

Introducción a los Sistemas Operativos

Capítulo 2

Sistema Operativo

Es difícil definirlo...

- Un programa que controla la ejecución de los programas de aplicación
- Un interfaz entre las aplicaciones y el hardware

Objetivos del Sistema Operativo

- Comodidad
 - Hace que el ordenador sea más cómodo de usar
- Eficiencia
 - Permite un uso eficiente de los recursos del sistema
- Capacidad de evolucionar
 - Permitir desarrollo efectivo, testeo e introducción de nuevas funcionalidades sin interferir con el servicio

El Sistema Operativo como Interfaz Usuario/Computador

Niveles del Computador

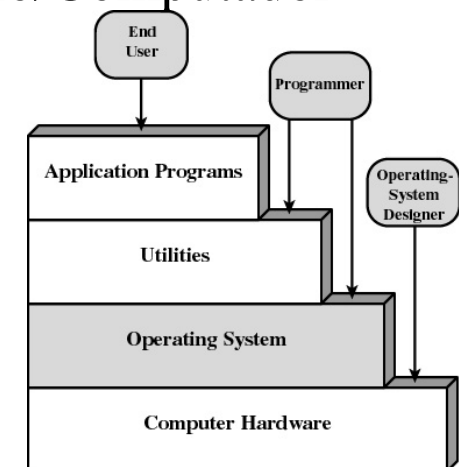


Figure 2.1 Layers and Views of a Computer System

Servicios proporcionados por el Sistema Operativo

- Desarrollo de programas
 - Editores y depuradores (debuggers)
- Ejecución de programas
- Acceso a dispositivos de entrada/salida
- Controlar el acceso a los ficheros
- Acceso al sistema

Servicios proporcionados por el Sistema Operativo

- Detección y respuesta a errores
 - Errores hardware internos y externos
 - error de memoria
 - fallo en un dispositivo
 - errores software
 - desbordamiento aritmético (overflow)
 - acceso prohibido a posiciones de memoria
 - el S.O. no puede conceder la petición de una aplicación

Servicios proporcionados por el Sistema Operativo

- Contabilidad
 - realizar estadísticas
 - monitorizar el rendimiento
 - usado para anticipar futuras mejoras
 - usado para cobrarle a los usuarios

El Sistema Operativo como Gestor de Recursos

- Funciona igual que cualquier otro programa de ordenador
 - Es un programa que se ejecuta
- El sistema operativo renuncia al control del procesador para que lo usen otros programas
 - Depende del procesador para volver a tomar el control
- No realiza trabajo neto
 - Sólo dirige al procesador en el uso de recursos y en la temporización de otros trabajos

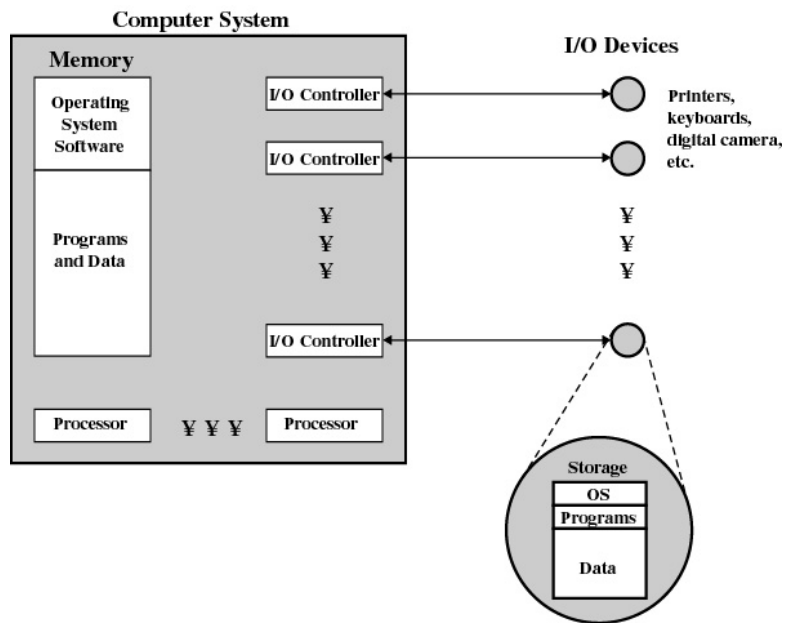


Figure 2.2 The Operating System as Resource Manager

Núcleo (Kernel)

- Parte del sistema operativo que está en memoria principal
- Contiene las funciones más frecuentemente utilizadas

Evolución de un Sistema Operativo

- Actualizaciones de hardware y nuevos tipos de HW
- Nuevos servicios
- Mejoras

Evolución de los Sistemas Operativos

- Procesamiento Serie (final de los 40, mitad de los 50)
 - Sin sistema operativo
 - Las máquinas ejecutan desde una consola con luces e interruptores, dispositivo de entrada e impresora
 - Libro de reserva (Schedule tome)
 - El 'setup' incluía la carga del compilador, el programa fuente, salvar el programa compilado, cargarlo y linkarlo

Evolución de los Sistemas Operativos

- Sistemas con procesamiento por lotes
 - Mitad de los años 50, principios de los 60
 - Monitores
 - Software que controla los programas que se ejecutan
 - Los lotes se ejecutan juntos
 - El programa devuelve el control al monitor cuando termina
 - El monitor residente permanece en memoria principal y está listo para ejecutarse

Lenguaje de control de procesos (JCL)

- Tipos especial de lenguaje de programación
- Proporciona instrucciones al monitor
 - qué compilador utilizar
 - qué datos usar

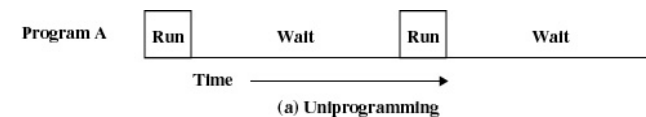
Características Hardware

No son estrictamente necesarias aunque se incluían:

- Protección de memoria
 - Que no se altere el espacio de memoria del monitor
- Temporización
 - Evitar que un trabajo monopolice el sistema
- Instrucciones privilegiadas
 - Instr. máquina que sólo debe ejecutar el monitor: E/S,...
- Interrupciones
 - Computadores más modernos ofrecen más flexibilidad

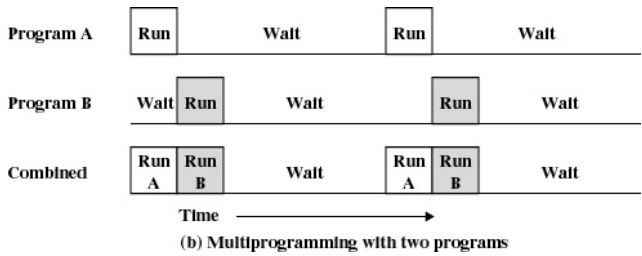
‘Monoprogramación’

- El procesador debe esperar a que terminen las instrucciones de entrada/salida para poder continuar



Multiprogramación

- Cuando un trabajo necesita esperar entrada/salida, el procesador puede cambiar a otro trabajo



Multiprogramación

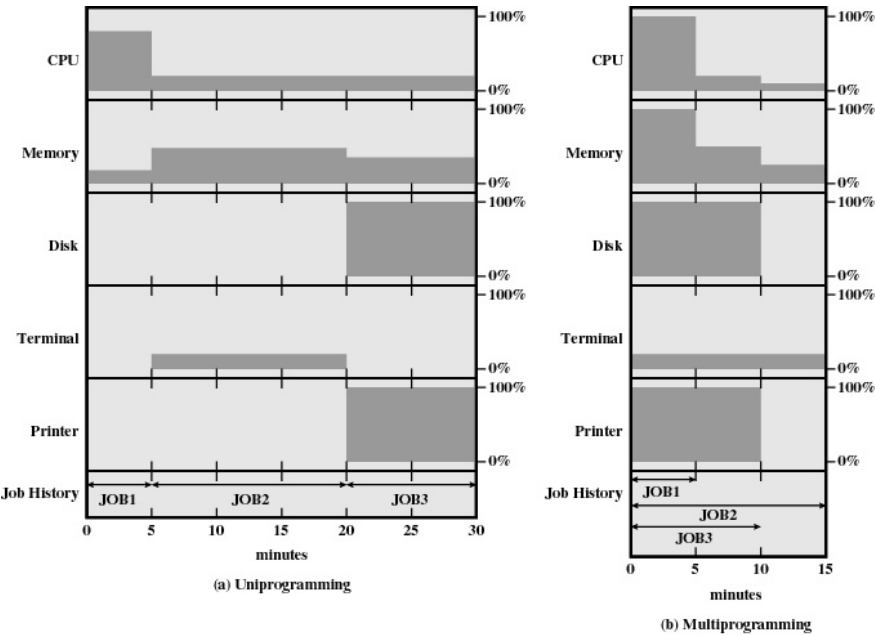
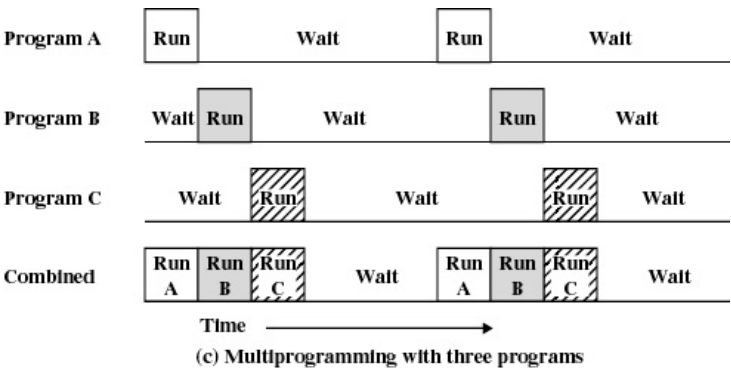


Figure 2.6 Utilization Histograms

Ejemplo

| | TRABAJO1 | TRABAJO2 | TRABAJO3 |
|-------------------|-----------------|----------|----------|
| Tipo de trabajo | Cálculo intenso | I/O | I/O |
| Duración | 5 min. | 15 min. | 10 min. |
| Memoria | 50 K | 100 K | 80 K |
| Necesita disco | No | No | SI |
| Necesita terminal | No | SI | No |
| Necesita imprimir | No | No | SI |

Efectos de la Multiprogramación

| | Monoprogramación | Multiprogramación |
|---------------------------|------------------|-------------------|
| Uso del Procesador | 22% | 43% |
| Uso de la Memoria | 30% | 67% |
| Uso del Disco | 33% | 67% |
| Uso de la Impresora | 33% | 67% |
| Tiempo Transcurrido | 30 min. | 15 min. |
| Productividad (ratio) | 6 trabajos/hr | 12 trabajos/hr |
| Tiempo Medio de Respuesta | 18 min. | 10 min. |

Tiempo Compartido

- Utilizar multiprogramación para gestionar varios programas interactivos
- El tiempo del procesador se comparte entre múltiples usuarios
- Muchos usuarios simultáneamente acceden al sistema a través de terminales

Multiprogramación por lotes versus Tiempo Compartido

| | Multiprogramación por Lotes | Tiempo Compartido |
|---|--|---|
| OBJETIVO PRINCIPAL | MAXIMIZAR EL USO DEL PROCESADOR | MINIMIZAR EL TIEMPO DE RESPUESTA |
| Fuente de directivas al sistema operativo | Comandos del lenguaje de control de trabajos proporcionados con el trabajo | Comandos introducidos desde el terminal |

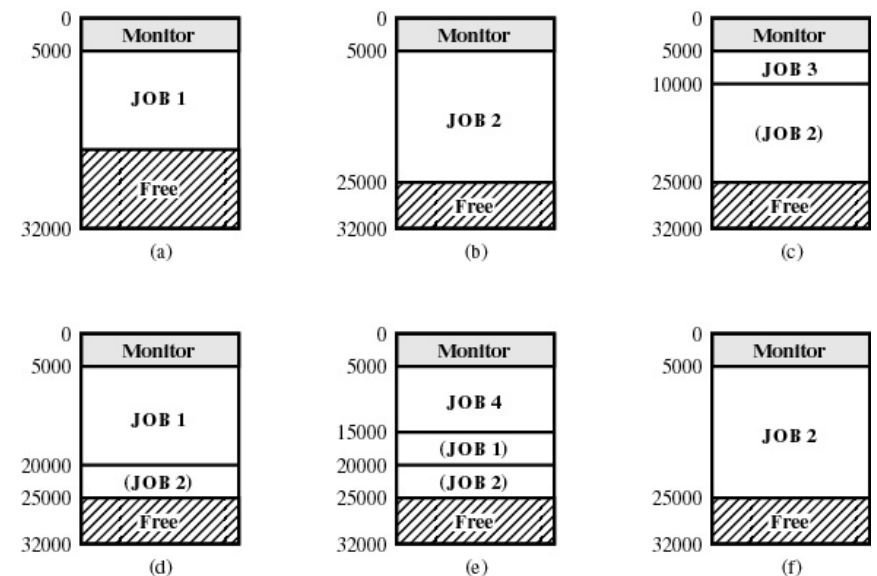


Figure 2.7 CTSS Operation

Mayores Logros

- Procesos
- Gestión de Memoria
- Protección y seguridad de la Información
- Planificación y gestión de recursos
- Estructura del Sistema

Procesos

- Un programa en ejecución
- Una instancia de un programa ejecutando en un ordenador
- La entidad que puede ser ‘asignada a’ y ‘ejecutada en’ un procesador
- Una unidad de actividad caracterizada por: un hilo secuencial de ejecución, un estado actual y un conjunto de recursos asociados

Dificultades con el diseño del software del sistema

- Sincronización inadecuada
 - Asegurar que un proceso que espera una entrada/salida reciba una y sólo una señal
- Fallo en la exclusión mutua
- Comportamiento indeterminado
 - los programas deben depender sólo de la entrada proporcionada, no debe confiar en áreas de memoria común
- Interbloqueos (deadlocks)

Procesos

- Consta de tres componentes
 - Un programa ejecutable
 - Los datos asociados que necesita el programa
 - El contexto de ejecución del programa
 - Toda la información que necesita el sistema operativo para gestionar el proceso

Process

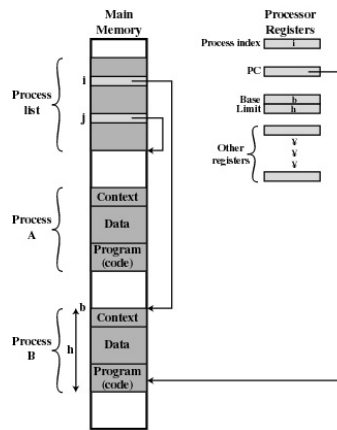


Figure 2.8 Typical Process Implementation

Gestión de Memoria

- Aislar procesos
- Gestión y asignación automática
- Soporte para programación modular
- Protección y control de acceso
- Almacenamiento de larga duración

Memoria Virtual

- Permite a los programadores direccionar la memoria desde un punto de vista lógico
- Evitar que haya un vacío entre el desalojo de un proceso a disco y la carga del siguiente
- Necesita soporte hardware
- El HW y el SO proporcionan al usuario un “procesador virtual”

Sistema de Ficheros

- Implementa el almacenamiento de larga duración (long-term)
- La información se almacena en objetos con nombres llamados ficheros

Paginación

- Permite a los procesos estar constituidos por un número de bloques de tamaño fijo, llamados páginas
- La dirección virtual se descompone en un número de página y un desplazamiento dentro de la página
- Cada página puede situarse en cualquier sitio de la memoria (marcos de página, frames)
- La dirección real o física se refiere a la memoria principal

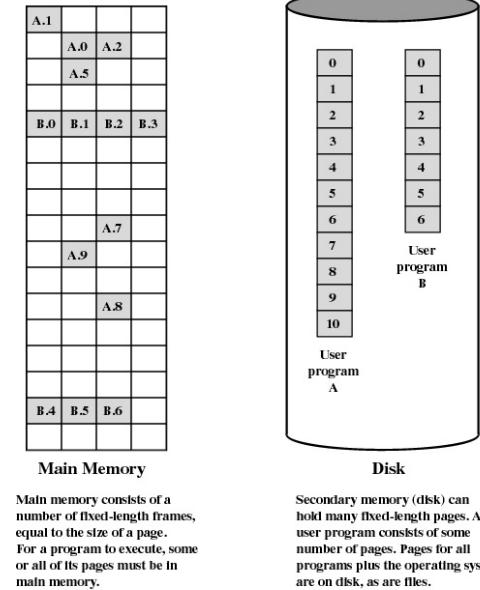


Figure 2.9 Virtual Memory Concepts

Direccionamiento de la Memoria Virtual

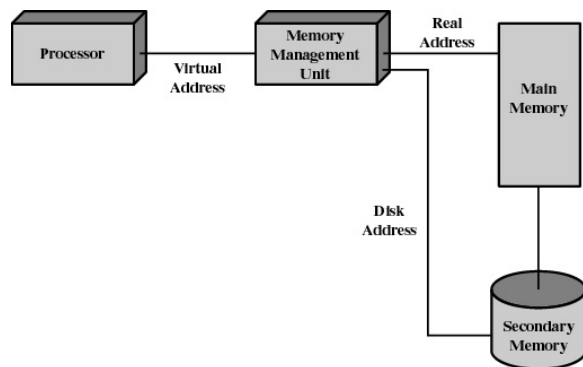


Figure 2.10 Virtual Memory Addressing

Seguridad y Protección de la Información

- Control de Acceso
 - regular el acceso de los usuarios al sistema
- Control del flujo de la Información
 - regular el flujo de datos entre el sistema y la entrega a los usuarios
- Certificación
 - comprobar que los controles de acceso y de flujo se realizan de acuerdo a las especificaciones

Planificación y Gestión de Recursos

- Justicia
 - Proporcionar un acceso igual y justo a todos los procesos de la misma clase (prioridad)
- Respuesta diferencial
 - discriminar entre diferentes clases de trabajos
- Eficiencia
 - maximizar la productividad, minimizar el tiempo de respuesta y acomodar al mayor numero de usuarios posible

Elementos Principales del Sistema Operativo

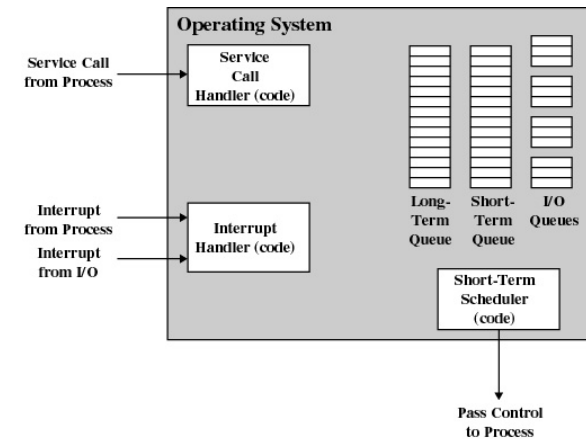


Figure 2.11 Key Elements of an Operating System for Multiprogramming

Estructura del Sistema

- Vista del sistema como una serie de niveles
- Cada nivel realiza un conjunto de funciones
- Cada nivel confía en el inmediato inferior para que realice la funciones primitivas
- Esto descompone el problema en un número de subproblemas manejables

Diseño de Sistemas Operativos Jerarquía

| Nivel | Nombre | Objetos | Ejemplos de Operaciones |
|-------|----------------------|--|---|
| 13 | Shell | Entorno de programación del usuario | Comandos en lenguaje del shell |
| 12 | Procesos del usuario | Procesos del usuario | Quit, kill, suspend, resume |
| 11 | Directorios | Directorios | Create, destroy, attach, detach, search, list |
| 10 | Dispositivos | Dispositivos externos, tales como impresoras, pantallas y teclados | Open, close, read, write |
| 9 | Sistema de ficheros | Ficheros | Create, destroy, open, close read, write |
| 8 | Comunicaciones | Pipes (tuberías) | Create, destroy, open. close, read, write |

Diseño de Sistemas Operativos Jerarquía

| Nivel | Nombre | Objetos | Ejemplos de Operaciones |
|-------|---------------------------------|--|-------------------------------|
| 7 | Memoria Virtual | Segmentos, páginas | Read, write, fetch |
| 6 | Almacenamiento secundario local | Bloques de datos, canales de dispositivos | Read, write, allocate, free |
| 5 | Primitivas de procesos | Primitivas de procesos, semáforos, lista de preparados, bloqueados | Suspend, resume, wait, signal |

Diseño de Sistemas Operativos Jerarquía

| Nivel | Nombre | Objetos | Ejemplos de Operación |
|-------|---------------------------|--|---------------------------------------|
| 4 | Interrupciones | Rutinas de tratamiento de interrupción | Invoke, mask, unmask, retry |
| 3 | Procedimientos | Procedimientos, pila de llamadas | Mark stack, call, return |
| 2 | Conjunto de Instrucciones | Pila de evaluación, interprete de micro-programa, datos escalares y arrays | Load, store, add, subtract branch |
| 1 | Circuitos Electrónicos | Registros, puertas, buses, etc. | Clear, transfer, activate, complement |

Características de los Sistemas Operativos Modernos

- Arquitectura de Microkernel
 - asigna sólo unas pocas funciones esenciales al kernel
 - direccionamiento del espacio
 - comunicación interprocesos (IPC)
 - planificación básica

Características de los Sistemas Operativos Modernos

- Multithreading
 - el proceso se divide en threads que se pueden ejecutar simultáneamente
- Thread (en español: hilos, hebras)
 - unidad de trabajo planificable
 - se ejecuta secuencialmente y es interrumpible
- Un proceso es una colección de uno o más threads más los recursos asociados

Características de los Sistemas Operativos Modernos

- Multiprocesamiento simétrico (SMP)
 - hay múltiples procesadores
 - estos procesadores comparten el mismo espacio de memoria y I/O
 - todos los procesadores pueden realizar las mismas funciones

Características de los Sistemas Operativos Modernos

- Sistemas operativos distribuidos
 - clusters de PCs, cada máquina es completa por separado: procesador, memoria, disco, ...
 - proporciona la ilusión de una única memoria principal y un único espacio de memoria secundaria (disco)
 - usado para sistemas de ficheros distribuidos

Características de los Sistemas Operativos Modernos

- Diseño Orientado a Objetos
 - usado para añadir extensiones modulares a un kernel pequeño
 - permite a los programadores ‘personalizar’ un sistema operativo sin afectar a la integridad del sistema
 - simplifica el desarrollo de sistemas distribuidos

Windows 2000

- Explota la potencia de los procesadores actuales de 32 bits
- Proporciona multitarea real en un entorno mono-usuario
- Computación Cliente/Servidor

Arquitectura del Windows 2000

- Estructura modular por flexibilidad
- Se ejecuta en una gama de plataformas hardware
- Soporta aplicaciones escritas para otros sistemas operativos (windows 95, 98,..., MSDOS,...)

Organización del SO

- Arquitectura de microkernel modificada
 - No es un microkernel puro
 - Muchas funciones del externas al microkernel se ejecutan en modo kernel
- Cualquier módulo puede ser eliminado, actualizado o reemplazado sin necesidad de volver a escribir el sistema entero

Estructura por capas

- Capa de abstracción del hardware (HAL)
 - Aisla al sistema operativo de la diferencias entre plataformas hardware específicas
- Microkernel
 - Los componentes más usados y fundamentales del sistema operativo
- Drivers de dispositivos
 - Traducen llamadas de I/O del usuario peticiones al dispositivo hardware específico

Ejecutiva del W2K

- Gestor de entrada/salida
- Gestor de objetos
- Monitor de seguridad
- Planificador de Procesos/threads
- Servicio de llamadas a procedimiento local (LPC)
- Gestor de memoria virtual
- Gestor de cache (de disco)
- Módulos de ventana y gráficos

Procesos de usuario

- Procesos de soporte especial del sistema
 - Ej.: proceso de conexión y gestión de sesión
- Procesos servidores
- Subsistema de entorno
- Aplicaciones de usuario

Modelo cliente/servidor

- Simplifica la ‘ejecutiva’
 - Es posible construir una variedad de APIs
- Mejora la fiabilidad
 - Cada servicio se ejecuta en un proceso aparte con su propia partición de memoria
 - los clientes no pueden acceder al hardware directamente
- Proporciona una forma uniforme para que las aplicaciones se comuniquen via LPC
- Proporciona una base para la programación distribuida

Threads y SMP

- Diferentes rutinas pueden ejecutarse simultáneamente en diferentes procesadores
- Múltiples threads de ejecución de un proceso pueden ejecutarse en diferentes procesadores simultáneamente.
- Los procesadores servidores pueden usar múltiples threads
- Comparten datos y recursos entre procesos

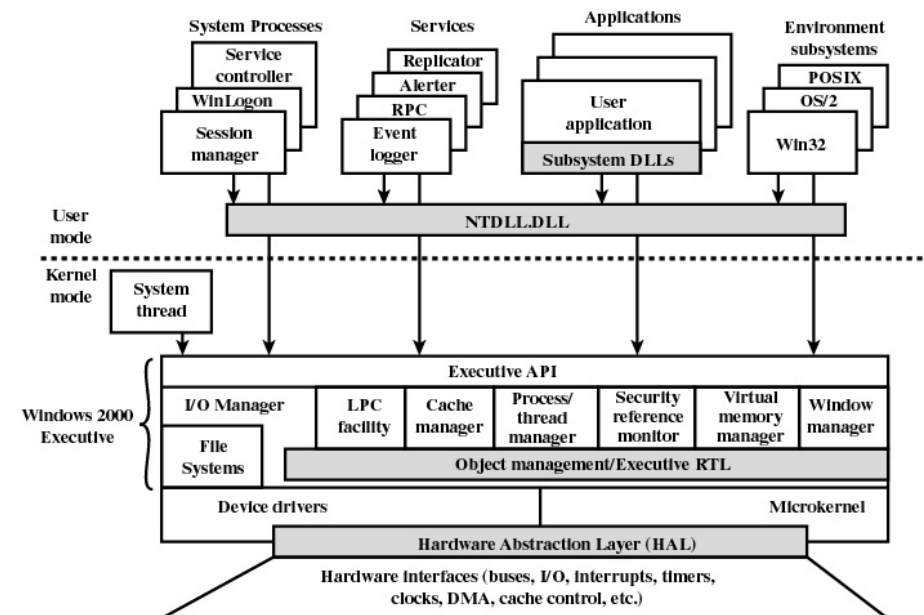


Figure 2.13 Windows 2000 Architecture

UNIX

- El hardware es ‘rodeado’ por el sistema operativo
- El sistema operativo es lo que se llama kernel
- Viene con un conjunto de servicios e interfaces
 - shell
 - compilador de C
 - ...

UNIX

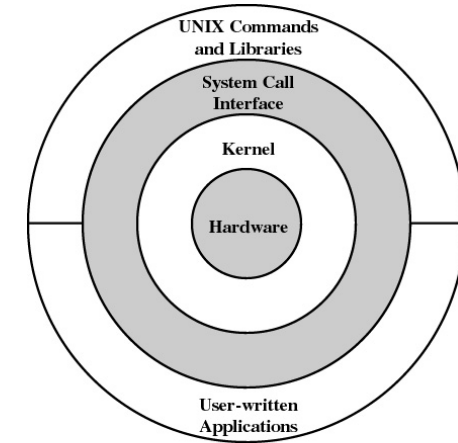


Figure 2.15 General UNIX Architecture

Sistemas UNIX modernos

- System V Release 4 (SVR4)
- Solaris 2.x
- 4.4BSD
- Linux