

Sistemas Operativos

Introducción

Curso 2024

Facultad de Ingeniería, UDELAR

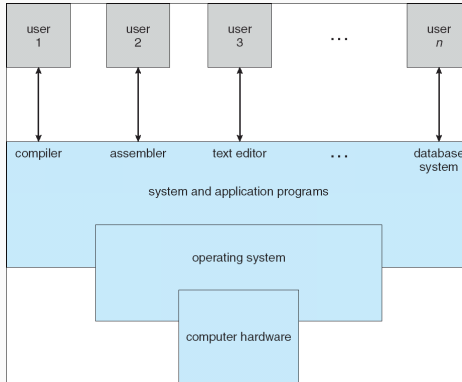
1. Introducción a los sistemas operativos
2. Evolución histórica de los sistemas operativos

Introducción a los sistemas operativos

Introducción

¿Qué es un sistema operativo?

- Es un programa
- Funciona como intermediario entre el usuario y los programas y el hardware



- Metas
 - Brindar un entorno para que los usuarios puedan ejecutar programas en forma conveniente
 - Brindar un entorno para que los programas usen el hardware con facilidad
 - Administrar el hardware de forma eficiente y equitativa
 - Proveer un entorno sin interferencias a cada usuario
- Todas las aplicaciones requieren un conjunto de operaciones comunes que son incorporadas al sistema operativo

Funciones básicas

- Administración de procesos
- Manejo de interrupciones
- Administración de memoria
- Manejo del sistema de archivos
- Administración de seguridad
- Control de entrada/salida

- El sistema operativo es un:
 - Administrador de recursos
 - Administra todos los recursos disponibles.
 - Decide como asignar estos recursos a los programas que los requieren según los pedidos y asignaciones que tenga.
 - Programa de control y abstracción del hardware
 - Controla la ejecución de los programas para la prevención de errores y mal uso del sistema.
 - Implementa funciones comunes de acceso al hardware.
- Frecuentemente, la porción residente del propio sistema operativo se denomina núcleo del sistema (**kernel**).

¿Por qué estudiar sistemas operativos?

- Podremos aprender como trabajan realmente las computadoras (junto con lo que vieron arquitectura)
- Hay sistemas operativos en todos los sistemas informáticos (casi) por lo que seguramente van a tener que interactuar con alguno y conocer sus limitaciones
- Los sistemas operativos son ejemplos de soluciones maduras a problemas difíciles de ingeniería.

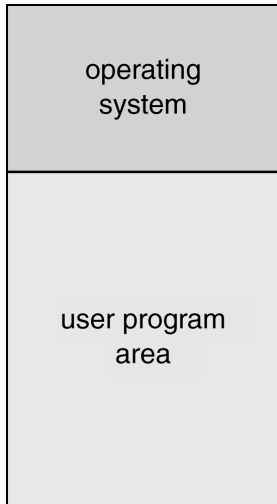
¿Por qué estudiar sistemas operativos?

- El diseño de los sistemas operativos muestra la importancia de establecer un diseño cuidadoso antes de empezar a programar
- Los sistemas operativos son programas de misión crítica con muchas líneas de código que deben funcionar eficientemente.
- La naturaleza concurrente de los sistemas operativos modernos nos permite motivar la necesidad de aprender programación concurrente

Evolución histórica de los sistemas operativos

Perspectiva histórica: Sistemas batch ('70)

- En las primeras épocas los sistemas eran grandes y costosos.
- Constaban de un entrada de trabajos o cola y una salida impresa.
- Prácticamente no había interacción alguna con el usuario.
- La función principal del SO era la del cargador (loader) de programas y soporte de entrada/salida (E/S) a dispositivos (operaciones comunes).
- El sistema soportaba un único trabajo a la vez.



Perspectiva histórica: Sistemas batch ('70)

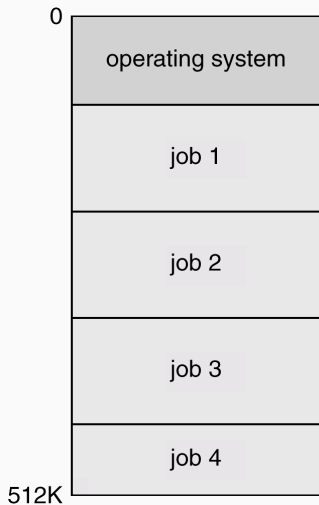
- Lectora de Entrada \Rightarrow Proceso \Rightarrow Salida y resultado
- Las tareas relacionadas, se agrupaban en conjuntos de trabajos o lotes (**batches**) para su procesamiento más eficiente.
- Con el advenimiento de los discos se comenzó a realizar el **spool** de los dispositivos haciendo más rápidas las operaciones e introduciendo por primera vez el solapamiento o concurrencia de operaciones.
- spool = Simultaneous Peripheral Operations On-Line
- El **spooler** es un buffer (una región de memoria) donde se guardan datos a la espera de su procesamiento por un dispositivo más lento
- El recurso más caro de la época era el procesador (CPU), que tenía un bajo porcentaje de utilización.

Perspectiva histórica: Batch multiprogramado ('80)

- Fue una mejora a los sistemas batch en los comienzos de la década del 80
- El disponer de un conjunto de trabajos (**pool de jobs**) en memoria secundaria permitió desarrollar técnicas de planificación de despacho (**job scheduling**) así como de multiprogramación.
- El sistema debía seleccionar un subconjunto de trabajos o lotes (**jobs**) que estaban en memoria secundaria para cargar en memoria principal.
- El sistema operativo seleccionaba un trabajo para ejecutar. Cuando el trabajo seleccionado debía esperar por alguna tarea (p. ej.: ejecución de una E/S), el sistema elegía otro para utilizar el procesador.

Perspectiva histórica: Batch multiprogramado ('80)

- Todo esto implicó el desarrollo de técnicas incipientes para el manejo de la memoria, ya que había que compartirla entre todos los trabajos.
- La multiprogramación incrementa la utilización del recurso procesador.



Perspectiva histórica: Sistemas de tiempo compartido ('80)

- Los sistemas batch multiprogramados no tenían interacción con el usuario además de un tiempo de retorno (**turnaround time**) extenso.
- El **debug** de un programa seguía siendo tortuoso (**dump** de la memoria).
- Sistemas de tiempo compartido, ejecutan programas en forma concurrente con una elevada tasa de despacho de procesador (**context switch**) de forma tal de permitir que los usuarios interactúen directamente con el sistema como si fueran su único usuario.

Perspectiva histórica: Sistemas de tiempo compartido ('80)

- Se debe combinar multiprogramación con técnicas de planificación de CPU (**scheduling**) para proveer a cada usuario con una porción adecuada del sistema.
- Los sistemas de tiempo compartido (**time sharing systems**) son una extensión lógica de los sistemas multiprogramados.
- Los usuarios utilizaban terminales para implementar la interacción y eran atendidos por un intérprete de comandos (**multiusuarios**).
- La interacción era un resultado de la transmisión carácter a carácter.

Perspectiva histórica: Sistemas de tiempo compartido ('80)

- Todos los usuarios creían tener el computador a su disposición.
- Si bien un procesador ejecuta un único proceso por vez, el despacho del mismo 30 ó 40 veces por segundo entre diferentes tareas, le brinda a los usuarios la sensación de que está para su uso exclusivo.
- Este intercambio es tan frecuente que el usuario puede interactuar con su trabajo con total comodidad.
- La necesidad de acceder y actualizar datos en forma concurrente, creó la necesidad de evolucionar el sistema de archivos a uno multiusuario, incorporando técnicas de protección y serialización del acceso.

Perspectiva histórica: Sistemas de tiempo compartido ('80)

- También apareció el problema de que los usuarios podrían ver la memoria de los procesos de otros usuarios
- En estas condiciones, los procesos deben contar con la memoria y procesador necesarias para su ejecución eficiente dando forma a los requerimientos de los sistemas operativos de hoy día.
- Comenzó a aparecer la idea de la memoria virtual
 - Los procesos ven un espacio de memoria virtual que el sistema operativo se encarga de mapear a la memoria física
 - Permite que varios procesos corran juntos en memoria sin requerir modificaciones

Perspectiva histórica: Computadoras personales ('80)

- Con costos de hardware decrecientes fue posible el diseño y uso de computadores personales.
- El sistema era diseñado en base a que sería dedicado a un único usuario.
- Con en un principio modestos recursos de procesador, el énfasis y desarrollo estuvo por mejorar la interfase con el usuario.
- Para ello el sistema operativo debió maximizar la habilidad de interacción con el usuario en vez de uso de CPU, etc.
- La interfase de comandos habitual y diseñada para técnicos fue sustituida por la interfaz de ventanas que hoy conocemos.

Perspectiva histórica: Computadoras personales ('80)

- Se introdujeron nuevos dispositivos que mejoran la interacción con el usuario (audio, ratón, video, micrófono, cámara, disquete, etc.).
- Finalmente, los PC invadieron el ambiente empresarial al ser dispuestos en red. Para ello, utilizando sistemas homogéneos y servidores con habilidades específicas de impresión, base de datos, sistema de archivo, seguridad, correo, etc.
- Esta es la disposición reciente dónde las aplicaciones se implementan en modalidad **cliente-servidor**.

Perspectiva histórica: Sistemas paralelos ('90)

- Desde comienzos de los 90, acompañando el desarrollo del hardware
- Sistemas donde se dispone de más de un procesador permiten la ejecución simultánea y sincronizada de más de un proceso.
- Se clasifican en:
 - Sistemas altamente integrados (**tightly coupled**). Son sistemas en donde los canales de interconexión son de alta velocidad (bus común o memoria compartida).
 - Sistemas poco integrados (**loosely coupled**). Sistemas en donde los canales de interconexión son de baja velocidad relativa. Sistemas en red.

Perspectiva histórica: Sistemas paralelos ('90)

- Taxonomía de Flynn (1972):
 - Las cuatro clasificaciones definidas por Flynn se basan en el número de instrucciones concurrentes (control) y en los flujos de datos disponibles en la arquitectura
 - **SISD** (Single Instruction, Single Data): Una arquitectura secuencial donde no existe paralelismo. (Sistemas monoprocesadores).
 - **SIMD** (Single Instruction, Multiple Data): Sistemas que ejecutan la misma instrucción sobre un conjunto distinto de datos (Sistemas vectoriales).
 - **MISD** (Multiple Instruction, Single Data): Utilizado para paralelismo redundante.
 - **MIMD** (Multiple Instruction, Multiple Data): Sistemas con procesadores autónomos que ejecutan en forma simultánea diferentes instrucciones sobre diferentes datos.

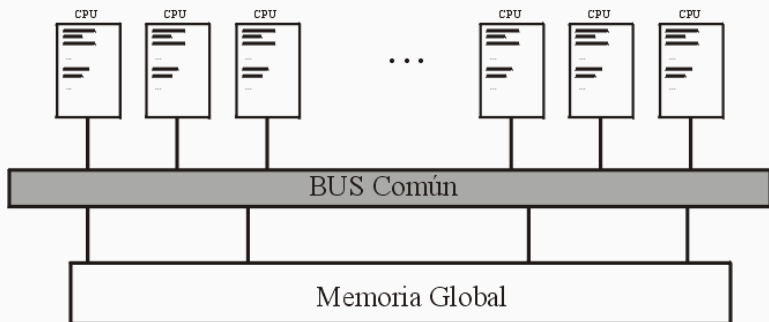
Perspectiva histórica: Sistemas paralelos ('90)

Dentro de los sistemas **MIMD** se suelen hacer dos divisiones:

- **Sistemas de memoria compartida:** Son sistemas donde los procesadores coexisten en un único nodo o en un sistema altamente integrado. Escalan poco debido a que el acceso a memoria desde los procesadores se convierte en un “cuello de botella”.
- **Sistemas de memoria distribuida:** Son sistemas donde los nodos son independientes que se interconectan a través de una red de alta velocidad. Escalan a miles de procesadores.

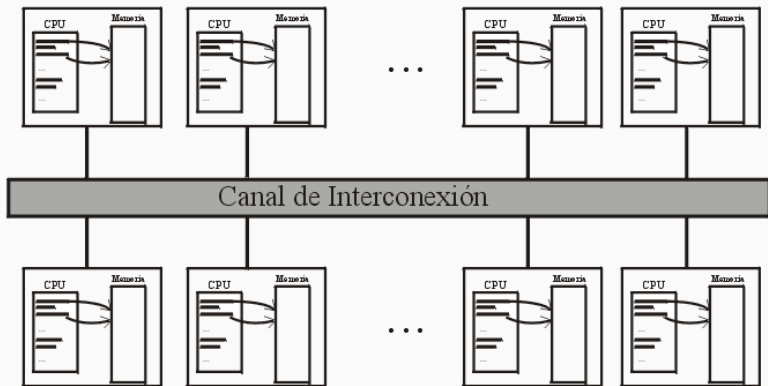
Perspectiva histórica: Sistemas paralelos ('90)

Arquitecturas de memoria compartida



Perspectiva histórica: Sistemas paralelos ('90)

Arquitecturas de memoria distribuida



Sistemas asimétricos:

- Al surgir los sistemas multiprocesadores los núcleos de los sistemas operativos se modificaron para soportar este tipo de sistema. La forma más sencilla fue asignar la ejecución de código del núcleo a un único procesador.
- De esta forma, los sistemas operativos no tenían que lidiar con la programación concurrente, ya que su código estaba restringido a ejecutarse en un único procesador.
- Posteriormente, se empezó a asignar ciertas tareas a otros procesadores, generando una jerarquía entre ellos.

Sistemas simétricos:

- Al avanzar el diseño de los sistemas operativos se desarrollaron sistemas en donde el código del núcleo se dispone en la memoria común y puede ser ejecutado por cualquier procesador. Se pierde la jerarquía de los sistemas asimétricos y todos los procesadores pasan a ser simétricos.
- Pasó a ser una necesidad crítica que el núcleo sea reentrante. Los proveedores debieron rediseñar totalmente sus sistemas.
- El código del sistema operativo, al igual que el ancho de banda de la memoria, se transforman en recursos críticos, que determinan la escalabilidad del sistema.

Dentro de los sistemas multiprocesadores se los caracteriza en dos tipos:

- **Sistemas UMA** (Uniform Memory Access):
 - Acceso uniforme a la memoria. Cada procesador accede a cualquier lugar de memoria con el mismo costo en cuanto al tiempo.
- **Sistemas NUMA** (Non-Uniform Memory Access):
 - Acceso no uniforme a la memoria. Los procesadores tienen conjuntos de memoria a la cual acceden más rápido que el resto.

Perspectiva histórica: Sistemas de tiempo real ('00)

- En sistemas de este tipo estricto, todo resultado debe producirse en un cierto tiempo, o de lo contrario el sistema falla.
- En la práctica, un sistema de tiempo compartido con prioridades dinámicas y despacho preemptivo en general puede ser utilizado en estas condiciones.
- Dos tipos:
 - **Hard**
 - Todas las demoras del sistema deben estar acotadas.
 - En general no se usa almacenamiento secundario en disco.
 - Sistemas especializados.
 - **Soft**
 - Sistemas de propósito general con procesos de tiempo real con mayor prioridad.
 - No son tan estrictos como los otros pero pueden combinar otros procesos de menor prioridad.

- Sistemas multimedia
 - Sistemas especializados en la incorporación de datos multimedia (audio y video).
 - Estos tipos de datos deben reproducirse bajo ciertas restricciones de tiempo a los usuarios.
- Sistemas virtuales
 - Sistemas que corren como aplicaciones de otros sistemas operativos.
 - Permiten mover un sistema de un hardware a otro sin detenerlo.

- Sistemas de mano
 - Estos utilizan sistemas operativos embebidos que tienen limitaciones de recursos y altos requerimientos de prestaciones.
 - Actualmente se parecen cada vez más a computadores personales y usan sistemas operativos similares.
- Sistemas en la nube
 - Toda la infraestructura del sistema se encuentra en un centro de datos remoto
 - Usan virtualización como forma de implementación