Sistemas Operativos
Tema 1: conceptos generales

Contenidos

- ¿ Qué es un SO ?
- Evolución histórica de los SO
- Tipos de sistemas informáticos

Elementos de un sistema informático

- Hardware (lo tangible)
- Software (programas, lo intangible)
  - software del sistema
  - aplicaciones
- Personas (usuarios del sistema)
  - usuarios
  - programadores

El SO **controla y coordina** el uso del *hardware* entre los distintos *programas* para diversos *usuarios*

¿Qué es un sistema operativo?

- Un programa que sirve de intermediario entre los usuarios y el hardware
- Pertenece al software del sistema
- Objetivos:
  - Ejecutar las aplicaciones de los usuarios
  - Administrar eficientemente los recursos de la máquina → eficiencia
  - Facilitar la interacción con el computador → usabilidad

Definiciones breves

- Un **sistema** de software cuyo fin es que un sistema informático sea **operativo** (utilizable).
- Conjunto de programas que gestionan los recursos del sistema, optimizan su uso y resuelven conflictos.

¿Qué es un sistema operativo?

- Es un **administrador de recursos**
  - como si fuera un gobierno del hardware
  - programa de control
  - ojo, ***no realiza trabajo productivo***
- Es una **interfaz** con el hardware
  - añade características no existentes en el hw
  - oculta características inconvenientes del hw
  - **máquina extendida**

El SO proporciona un **ambiente** de ejecución de programas

En caso de conflictos debe decidir de forma **eficiente** y **justa**

El SO como administrador de recursos

- Tenemos dos participantes en el sistema: los **procesos** y los **recursos**.
  - Un proceso es un programa en ejecución
  - Un recurso puede ser real o virtual, físico o lógico
- Los procesos **compiten** por el uso de recursos escasos.
- Necesitamos un árbitro imparcial que asigne recursos a los procesos, de forma **justa** y **eficiente**.

El SO como administrador de recursos (2)

- El SO debe determinar a quién se le entregan los recursos, qué cantidad de recursos se conceden, en qué momento y durante cuánto tiempo.

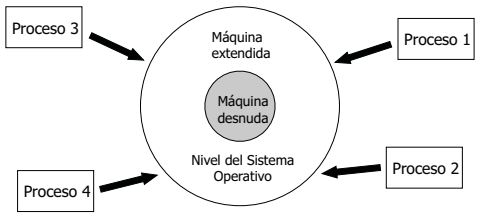
  → **políticas de gestión de recursos**

El SO como administrador de recursos (y 3)

- Criterios de gestión de recursos:
  - Optimizar el rendimiento del sistema
  - Reparto justo → evitar acaparamientos e **inanición** de procesos perjudicados
  - Garantizar la seguridad e integridad de la información
  - …
- Normalmente, los distintos criterios entran en conflicto
  - Ej. no se puede maximizar el rendimiento y a la vez dar un reparto justo

El SO como interfaz

- Es una capa entre el usuario y el hardware.
- La interfaz ofrece una máquina extendida que es una **abstracción** de la realidad.



El SO como interfaz (2)

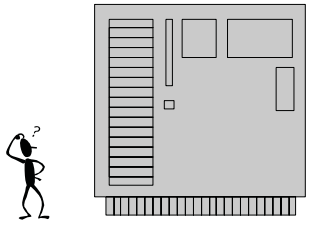
- Esa abstracción es más cómoda, más **conveniente**, más usable para el usuario y para el programador.
- Esta interfaz puede ser independiente del hardware: ganamos **portabilidad**.

Recorrido histórico: tipos de Sistemas

- Primeros sistemas
- Sistemas por lotes
- Mejoras en la gestión de la E/S
- Sistemas por lotes multiprogramados
- Sistemas de tiempo compartido
- Ordenadores personales
- Sistemas paralelos:multiprocesadores
- Sistemas distribuidos
- Sistemas de tiempo real

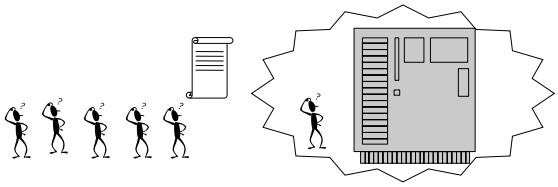
Primeros sistemas

- Los primeros sistemas de computación se caracterizaban
  - Gran tamaño
  - Prácticamente sin ningún soporte lógico (tableros enchufables, tarjetas perforadas,...)



Primeros sistemas

- Organización del trabajo:
  - usuario experto: operador/programador
  - un solo usuario en cada momento (tiempo asignado, “listas de reserva”)



Primeras mejoras

- Dispositivos físicos
  - Lectoras de tarjetas, impresoras y cintas magnéticas
- Elementos lógicos: aparece el primer software de sistema,
  - ensambladores, compiladores, cargadores
  - manejadores de dispositivos
  - bibliotecas con subrutinas de uso frecuente
  - Finalmente aparecieron los primeros compiladores de lenguajes de alto nivel (FORTRAN, COBOL), simplificando la labor de programación pero aumentando la carga de trabajo del computador

Problemas

- El modo de trabajo hacía que la máquina estuviera parada mucho tiempo:
  - tiempo de **puesta a punto** (*setup time*)
  - tiempo de corrección manual de errores
  - tiempo sobrante por finalización temprana (raro)
- Equipos muy caros

Sistemas por lotes

- Objetivo: sacar más provecho a la máquina gracias a una mejor organización del trabajo.

Los operadores agrupaban los trabajos por **lotes**, que eran trabajos con necesidades similares y que eran ejecutados en la computadora como un *grupo de tareas*. A medida que la computadora quedaba libre, se ejecutaba un lote.

Sistemas por lotes

- Primer paso: aparición del **operador** especialista.
  - El programador no manipula directamente el equipo.
  - El programador entrega su **trabajo** (**job**) al operador.
  - El operador somete la tarea al sistema y entrega los resultados al programador.
  - El programador corrige sus errores mientras el operador sigue ejecutando otras tareas.
- Resultado: aumento de la productividad.

Sistemas por lotes (2)

- Segundo paso: agrupar las tareas en **lotes** que se procesan de forma automática
  - **Procesamiento por lotes** (***batch processing***)
- El operador puede preparar lotes con trabajos que requieren una misma operación (ej. cargar el compilador)
- El operador lanza el lote, y éste se ejecuta sin más intervención (*secuencia automática de trabajos*)

Sistemas por lotes (y 3)

- Necesario automatizar ciertas acciones comunes
  - Control de la finalización de tareas
  - Tratamiento de errores
  - Carga y ejecución automática de la siguiente tarea
- En lugar de dar órdenes al operador, ¿Porqué no dárselas directamente al computador?

El primer sistema operativo

- Es necesario que el computador tenga un pequeño **monitor residente** (*controlador*) que realice automáticamente las acciones anteriores. ¿Cómo?
- Distinguiendo entre:
  - Tarjetas de instrucciones de programas de usuarios
  - Tarjetas de control (Primer lenguaje de control de sistema: $FTN, $ASM, $RUN, $JOB, $END)

Elementos de un sistema por lotes

- Lenguaje de control de tareas (JCL, job control language)
  - el lote se escribe usando un JCL
  - define qué programas hay que cargar, qué datos leer, etc.
  - se escribe en tarjetas perforadas, cinta, etc.
- Monitor residente
  - programa fijo en memoria con rutinas imprescindibles para que el sistema por lotes funcione: intérprete del JCL, cargador de programas, rutinas de E/S...
  - automatiza tareas del antiguo operador
  - es el primer sistema operativo auténtico

El monitor residente: protección

- **Se empiezan a considerar aspectos de protección:**
  - proteger la memoria ocupada por el monitor residente
  - impedir accesos directos a la E/S
  - evitar que una tarea deje bloqueado al sistema
- **Todo ello requiere cierto apoyo del hardware**

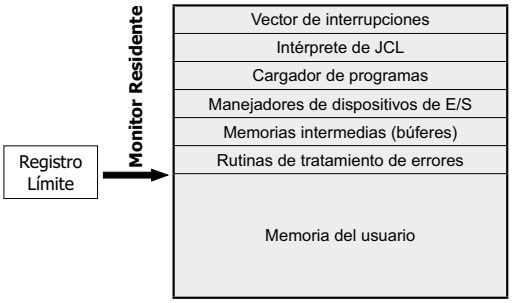
Cuestiones

- ¿ Quién carga el intérprete del lenguaje de control ?
- ¿ Cómo se debe actuar en caso de fallo del programa de usuario ?
- ¿ Cómo garantizar el uso correcto de los dispositivos de E/S ?
- ¿ Cómo los programas realizan las operaciones de E/S ?

Cuestiones (2)

- ¿ Cómo podemos diferenciar si las instrucciones de E/S son utilizadas por el usuario o por el monitor residente ?
- ¿ Cómo proteger al monitor residente ?
- ¿ Cómo garantizar el control del sistema ?

La memoria en un sistema con monitor residente



SO y arquitectura del computador

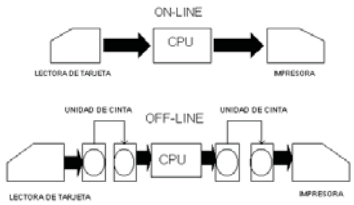
- Los SO y la arquitectura de los computadores se han influido mucho entre ellos dos
- Los SO se crearon para facilitar el uso del hardware
- A medida que se diseñaron y usaron los SO, se hizo evidente que podrían simplificarse si se modificaba el diseño del hardware
- A lo largo de la evolución de los SO se observa que los problemas de los SO han dado pie a la introducción de nuevas características del hardware

El problema de la E/S

- La E/S era muy lenta en comparación con la CPU.
- Esto provocaba que la CPU quedara ociosa mucho tiempo esperando por la terminación de operaciones de E/S.
- Algunas técnicas para tratar el problema:
  - **Operación fuera de línea (*offline*)**
  - **Uso de búferes**
  - **Spooling**

Operación fuera de línea (off-line)

- El computador central dialoga directamente sólo con dispositivos rápidos (cintas magnéticas).
- Un pequeño computador (*satélite o canal*) se encarga de las transferencias con dispositivos lentos (tarjetas, impresora).



Operación fuera de línea (off-line)

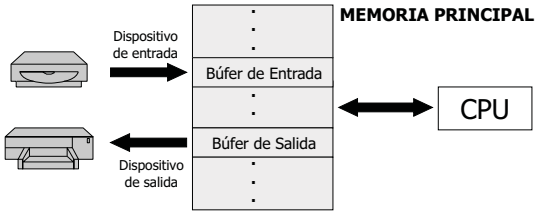
- Resultado:
  - mejor aprovechamiento del procesador central
  - ejecución paralela de cálculos y operaciones de E/S
- Se puede incrementar la velocidad utilizando varios satélites.

Operación fuera de línea (off-line)

- Para trabajar con *offline*, no hace falta recompilar los programas antiguos.
- Los trabajos siguen usando los mismos servicios para la E/S. Lo que cambia es su implementación en el S.O: **independencia del dispositivo**.

Búferes

- Esquema de operación de E/S en el que las transferencias de E/S se realizan a través de un área intermedia de memoria (búfer)
- La operación de E/S se realiza sólo cuando el dispositivo está preparado.



Búferes

- La CPU sólo espera por E/S cuando el búfer está vacío (entrada) o lleno (salida)
- El uso de búferes permite solapar operaciones de E/S de una tarea con instrucciones de CPU de esa misma tarea

Búferes

- El uso de búferes no resuelve totalmente el problema de la lentitud de los equipos de E/S
- Los búferes sólo sirven para amortiguar picos de alta actividad de E/S.
- Su eficacia depende fundamentalmente de la velocidad de los equipos de E/S y del tipo de tareas en ejecución
  - Si la E/S es muy lenta, los búferes de entrada se vacían y los de salida se congestionan.
  - Las tareas con muchos requerimientos de E/S (I/O-bound jobs/CPU-bound jobs) provocaran el mismo efecto

Spooling
(Simultaneous Peripheral Operation On-Line)

- Este esquema de funcionamiento de operación de la E/S surge gracias a la aparición de los discos
- Se utiliza el disco como un enorme búfer.
- El proceso lee/escribe sobre el disco, en lugar del dispositivo de E/S.
- Mientras se ejecuta un trabajo, el S.O.:
  - lee los datos del siguiente trabajo de la cinta/tarjetas al disco
  - imprime la salida del anterior trabajo, del disco a la impresora

Spooling
(Simultaneous Peripheral Operation On-Line)

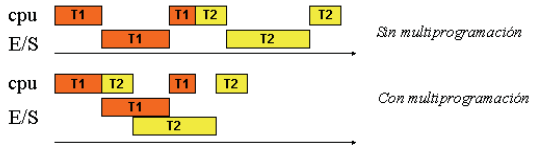
- Permite solapar la E/S de un proceso ya terminado con las operaciones en CPU de otro proceso.
- Introduce una estructura de control, Fondo de trabajos (job pool): el lote está en el disco, así que el S.O. puede elegir el trabajo más adecuado.

Multiprogramación

- El modo de operación *offline* el el *spooling* aumentan el rendimiento de la CPU pero tienen sus limitaciones
- En algún momento la CPU quedará ociosa esperando por alguna operación de E/S
- Al existir la posibilidad de tener varios trabajos en un dispositivo de acceso directo, como un disco, es posible la planificación de trabajos
  - El SO puede escoger qué trabajo ejecutara a continuación

Multiprogramación

- Cuando un proceso se bloquea al esperar por la E/S, ejecutamos en la CPU instrucciones de otro proceso.
- Los procesos entrelazan su ejecución: concurrencia.
- La CPU y la E/S trabajan a la misma vez ⇒ se terminan más trabajos en menos tiempo



Multiprogramación

- Los sistemas multiprogramados son más complejos:
  - Cuando la CPU queda libre, ¿a qué proceso elegimos? (**planificación de la CPU**)
  - conflictos por acceso simultáneo a la E/S (**planificación de dispositivos**)
  - varios procesos a la vez en memoria **(gestión de memoria)**
  - **Protección**
  - **Solución a situaciones de interbloqueo**

Tiempo compartido (time sharing)

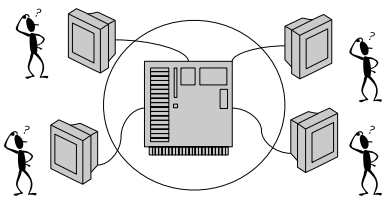
- Multiprogramación + interactividad = tiempo compartido
- Los sistemas por lotes no son interactivos (el usuario no interviene durante la ejecución de su trabajo)
  - Con la multiprogramación e interactividad, se replantea su caracterización: antes era por el agrupamiento de tareas similares mientras que ahora se caracterizan por la falta de interacción entre los usuarios y las tareas en ejecución

Tiempo compartido (time sharing)

- Idea: la CPU reparte su tiempo entre los distintos procesos.
- Cada proceso dispone de una rodaja de tiempo periódica. Si el periodo es lo bastante pequeño, el usuario no lo percibe.
- Con el t.c. se pierde productividad de CPU, pero se gana en productividad humana

Tiempo compartido (time sharing)

- Por tanto con el tc se consigue:
  - **tiempos de respuesta** cortos
  - Los usuarios tienen la impresión de poseer un ordenador particular



Ordenadores personales

- La aparición del microprocesador permitió fabricar computadores baratos, asequibles para el consumo de masas => ordenadores personales (años 80)
- Destinados al uso individual y no experto.
- Máxima importancia a la facilidad de uso, bajos tiempos de respuesta, etc.
- Interfaces de usuario: sistemas WIMP (windows, icons, menus, pointers).
- Utilizan tecnología de los grandes S.O., pero prescinden de ciertos servicios (protección, multiprogramación, etc.)

Sistemas paralelos - multiprocesadores

- Sistemas con más de un procesador. Pueden ejecutar varias instrucciones simultáneamente (en paralelo).
- Sistemas estrechamente acoplados: los procesadores comparten una memoria común.
- Sólo hasta decenas o centenares de procesadores.
- Ventajas:
  - aumento de velocidad de procesamiento con bajo coste
  - cierta tolerancia a fallos
- Inconvenientes:
  - necesidad de sincronización entre procesos

Sistemas distribuidos

- Múltiples procesadores conectados mediante una red.
- Sistemas débilmente acoplados: los procesadores no comparten memoria ni reloj.
- Escalable hasta millones de procesadores (ej. Internet)

Sistemas distribuidos (2)

- Ventajas:
  - compartición de recursos dispersos
  - ayuda al trabajo cooperativo de equipos humanos
  - aumento de velocidad
  - fiabilidad (tolerancia a fallos, alta disponibilidad)
- Complicaciones:
  - no comparten memoria: la comunicación es más compleja y no se puede tener un estado global visible por todos los nodos al instante.
  - red de comunicaciones no fiable
  - heterogeneidad de los nodos

Sistemas de tiempo real

- Para poder ejecutar satisfactoriamente tareas que han de completarse en un plazo prefijado (ej. sistemas de control industrial, sistemas multimedia)
- Dos tipos:
  - **s.t.r. crítico**: para tareas que siempre deben cumplir los plazos de terminación. Adecuados para la industria. Muy simples, incompatibles con tiempo compartido, memoria virtual, etc.
  - **s.t.r. no crítico**: intentan cumplir los plazos, pero no los garantizan al 100%. Adecuados para multimedia, etc.

Otros sistemas

- Sistemas empotrados (*embedded systems*)
- Sistemas de propósito específico
  - Teléfonos móviles
  - Consolas de videojuegos
  - …