre componentes.