

## **2. Evolución y tendencias de los Sistemas Operativos**

Una rápida presentación de la evolución de los sistemas operativos permite entender mejor el por qué de sus características actuales e introducir algunos términos de uso corriente.

En los primeros ordenadores, no existía sistema operativo (Fase 0). Cada usuario reservaba la máquina durante un tiempo determinado y disponía entonces de todos sus recursos, mediante la codificación o introducción de todas las instrucciones en lenguaje máquina. Tras escribir el programa, lo cargaba manualmente mediante interruptores, cintas de papel o tarjetas perforadas. El seguimiento se realizaba a través de la consola mediante una serie de indicadores luminosos, y en caso de error, se podía detener el programa y examinar el contenido de memoria; es decir, la interacción directa (ejecución paso a paso, modificación directa de la memoria) era la principal herramienta de puesta a punto de los programas.

### ***2.1 Primera Fase (Fase 1)***

El manejo del ordenador se hacía muy trabajoso y se producían muchos errores. En un primer paso, se intentan resolver las diferencias entre el lenguaje máquina y el humano. La solución vino dada por el desarrollo de componentes; componentes, que surgieron de forma independiente y desorganizada.

Así, los primeros componentes o sistemas de 'software' básico que se desarrollaron fueron, por una parte, los de ayuda al desarrollo de los programas (ensambladores, compiladores, ayudas para la puesta a punto), y por otra parte, los subprogramas de entrada/salida. Estos no eran más que unas subrutinas especiales ('device/driver') que se introducían en memoria junto con los programas de usuario y que resolvían los problemas de entrada/salida: sincronizando la E/S de datos con el trabajo de la UCP, conmutando automáticamente las IOAREAS<sup>1</sup>, detectando algunos errores.

Es decir, gestionaban el uso de 'buffers', 'flags', registros, bits de control, bits de estado, etc...

Otros componentes desarrollados en este periodo, fueron los programas de ayuda: generadores de listados, cargadores, volcadores de memoria, combinadores de programas, etc.

En esta fase, también hay que destacar la creación de la asociación de usuarios SHARE para el intercambio de programas, y el nacimiento de la idea de las bibliotecas de programas.

## 2.2 Segunda Fase (Fase 2)

El modo operativo existente, llamado de puerta abierta, era poco económico, ya que se utilizaba mal un material muy costoso. La segunda fase viene marcada por el objetivo de aumentar el rendimiento reduciendo el tiempo de preparación o 'set-up time'.

Como primer paso para conseguir dicho objetivo, se instauró la figura del operador; profesional conocedor del sistema que para aumentar el rendimiento, agrupaba para su ejecución los trabajos (jobs) por similitud de necesidades.

Sin embargo, el tiempo de preparación de los trabajos seguía siendo demasiado alto comparado con el tiempo de proceso (run time). Para reducir el tiempo de preparación de los trabajos se necesitaba un programa de control, que hiciese de operador automático, y que asegurase el proceso continuado de trabajos. Este programa permitiría la ejecución en secuencia de una serie de trabajos (programas y juegos de datos) preparados de antemano, automatizando el paso de un trabajo al siguiente. Para poder comunicarse con él se creó un lenguaje de control (Operating System Control Lenguaje u O.S.C.L.).

Hacia finales de los años 50, empezaron a aparecer los primeros sistemas operativos que se llamaron **monitores de encadenamiento o residentes**, que reunían en torno a dicho programa el software básico existente.

La principal función de tales sistemas (Figura 1.2.) era la gestión de recursos: memoria, procesador, entradas/salidas. La automatización de esta gestión implica la protección del sistema contra posibles errores:

- La limitación del tiempo de ocupación del procesador, para evitar que un bucle sin fin en un programa bloquee todo el sistema.
- La supervisión de entradas/salidas para evitar, sobre todo, bucles en la utilización de los periféricos.

---

1 Ioarea: zona de almacenamiento temporal para el flujo de datos de E/S.

- La protección de la zona de memoria reservada al monitor, para evitar su modificación como consecuencia de un error de direccionamiento en un programa de usuario.

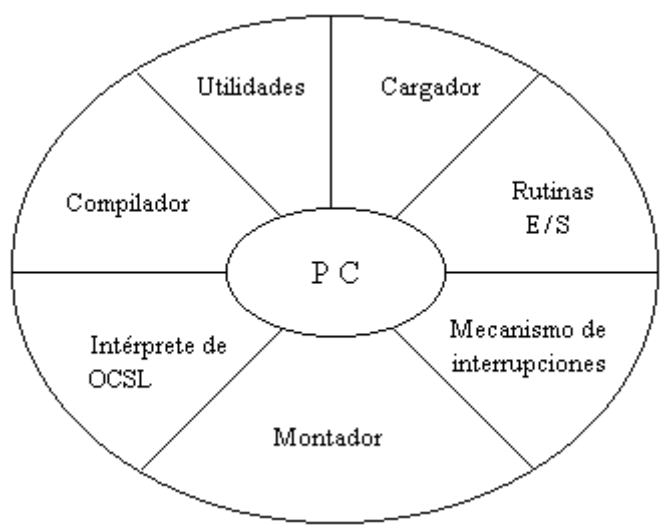


Figura 1.2: Monitor de encadenamiento.

La necesidad de asegurar estas funciones es el origen de diversas ampliaciones del hardware: gestión de un reloj, restricciones del empleo de determinadas instrucciones y protección de la memoria.

El uso de monitores de encadenamiento mejoró notablemente el rendimiento en la utilización del procesador. Este rendimiento, sin embargo, quedaba restringido por el hecho de que el procesador estaba completamente ocupado a lo largo de toda la ejecución de los programas, incluidas las operaciones de entrada/salida, y la velocidad posible para estas últimas estaba limitada por la de los dispositivos mecánicos de E/S que es intrínsecamente menor que la de los dispositivos electrónicos (Figura 1.3.).

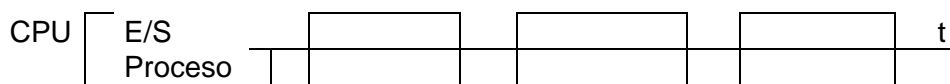


Figura 1.3. Gráfico de utilización del ordenador

### 2.3 Tercera Fase

Con el objeto de reducir el tiempo empleado en las operaciones de E/S se ideó una solución que consistió en utilizar dos ordenadores (**Fase 3a**). Los programas se ejecutaban en el ordenador principal y las entradas/salidas se hacían en cinta magnética con un flujo elevado de transferencia. Un ordenador auxiliar, en 'off-line', se

encargaba de constituir las cintas de entrada a partir de tarjetas perforadas y de listar en impresora el contenido de las cintas de salida: es lo que se denomina procesamiento satélite. Una planificación adecuada de la entrega de trabajos permitía utilizar los dos ordenadores en paralelo y por consiguiente explotar mejor las capacidades de tratamiento del ordenador principal. En contrapartida, había una cierta rigidez en la explotación (problemas de horarios para la entrada de trabajos, tiempo de respuesta elevado), perjudicial para el bienestar de los usuarios. Estos sistemas de tratamiento por lotes (batch processing systems) estaban bastante extendidos a principios de los 60. En ellos, los programas trabajaban con dispositivos lógicos y el sistema operativo establecía la correspondencia con los dispositivos físicos interpretando los comandos de OSCL.

Entre 1960 y 1970, aprovechando los importantes progresos que se dan, tanto en la tecnología 'hardware' como en la concepción de los sistemas; se superaron en gran medida las limitaciones citadas y se logra reducir al mínimo los tiempos muertos de CPU. Los pasos que se dieron para ello, fueron los siguientes:

### 2.3.1 DESARROLLO DE LOS CANALES

Se desarrollaron unos procesadores autónomos especializados en la transferencia de información (unidades de intercambio o **canales**) que permitían descargar el procesador central de la gestión detallada de las entradas/salidas.

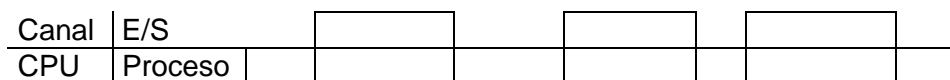


Figura 1.4. Reparto de operaciones entre el Canal y la UCP

Un canal, o unidad de intercambio, es un procesador capaz de ejecutar entradas/salidas de manera autónoma, paralelamente al tratamiento propiamente dicho en la CPU. Por consiguiente, la unidad central y los canales tienen acceso a informaciones comunes en memoria central y la velocidad relativa del tratamiento y de la transferencia se convierte en un factor importante.

### 2.3.2 BUFFERING O ALMACENAMIENTO INTERMEDIO

El objetivo, es solapar la entrada/salida de datos de un trabajo con su propio cómputo. La capacidad de tratamiento de los dispositivos periféricos, tales como la lectora de tarjetas o la impresora de líneas, tributarios estos de elementos mecánicos,

es muy débil en relación con la capacidad de tratamiento de la unidad central (Figura 1.5.).

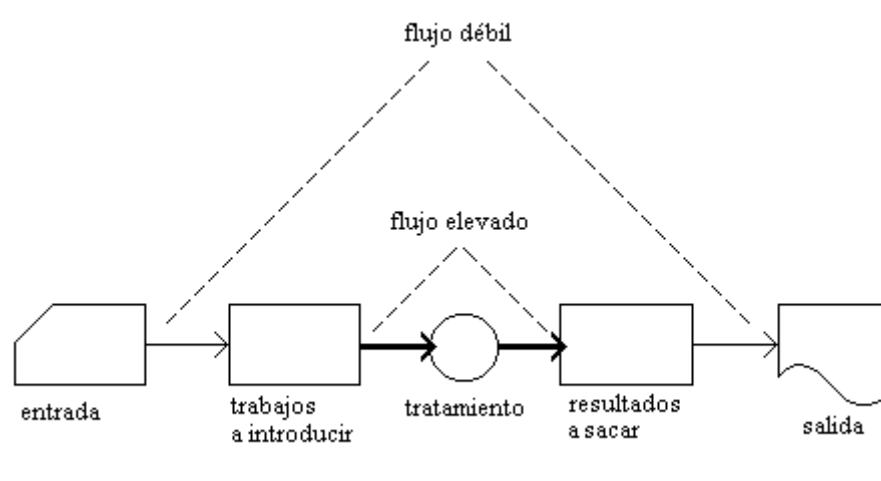


Figura 1.5. Flujo de datos

La duración media del tratamiento en la unidad central, del contenido de una tarjeta es muy inferior (en un factor de 1.000 o 10.000) a la duración de su lectura; asimismo, la unidad central puede producir numerosas líneas de resultados mientras se realiza la impresión de una línea. De esto se derivan dos consecuencias relativas a la organización de un sistema operativo:

- es necesario leer numerosos datos de antemano para garantizar una tasa de actividad alta en la unidad central;
- es necesario conservar en memoria un gran número de resultados a la espera de impresión.

Es decir, debe acumularse la información en memoria, en 'buffers' de entrada o de salida.

Sin embargo, no se suele cumplir el objetivo marcado; la CPU o los dispositivos quedan en espera. Por otro lado, el manejo de 'buffers' es función del sistema operativo por lo que su programación se hace más compleja.

### 2.3.3 TÉCNICA DE SPOOLING

Para evitar el bloqueo de la memoria central por los buffers de entrada/salida, se guardan los datos en una memoria secundaria de gran capacidad, haciendo que el flujo de transferencia entre memoria central y memoria secundaria sea elevado en relación con el flujo con los dispositivos de entrada/salida lentos. Esto fué posible con el desarrollo de los dispositivos de almacenamiento directo (DAAD) ya que las cintas

magnéticas son por naturaleza secuenciales y en consecuencia no se pueden leer diferentes partes de ella simultáneamente.

Así, utilizando un DAAD como un gran 'buffer' de entrada/salida se logra el solapamiento de la entrada/salida de datos, código, etc... de unos trabajos con el procesamiento de otro. Además, puesto que el DAAD de 'spool' es un depósito de trabajos, el sistema podría elegir a cual dar paso, es decir podría realizar una planificación de trabajos.

El esquema de la figura 1.6 pone en evidencia dos puntos importantes en estos sistemas: el papel central de la memoria principal (los sistemas primitivos estaban organizados alrededor del procesador), y la importancia del flujo de información entre memoria principal y memoria secundaria. Estas transferencias entre los dos niveles de memoria están controladas por el sistema operativo. En consecuencia, el tamaño de la memoria principal y el flujo de transferencia entre los dos niveles de memoria se convierten en parámetros determinantes para el rendimiento del sistema.

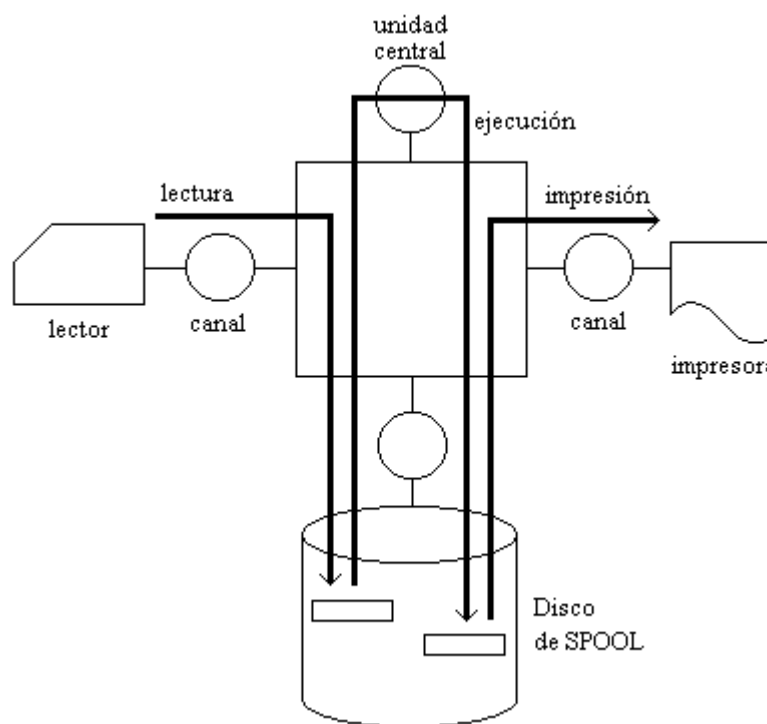


Figura 1.6. Flujo de información en un sistema con Spooling

Para facilitar el uso eficiente de los canales y de los nacientes mecanismos de sincronización e interrupción, así como la gestión de varios tipos de memoria se ampliaron los monitores residentes o de encadenamiento pasándose a los **ejecutores o ejecutivos**.

#### 2.3.4 MULTIPROGRAMACIÓN (FASE 3B-I)

En un sistema operativo con entradas/salidas simultáneas, la memoria reservada al usuario está dividida en una zona de buffers de entrada, una zona de buffers de salida y una zona reservada al trabajo en curso del tratamiento. Los trabajos se ejecutan en secuencia, desarrollándose de forma simultánea las entradas/salidas y el tratamiento.

A cerca de este modo de funcionamiento se pueden hacer las observaciones siguientes: por un lado, cuando para la ejecución del trabajo en curso se deben leer datos, la unidad central queda inactiva mientras se realiza esta operación, y por otro, un trabajo que llegue durante la ejecución de otro quedará en espera, retrasado hasta el final de la ejecución del anterior.

Estas observaciones llevan a considerar un modo de funcionamiento del sistema distinto:

- un trabajo a la espera de ejecución podría utilizar la unidad central liberada por otro trabajo que quede a la espera de entrada o emisión de datos,
- la unidad central debería reasignarse antes del final de un trabajo para superar las variaciones en el tiempo de respuesta.

Para ello, es necesario que el tiempo de reasignación de la unidad central sea muy inferior en relación con la duración de una transferencia entre niveles de memoria. Esto implica la presencia en memoria principal, de varios programas o partes de programas. Este modo de funcionamiento se llama **multiprogramación** y necesita que el sistema operativo incluya un programa que se encargue de realizar la conmutación de la CPU entre procesos, este programa es el **planificador de bajo nivel** o dispatcher.

Las principales ventajas e inconvenientes de los sistemas multiprogramados son las siguientes:

- Complejidad: el sistema operativo es más complejo, ya que debe asegurar la utilización compartida de la memoria y la protección mutua de los programas.
- Se requieren ciertas características hardware, para la reubicación de los programas y la protección de la memoria.
- Aumenta el rendimiento específico del sistema: un sistema multiprogramado permite equilibrar mejor la carga de los diversos recursos

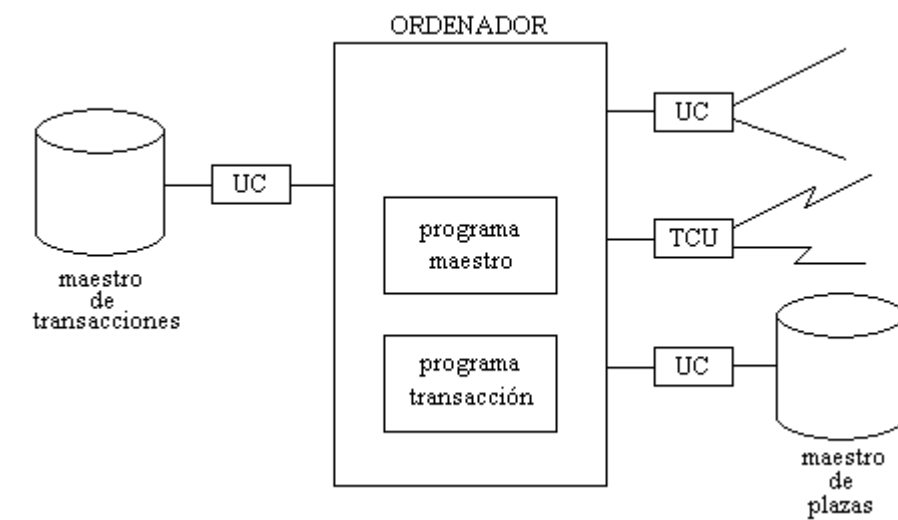
(unidad central, memoria, dispositivos de entrada/salida) incrementando el volumen de trabajo realizado.

- Disponibilidad para el usuario: la multiprogramación permite reducir el tiempo de respuesta para los trabajos cortos en un sistema de tratamiento secuencial.

Es necesario precisar que estas características están ligadas, a la tecnología del momento y a la idea de compartición del hardware. Así los sistemas multiprogramados han evolucionado con la disminución del costo de la memoria principal, el desarrollo de memorias secundarias con acceso rápido y posteriormente, por el desarrollo de sistemas compartidos y ordenadores personales.

### 2.3.5 SISTEMAS TRANSACCIONALES (FASE 3B-II)

Paralelamente, para dar respuesta a las necesidades de las compañías aéreas se desarrolla el procesamiento ON-LINE. Básicamente necesitaban un buen tiempo de respuesta y un fichero maestro de plazas permanentemente actualizado en un dispositivo de fácil y rápido acceso. La solución a estas necesidades vino dada por una combinación hard y soft: se desarrollaron procesadores más rápidos, nuevos periféricos (terminales), un sistema de comunicaciones, y un programa maestro que podía atender simultáneamente varias transacciones (proceso concurrente de transacciones).



Canal					
CPU	P.M.		P. ti		P.M.
					P. tj

Figura 1.7. Reparto de operaciones de los primeros sistemas transaccionales

Las consecuencias de esta organización de entradas/salidas sobre la estructura del sistema, así como de la necesidad de obtener un sistema de uso general que integrase tanto los servicios Batch como los On-Line y que tuviese buen nivel de throughput y un buen tiempo de respuesta y condujeron a la extensión de los sistemas operativos ejecutivos a los **supervisores**.

### 2.3.6 SISTEMAS DE TIEMPO COMPARTIDO O TIME SHARING (FASE 3C-I)

Para hacer realidad la tentativa de interacción lograda en el paso anterior, es decir, para lograr interacción aun coste razonable, se desarrollaron un nuevo tipo de aplicaciones de teleproceso que repartían el tiempo de CPU entre varios usuarios según un esquema de tiempos predefinido (time slicing).

La explotación de los recursos en tiempo compartido ofrece a los usuarios la posibilidad de interacción que tendría con un sistema individual, haciéndole beneficiario de servicios comunes a un precio asequible. Los usuarios tienen acceso al sistema a través de terminales y lo utilizan de forma interactiva. El sistema les garantiza un tiempo de respuesta aceptable (del orden del segundo) para la ejecución de tareas elementales, tales como la edición y la puesta a punto interactivas de programas.

Esta forma de explotación se logra conmutando el procesador sucesivamente, por 'slices' o espacios ('quantum') de tiempo muy breves entre los usuarios. Esto es posible gracias a las características del trabajo interactivo en que la actividad de un usuario está dividida entre un "tiempo de reflexión" durante el cual elabora una solicitud y un "tiempo de espera" durante el cual espera la ejecución del servicio correspondiente por el sistema.

La experiencia demuestra que el tiempo de reflexión es mucho más largo, por término medio, que el tiempo de espera aceptable: Por tanto, el sistema puede servir a numerosos usuarios durante los tiempos muertos introducidos por la reflexión. Un cálculo aproximado, es decir utilizando duraciones medias, permite establecer órdenes de magnitud 1:1000. (Hay que tener en cuenta que las fluctuaciones alrededor de estas medias tienen un importante papel.)

Considérese un sistema de tiempo compartido que sirve a 100 usuarios, cuyo comportamiento medio es idéntico. Se admite que el tiempo de reflexión es por término medio, 9 veces más largo que el tiempo de espera, por lo que éste representa el 10% del tiempo total. Por término medio, habrá 10 usuarios "activos"; es decir, a la espera de ejecución de un servicio solicitado.

Suponiendo que la unidad central se comparte por quantum de 50 ms. y que la ejecución de una solicitud elemental utiliza un sólo 'slice' de tiempo, el tiempo de respuesta será del orden de medio segundo. En este cálculo, de manera implícita se admite que los programas de los usuarios activos se encuentran en memoria principal; así, el tiempo de conmutación entre los programas de dos usuarios se reduce al tiempo de conmutación de la unidad central, tiempo muy inferior a la duración del quantum.

Considerando la relación entre el quantum y el tiempo de carga de un programa desde disco, la multiprogramación es, de hecho, necesaria para el funcionamiento de un sistema en tiempo compartido. Pero el sistema realmente es más complejo, ya que el tamaño de la memoria principal no permite tener cargado simultáneamente todo el conjunto de los programas de los usuarios activos. Por consiguiente es posible que una información no esté en memoria principal en el momento que la unidad central deba tener acceso a ella.

El papel del sistema de gestión de la memoria es reducir, tanto como sea posible, la probabilidad de este evento.

El desarrollo de los sistemas en tiempo compartido pone de manifiesto la importancia de la interacción hombre-máquina y con especial énfasis, la del lenguaje de control que es el interfaz presentado por el sistema a los usuarios. Durante mucho tiempo no se consideró este aspecto; los lenguajes de control concebidos de manera empírica, eran muy a menudo rígidos e incómodos.

El sistema Unix, concebido en 1970-71 y actualmente muy extendido, debe gran parte de su éxito a su flexibilidad, a su potencia y a la facilidad de utilización de su lenguaje de control.

La aparición y la extensión de terminales gráficos, que permiten seguir simultáneamente (mediante "ventanas") la evolución de múltiples actividades y combinar texto, imágenes y quizás entradas/salidas habladas, hacen vislumbrar profundas modificaciones en este campo y transformaciones de la noción misma de lenguaje de control.

### 2.3.7 SISTEMAS DE TIEMPO REAL (FASE 3C-II)

Los primeros trabajos en tiempo real estaban dirigidos a dar solución a problemas concretos: control de procesos, seguimiento, etc. y en ellos, los sistemas en tiempo real se utilizaban como dispositivos de control en aplicación dedicadas.

Los sistemas operativos en tiempo real tienen restricciones temporales bien definidas, por lo que la toma de datos, el procesamiento y el control deben realizarse dentro de unos límites bien definidos.

#### 2.4 Cuarta Fase

Los usuarios de sistemas informáticos presentan nuevas necesidades:

- Demanda de utilización compartida de recursos, justificada por consideraciones económicas y por la necesidad de acceso a informaciones y recursos compartidos, que pueden estar distribuidos geográficamente.
- Búsqueda de una mejor adecuación de la estructura de los sistemas, a la de las aplicaciones tratadas; lo que lleva a la descentralización del sistema y a la posible distribución geográfica de sus elementos.
- Necesidad de integrar en un conjunto único de aplicaciones, las desarrolladas por independiente.

Para responder a estas necesidades van apareciendo sucesivamente nuevos tipos de sistemas informáticos:

1. Redes de teleinformática que permiten la interconexión de sistemas existentes, la transmisión de datos entre estos sistemas y el acceso a servicios alejados.
2. Redes locales, construidas sobre una vía de comunicación de flujo grande (10 Mbit/s), geográficamente concentradas y concebidas para aplicaciones concretas:
  - a. sistemas de control de procesos (o en general sistemas de tiempo real),
  - b. sistemas de comunicación y de gestión de documentos (o sistemas de burótica), orientados hacia el intercambio de información, la producción y archivo de documentos transaccionales.

Las redes locales de uso general tienden a sustituir a los sistemas de tiempo compartido; cada usuario dispone de su máquina individual cuyos rendimientos son ahora suficientes para la mayor parte de las aplicaciones y la red permite:

- la comunicación entre usuarios y
- el acceso a servicios comunes, demasiado costosos para ser individuales: almacenamiento de grandes cantidades de información, impresión de documentos, procesadores de gran potencia, etc. Estos servicios están

gestionados por sistemas especializados o "servidores", a los cuales se dirigen los clientes solicitantes de servicios.

Para la concepción de los servidores y de los sistemas que gestionan las máquinas individuales, los conceptos y las técnicas desarrollados para los sistemas centralizados siguen siendo válidos.

Por otro lado, la distribución y la comunicación de información introducen nuevos problemas, tales como la coordinación de actividades a distancia y el mantenimiento de la coherencia e integridad de informaciones compartidas.

Sin embargo, el desarrollo de las redes y de las máquinas individuales hacen que tiendan a desaparecer los grandes sistemas centralizados y los sistemas en tiempo compartido (justificados económicamente para numerosas aplicaciones).

Las necesidades de cálculo numérico para aplicaciones especializadas (aerodinámica, meteorología, etc..) llevan, por otra parte, a un desarrollo de sistemas de potencia muy elevada que incluye multiprocesadores especializados (procesadores vectoriales), para los cuales quedan por desarrollar nuevas arquitecturas de sistemas operativos.

En resumen, en esta fase, es decir, en el comienzo de los años 80, conviven redes, sistemas compartidos y máquinas individuales. Esta década está marcada por dos fenómenos:

1. La aparición de los microprocesadores y el crecimiento de sus prestaciones, que permiten disponer de una gran potencia de cálculo con costes cada vez más pequeños.
2. El desarrollo de las técnicas de transmisión de datos (teleinformática) y la progresiva integración de la función de comunicación en los sistemas informáticos (telemática).

En el gráfico de la figura 1.8, se puede ver la evolución en los últimos años de los sistemas operativos y los diversos conceptos comentados, distinguiendo tres clases de ordenadores en función de su tamaño y capacidad.

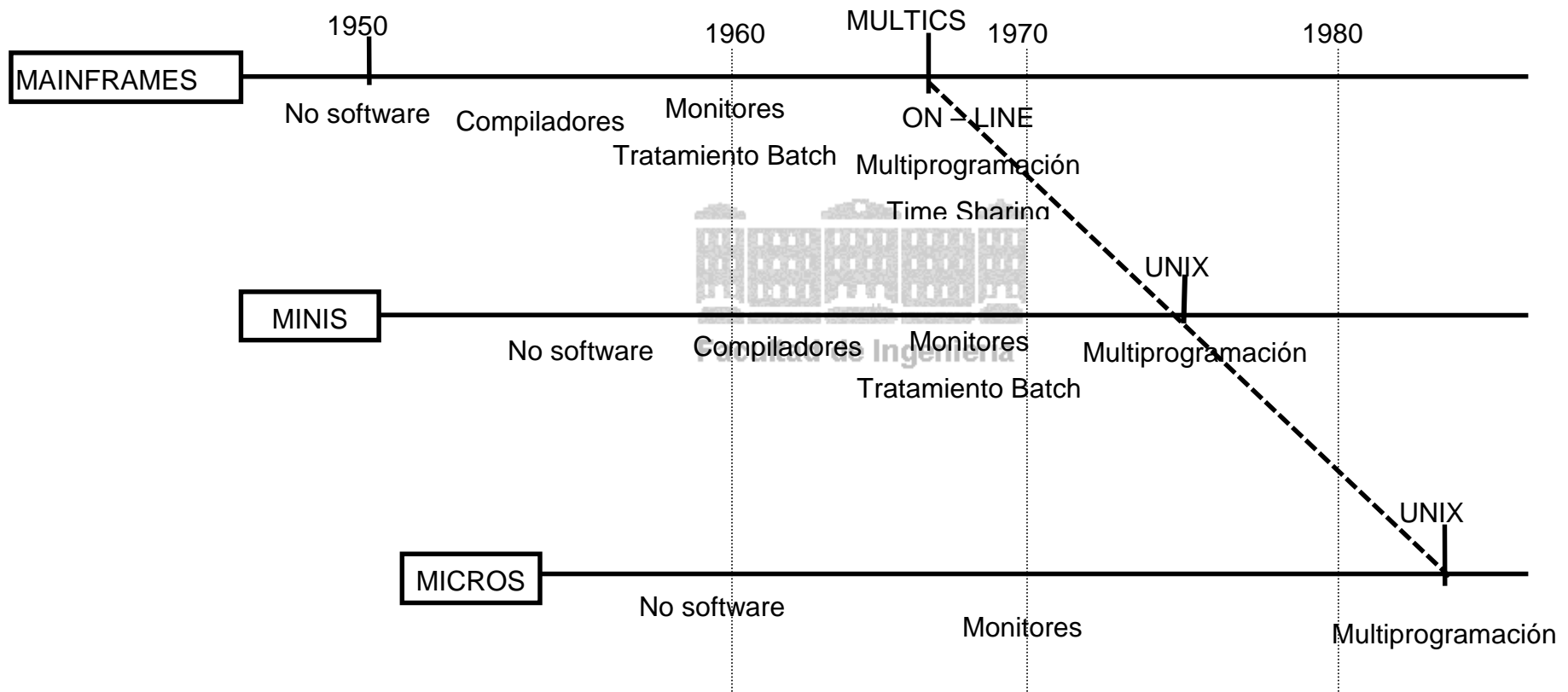


Figura 1.8. Evolución de los sistemas operativos

## 2.5 Tendencias

El camino hacia los sistemas futuros muestra una serie de tendencias claras:

### **User-Friendly**

Denotan sistemas que proporcionan fácil acceso (vía menús, iconos, manipuladores gráficos,...) y manejo de recursos, que no requieren amplios conocimientos o experiencia en computadores para su correcta utilización y es cómodo para trabajar. Así, los computadores personales serán omnipresentes y su utilización como recurso será menos significativo que su disponibilidad, fiabilidad, flexibilidad y facilidad de uso en relación al usuario.

### **Máquina virtual**

Denota cómo el usuario no tiene que preocuparse de los detalles físicos sino de la máquina que se presenta por medio del sistema operativo, la máquina real quedará oculta al usuario. El concepto de almacenamiento virtual perdurará. Las aplicaciones contemplarán máquinas virtuales, pudiéndose ejecutar en diferentes miembros de una familia de ordenadores. Así, si en una red de computadores un trabajo de usuario podrá ser realizado por un computador del no tenga conocimiento.

### **Proceso de datos distribuido**

Son sistemas en los que pueden cooperar varios ordenadores independientes interconectados. Cada proceso puede tratar datos locales y tomar decisiones locales, y también intercambiar información a través de una red, procesando información o leyendo decisiones que afectan a varios procesos. El concepto de proceso distribuido provocará que sean desarrollados sistemas operativos dispersos en los que sus funciones sean distribuidas entre varios procesadores a través de grandes redes de sistemas. El costo de la comunicación de datos continuará disminuyendo y las velocidades de transmisión de datos aumentarán.

### **Proceso de datos en paralelo**

El precio del hard continuará decrementándose, las capacidades de procesamiento y almacenamiento aumentarán y la escala de integración continuará aumentándose (VLSI a ULSI) lo que permitirá que el tamaño físico de procesadores y memorias disminuya, convirtiéndose en habituales las máquinas masivamente paralelas. Las arquitecturas distribuirán el control del sistema entre procesadores

localizados y los lenguajes se extenderán para poder explotar la concurrencia con más eficiencia.

Los sistemas operativos acordes con estos conceptos, tendrán muchas de sus funciones implementadas en microcódigo; de esta forma y aplicando los avances en ingeniería del software se logrará que sean más mantenibles, fiables y comprensibles.

