



ARQUITECTURAS DE LOS SISTEMAS OPERATIVOS

“Lo importante de los sistemas operativos es no olvidar que constituyen una parte integral de un ambiente de cómputo y, por tanto, necesitan ser comprendidos en alguna medida por todos los usuarios de computadoras”.¹

Harvey M. Deitel

INDICE	PÁGINAS
⇒ INTRODUCCIÓN	3
⇒ HISTORIA	4
⇒ FUNCIONES	7
⇒ CARACTERÍSTICAS GENERALES	8
⇒ CLASIFICACIÓN	10
⇒ COMPONENTES DE UN SISTEMA OPERATIVO	11
⇒ MONITOR	11
⇒ ADMINISTRADOR DE MEMORIA	12
⇒ ADMINISTRADOR DE DISPOSITIVOS PERIFÉRICOS	13
⇒ ADMINISTRADOR DE INTERRUPCIONES	15
⇒ EXISTEN SEIS CLASES DE INTERRUPCIONES	16
⇒ ADMINISTRADOR DE TRABAJOS A SER EJECUTADOS	17
⇒ MÉTODOS DE ASIGNACIÓN DE LA CPU PARA MULTIPROGRAMACIÓN	19
⇒ ADMINISTRADOR DE DATOS	22
⇒ LIBRERÍAS	23

¹ Deitel, H. M. (1993); *Introducción a los Sistemas Operativos: Segunda Edición*; Wilmington, Delarare; Addison-Wesley Iberoamérica S.A.; página 4.



INDICE	PÁGINAS
⇒ TIPOS DE ESTRUCTURAS DEL NÚCLEO	24
⇒ MONOLÍTICA.	24
⇒ JERÁRQUICA.	26
⇒ ARQUITECTURA MICRONÚCLEO.	26
⇒ NÚCLEO HÍBRIDO.	
⇒ CARACTERÍSTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS	27
⇒ MULTITHREADS.	28
⇒ MULTIPROCESAMIENTO SIMÉTRICO.	28
⇒ SISTEMAS OPERATIVOS DISTRIBUIDOS.	28
⇒ DISEÑO ORIENTADOS A OBJETOS.	
⇒ FORMA DE TRABAJO	
⇒ MÁQUINA VIRTUAL	28
⇒ CLIENTE-SERVIDOR	29
⇒ SERVICIOS QUE OFRECE	30
⇒ FORMAS DE OFRECER LOS SERVICIOS	32
⇒ PUNTO A PUNTO	32
⇒ CON SERVIDOR DEDICADO	32
⇒ MULTIPROGRAMACIÓN Y MULTIPROCESAMIENTO	33
⇒ MULTIPROGRAMACIÓN	33
⇒ MULTIPROCESAMIENTO	35
⇒ SISTEMAS DISTRIBUIDO	36
• CLUSTER	37
⇒ DIFERENTES SISTEMAS OPERATIVOS	
⇒ MONOUSUARIO	38
⇒ TIEMPO COMPARTIDO	39
⇒ TIEMPO REAL	41



INTRODUCCIÓN

El concepto sobre lo que se conoce en el área de sistemas como Arquitectura se puede plantear desde dos puntos de vista: el relacionado con el Hardware y el relacionado con el Software.

Desde el punto de vista del Hardware se plantean los distintos aspectos basados en la estructura física de una computadora. Allí nos encontramos con distintos elementos que en algunos casos funcionan mecánicamente (como por ejemplo algunos dispositivos periféricos) y en otros (su gran mayoría), su modo de funcionamiento está basado exclusivamente en la Electrónica.

Esto nos permitirá entender que aunque en una computadora los dispositivos existen físicamente, su modo de funcionamiento está basado en la lógica, dado que la misma esta formada por circuitos electrónicos que trabajan teniendo en cuenta las ideas originales planteadas por Pascal, Babbage, Von Neumann, Boole, etc.

Desde el punto vista del software debemos tener en cuenta, que el hardware es en sí un elemento físico, que comienza a funcionar a partir de órdenes y estas órdenes forman parte exclusivamente del software.

El software está formado básicamente por programas y por los datos que estos necesitan para poder darle utilidad al hardware. Todo programa está formado por instrucciones, y una instrucción es la orden que recibe el hardware para comenzar con su funcionamiento.

El grupo de programas más importante dentro del software son los **Sistemas Operativos**, estos están formados por un grupo de componentes o rutinas que ocultan toda la complejidad que tiene el hardware y de esta manera permitir que un sistema de computación, sea posible de usar por cualquier tipo de usuarios sin necesidad que sean especialistas.

En este capítulo trataremos de analizar las características generales de los sistemas operativos y saber responder a los siguientes interrogantes: qué es un sistema operativo, qué funciones cumplen, cuales son sus características principales, de que rutinas esta compuesta su estructura principal, cómo los podríamos clasificar y de qué tipo existen. De esta forma trataremos de ir entendiendo de manera más detallada la importancia de su existencia y sus principales funciones.



HISTORIA²

- **Los años cuarenta y los años cincuenta:**

Los sistemas operativos han evolucionado durante 40 años a través de distintas fases o generaciones que coinciden más o menos con las décadas. Los primeros sistemas de computación electrónicos digitales de los años cuarenta no tenían sistema operativo; las máquinas de esa época eran tan primitivas que con frecuencia los programas se introducían bit a bit mediante grupos de interruptores mecánicos. Posteriormente los programas se introducían en el sistema de computación en lenguaje de máquina mediante tarjetas perforadas y se desarrollaban lenguajes ensambladores para acelerar el proceso de programación.

En General Motors Research Laboratories se implantó el primer sistema operativo a principios de los años cincuenta para su IBM 701.

Los sistemas de los años cincuenta generalmente ejecutaban sólo una tarea por vez y simplificaban la transición entre tareas para obtener la máxima utilización del sistema de computación. Estos sistemas se denominaron **sistemas de procesamiento por lotes de secuencia única**, ya que los programas y los datos eran proporcionados en grupos o lotes.

- **Los años sesenta:**

Los sistemas de los años sesenta también eran sistemas de procesamiento por lotes, pero podrían aprovechar mejor los recursos del sistema de computación mediante la ejecución de varias tareas al mismo tiempo. Esos sistemas incluían dispositivos periféricos, como lectoras de tarjetas, perforadoras de tarjetas, impresoras, unidades de cinta y unidades de disco, pero era raro que una tarea utilizara a todos los recursos de un sistema de computación.

Los diseñadores de sistemas operativos observaron que mientras una tarea esperaba a que se completara una operación de E/S para poder seguir usando la CPU, otra tarea podría aprovechar al máximo el procesador ocioso. De manera similar, cuando una tarea estaba utilizando el procesador, otras podrían estar utilizando los diversos dispositivos de E/S.

Los diseñadores de sistemas operativos desarrollaron el concepto de **multiprogramación**, en el cual varias tareas se encuentran almacenadas en la memoria principal, permitiendo de esa manera que la CPU pueda conmutar una tarea a otra, según sea necesario para lograr que avancen varias tareas, mientras se mantienen en uso los dispositivos periféricos.

En 1964, IBM presentó su familia de sistemas de computación System/360 que fueron diseñados para ser de Hardware compatible para poder utilizar el sistema operativo OS/360.

² Deitel, H. M. (1993); *Introducción a los Sistemas Operativos: Segunda Edición*; Wilmington, Delaware; Addison-Wesley Iberoamérica S.A.; página 4.



Se desarrollaron sistemas operativos mas avanzados para atender al mismo tiempo a varios **usuarios interactivos**, que se comunicaban con el sistema de computación principal por medios de terminales que están en línea, o sea que se comunican directamente con él. Los sistemas de **tiempo compartido** se desarrollaron para permitir trabajar simultáneamente a un gran número de usuarios interactivos.

Muchos de los sistemas de tiempo compartido de los años sesenta eran sistemas con múltiples modos de operación, que también realizaban procesamiento por lotes y aplicaciones en **tiempo real** (como los sistemas de control de procesos industriales), y unas de sus principales características es que brindan una respuesta inmediata a los distintos eventos que ocurran en una secuencia de procesos.

Multics, fue el primer sistema operativo escrito con un lenguaje de programación de alto nivel, en lugar de estar escrito con un lenguaje ensamblador. Debido a esto los diseñadores del sistema operativo **UNIX** crearon el lenguaje C de programación.

Multics, TSS³ y CP/CMS⁴ incorporaron el concepto de **almacenamiento virtual**, con el cual, entre otro beneficios, los programas pueden hacer referencia a cantidades de memoria mucho mayores que la cantidad de memoria principal de un sistema de computación.

- **El surgimiento de una nuevo campo: La Ingeniería de Software**

Los sistemas operativos desarrollados durante los años sesenta eran grandes conglomerados de software escritos por que realmente no se comprendía que el software, lo mismo que el hardware, debía estar diseñado de modo que fuera confiable, comprensible y fácil de mantener. Debido al volumen de información que se generaba, los informáticos y profesionales de la industria comenzaron a dedicar considerables recursos a los problemas de la construcción de los sistemas de software, lo cual dio origen al campo de la **ingeniería de software**. El surgimiento de esta ingeniería y el reconocimiento de la importancia de desarrollar un método disciplinado y estructurado para la construcción de software confiable, comprensible y fácil de mantener, fueron fuertemente impulsados por las nefastas experiencias en el desarrollo de los sistemas operativos de los años sesenta.

- **Los años setenta:**

Los sistemas de los años setenta eran sobre todo sistemas de tiempo compartido con múltiples modos de operación, que permitían aplicaciones de procesamiento por lotes, de tiempo compartido y de tiempo real. Las computadoras personales estaban en la etapa inicial de su desarrollo y fueron impulsados por el avance inicial y continuo de la tecnología de los microprocesadores. Los sistemas experimentales de tiempo compartido de los años sesenta se transformaron en productos comerciales

³ En programación, una **estructura de datos** es una forma de organizar un conjunto de datos elementales con el objetivo de facilitar su manipulación. Un dato elemental es la mínima información que se tiene en un sistema. Un array es una estructura de datos.

⁴ El **CP / CMS** es un sistema operativo de tiempo compartido desarrollado por IBM. El mismo se desarrolló y utilizó durante la década de 1960 y comienzos de la década de 1970.



sólidos de los años setenta. Con ello aumentaron las comunicaciones entre los sistemas de computación en todo los Estados Unidos, así se generalizó el uso de los estándares de comunicaciones **TCP/IP** del departamento de defensa, especialmente en los ambientes militares y universitarios. Las comunicaciones en las **redes de área local** (Local Area Networks, LAN) se hicieron prácticas y económicas con el estándar **Ethernet** desarrollado en el Centro de Investigación de Palo Alto de Xerox.

Los problemas de seguridad aumentaron a causa de los grandes volúmenes de información que pasaban vulnerables por las líneas de comunicación, por ese motivo la **criptografía** recibió mucha atención, dado que fue necesario codificar los datos propios o privados de forma, que aunque fueran robados, carecieran de valor para cualquiera que no fueran los destinatarios originales.

- **Los años ochenta:**

La década de los años ochenta fue la de la **computadora personal** (PC) y de la **estación de trabajo**. La tecnología de los microprocesadores evolucionó hasta el punto en que fue posible construir sistemas de computación de escritorio tan poderoso como las microcomputadoras (mainframe) de los años setenta.

En los años setenta, sólo las grandes organizaciones podrían comprar sistemas de computación y utilizar la computación interactiva, en los ochenta, casi cualquiera podría tener su propio sistema de computación o tener acceso a uno.

La clave de esta década fue la de poder transferir información en las redes de computadoras, de esta forma proliferaron las aplicaciones de correo electrónico, transferencias de archivos y acceso a base de datos remotas.

- **Los años noventa y el futuro**

En los años noventa entramos en la era de una auténtica **computación distribuida**, en la cual los cómputos que realice un proceso se dividirán en subcómputos que podrán ejecutarse en otros procesadores.

Las redes tendrán una **configuración dinámica**, seguirán operando aunque se añadan o eliminen dispositivos o nuevos tipos software.

La conectividad se facilitará gracias a los estándares y protocolos de los sistemas abiertos. El mundo adoptará la filosofía arquitectónica de los **sistemas abiertos**, donde abundan arquitecturas abiertas de amplia disponibilidad, suministradas por muchos proveedores, como así también habrá lugar para arquitecturas de diseño propio que interactuarán con arquitecturas de los sistemas abiertos.

Los sistemas de computación están destinados a tener gran capacidad y a ser fáciles de transportar. Con el desarrollo de los protocolos de comunicación (OSI) y especialmente el de la red digital de servicios integrados (ISDN), las personas podrán comunicarse y transferir datos con una gran confiabilidad, además serán cada vez más accesibles y sus costos disminuirán constantemente.



FUNCIONES

- **Principales:**

1. **Crear una máquina virtual:** Otra función importante de los sistemas operativos en general, es la de ocultar toda la complejidad del hardware, transformando a la computadora en una máquina fácil de usar. Esto es el equivalente en colocar frente a los distintos usuarios una máquina virtual cuyas características sean distintas y más fáciles de manejar que una máquina real.
2. **Utilización compartida de recursos:** Es importante saber que un sistema operativo debe lograr que se compartan los recursos que formen parte de una computadora entre un cierto número de usuarios que trabajen en forma simultánea. Esto es importante ya que de esta forma se busca incrementar la disponibilidad de una computadora y maximizar la utilización de los distintos recursos entre los distintos usuarios.

Recursos que administra un Sistema Operativo ⁵

La memoria
Las unidades de comando (Periféricos)
Los dispositivos de E/S (Interrupciones)
Los trabajos a ser ejecutados (Procesos)
Los datos

- **Complementarias:**

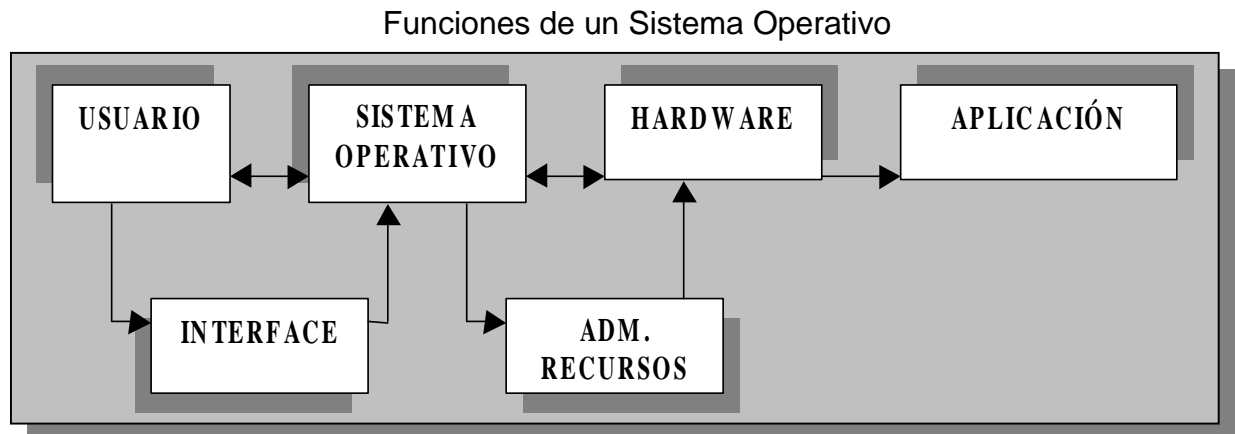
1. **Controlar los recursos existentes:** Debe tener en cuenta todos los recursos existentes, en que estado se pueden encontrar y las funciones que puedan realizar en cualquier momento.
2. **Proveer servicio:** Para la ejecución de los programas de aplicación y para el desarrollo de los mismos.
3. **Actuar como entorno de la aplicación en la cual el programa es ejecutado:** Esto significa aplicar un conjunto de rutinas para la supervisión de las operaciones que realiza la computadora, controlar el desarrollo de las operaciones internas de la CPU y dirigir las operaciones de E/S.
4. **Administrar las tareas:** Debe administrar la utilización de los diferentes recursos para las distintas tareas que se estén realizando durante una secuencia de procesamiento.
5. **Secuenciar las tareas:** Debe llevar un cierto orden con respecto a los diferentes trabajos que debe realizar.

⁵ Stallings, W. (2001); *Sistemas Operativos: 4ª Edición*; Madrid; Pearson Educación S. A.; página 10.



6. **Interpretar un lenguaje de control:** Debe comprender los diferentes tipos de órdenes que se le imparten para poder ejecutar las tareas.
7. **Administrar errores:** Debe tomar las adecuadas acciones para los diferentes tipos de errores que se produzcan durante el procesamiento de la información, como así también permitir la intervención externa de un usuario.
8. **Administrar las interrupciones:** Debe saber interpretar y satisfacer todas las interrupciones que se puedan llegar a producir durante el procesamiento de la información.
9. **Proteger:** Debe proteger la información de todos los usuarios, como así también debe asegurar la integridad de los datos.

Las funciones principales que cumple un sistema operativo con respecto a la relación que existe entre un usuario con el hardware para la realización de una aplicación se pueden graficar de la siguiente manera:



CARACTERÍSTICAS GENERALES

1. Concurrencia:

Para un sistema operativo la concurrencia consiste en la existencia de varias actividades simultáneas o paralelas durante una secuencia de procesamiento, la concurrencia implica paralelismo, simultaneidad y “pipeline”.

Los eventos paralelos o simultáneos pueden ocurrir entre diferentes recursos en un mismo intervalo de tiempo, en el caso de la multiprogramación, dos o más tareas quieren competir con el uso del procesador, desde el punto de vista del usuario, parecería que ocurre realmente, dada la velocidad del procesador, pero en la realidad esto nunca puede ocurrir, porque un procesador sólo puede atender una tarea por vez y lo que hace un sistema operativo es intercambiar las distintas tareas de acuerdo al orden en que vayan llegando.



En el caso que el sistema trabaje con más de un procesador, sí puede ocurrir que los distintos eventos se ejecuten de manera simultánea o paralela, dado que cada procesador podría atender a una tarea diferente en un mismo espacio de tiempo. En este caso los eventos “pipeline” pueden ocurrir en intervalos de tiempo superpuestos. A modo de ejemplo, si el sistema tuviera una unidad interna de hardware en la CPU que estuviera ejecutando una instrucción y otra una unidad que cargue en memoria la próxima instrucción a ejecutar, mientras que se decodifica el código de la instrucción actual, se estaría cargando la próxima instrucción en la CPU. Luego se superponen en el tiempo ambas funciones.

Estos eventos concurrentes se pueden realizar en un sistema de computación con varios niveles de procesamiento. El procesamiento paralelo necesita de la ejecución concurrente de muchos programas en la computadora. Esto es justamente la contrapartida del procesamiento secuencial.

2. Utilización conjunta de Recursos:

Para que un sistema pueda realizar varias tareas concurrentes deberá poder compartir los distintos recursos, para ello el sistema operativo debe proteger los mismos y evitar los problemas durante el acceso simultáneo a la información y a la ejecución simultánea de programas, para ello deberá tener en cuenta:

- 1) No disponer de infinitos recursos, esto aumentaría su costo en el procesamiento de la información.
- 2) Eliminar la información redundante.
- 3) Poder compartir los datos.
- 4) Permitir seguir trabajando a partir de los que hicieron otros.

3. Almacenamiento Permanente:

Esta es una de las características generales de todo sistema operativo, dado que para su utilización debe estar almacenado permanentemente en un medio de almacenamiento de cualquier tipo (electrónico, magnético u óptico), a partir de esto durante el inicio del sistema, buscará en el dispositivo preseleccionado a la parte principal del sistema operativo (núcleo) para cargarla en la memoria RAM y a partir de esto generar la interfaz para que los usuarios puedan acceder a la utilización de los distintos recursos de un sistema de computación.

Este tipo de almacenamiento permite además que los distintos usuarios de un sistema guarden sus datos o programas en los medios de almacenamiento correspondientes. El sistema operativo debe proporcionar un acceso fácil a los distintos datos, proteger la información de accesos no deseados y realizar copias de seguridad periódicas como resguardo de posibles fallas en el sistema.



CLASIFICACION

Los distintos tipos de sistemas operativos se pueden clasificar de la siguiente manera: **por la función de sus componentes**, **por las características de su residencia en memoria**, **por su estructura** (visión interna), **por su forma de trabajo**, **por los servicios que ofrecen** y finalmente, **por la forma en que ofrecen sus servicios** (visión externa).

1. **Por la función que cumplen los distintos componentes que forman parte la estructura principal de un sistema operativo (Núcleo):**
 - a. Monitor.
 - b. Administración de memoria.
 - c. Administración de las unidades de comando.
 - d. Administración de los dispositivos de E/S.
 - e. Administración de los trabajos a ser ejecutados.
 - f. Administración de los datos.
 - g. Librerías.
2. **Por la residencia en memoria de sus componentes:**⁶
 - a. Residentes.
 - b. Transitorios.
3. **Por el tipo de Estructura del núcleo.**
 - a. Monolítica.
 - b. Jerárquica.
 - c. Micronúcleo / Microkernel.
 - d. Híbrido.
4. **Por la forma de trabajo:**
 - a. Máquina Virtual.
 - b. Cliente-Servidor.
5. **Por los servicios que ofrece:**
 - a. Cantidad de:
 - i. Usuarios.
 - ii. Tareas.
 - iii. Procesadores.
6. **Por la forma de ofrecer los servicios:**
 - a. Sistemas Operativos Multiusuarios:
 - i. Punto a punto (Sistema Operativo Distribuido).
 - ii. Con servidor dedicado (Sistema Operativo de Red).

⁶ Programas Residentes y Transitorios. Capítulo I: Introducción a la Informática. Pág. 65.



COMPONENTES DE UN SISTEMA OPERATIVO

Se define como componentes a las distintas rutinas que forman parte de un sistema operativo. Cada componente cumple una función diferente y efectúan en conjunto la administración de los distintos recursos del sistema.

Estos componentes pueden permanecer permanentemente residentes en memoria y en algunos casos trabajar de forma transitoria, dependiendo del recurso que necesite utilizar el sistema operativo durante el procesamiento de la información.

Para administrar los recursos de la computadora el sistema operativo consta de 7 componentes:

1. Monitor / Supervisor / Ejecutivo / Núcleo / Kernel.

Es el componente central del sistema operativo, está siempre residente en memoria. El monitor es el que coordina y controla todos los demás componentes del sistema operativo.

Este componente trata todas las operaciones de E/S, los estados en que se encuentran las interrupciones y otras funciones de control de todos programas. Detecta todas las interrupciones que se producen y las almacena en las distintas colas de trabajo según el orden de las mismas. Lleva el control de los estados de los distintos programas interrumpidos y los reanuda cuando la interrupción ha sido procesada.

“El núcleo normalmente representa sólo una pequeña parte de lo que por lo general se piensa que es todo el sistema operativo, pero es tal vez el código que más se utiliza. Por esta razón, el núcleo reside por lo regular en la memoria principal, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesita”¹.

Este componente es traído a la memoria por un programa, generalmente independiente del sistema operativo denominado IPL (Initial Program Loader y/o Programa cargador inicial), Boot Record (Registro de encendido) o MBR (Registro de encendido maestro) cuya única misión es traer a la memoria al monitor, por este motivo se le considera que forma parte del sistema operativo. Una vez cargado el monitor en memoria este desaparece, en consecuencia diremos que el programa IPL, Boot Record o MBR es transitorio del sistema operativo.

¿Cuándo utilizaremos nuevamente este programa?: Cuando necesitemos cargar el monitor a la memoria, esto puede ser necesario por dos motivos:

- a. Para poder darle uso al software y el hardware que tengamos en la computadora en el momento del encendido.
- b. Por que el sistema ha quedado detenido y se necesite reencender el equipo, esto se realiza con un hardware del tipo PC, pulsando la tecla RESET que se encuentra en el gabinete o pulsando en forma correlativa las teclas Ctrl+Alt+Del.



Cuando pulsamos la tecla RESET realizamos un reencendido completo, o sea, que cargará todos los componentes del sistema cómo si recién encendiéramos el equipo. Sí el reencendido lo realizáramos pulsando las teclas Ctrl+Alt+Del el reencendido se realizaría en forma parcial, o sea, cargando solamente a algunos de los componentes del sistema operativo.

El programa IPL funciona con microcircuitos que cuando se reinicia el sistema por cualquiera de los métodos se pone a leer en la unidad de disco una dirección predeterminada y fija, y carga en la memoria a lo que se conoce como **Núcleo del Sistema Operativo** y le transfiere el control para que empiece a ejecutarse.

2. **Administrador de Memoria / Storage Allocator.**

Este componente realiza 3 funciones:

- a. Es responsable de la asignación de memoria principal a los programas, mantiene un mapa de memoria que le indica al administrador de los trabajos a ser ejecutados, que áreas de memoria están disponibles, y que áreas de memoria han sido asignadas a programas.
- b. En los sistemas operativos que permiten que más que un programa se encuentre residente en memoria al mismo tiempo (multiprogramación) se debe asegurar que ningún programa use memoria asignada a otros programas, esto se denomina protección de memoria.
- c. Es responsable del uso de almacenamiento auxiliar para el almacenamiento temporario de todo o parte de un programa durante su ejecución, esta función se requiere particularmente para dos operaciones: Swapping ⁷ y Overlaying ⁸. El Swapping y el Overlaying son técnicas de intercambio que se utilizan en el caso que la memoria real de una computadora no sea suficiente, implementándose el uso de memoria virtual.

3. **Administrador de Dispositivos Periféricos / Administrador de las Unidades de Comando / Device Handler.**

Es un conjunto de rutinas que conectan al sistema y a los programas de los usuarios con los dispositivos periféricos. Hay un Device Handler por cada tipo de periférico, son necesarios porque cada periférico realiza transferencias de datos hacia la memoria principal.

Con respecto a la administración de periféricos nos encontramos con 4 aspectos básicos:

- a. Administración de los canales y unidades.
- b. Administración de ciertas unidades (impresora, discos).
- c. Independencia de ciertas unidades.

⁷ Peterson, J. L., Silberschatz, A., *Sistemas Operativos: Conceptos fundamentales*, Editorial Reverté S. A., Barcelona, España, 1989, pág. 162.

⁸ <http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/capitulo4.html>



a. Administración de los canales y unidades:

Todas las unidades periféricas se encuentran vinculadas a la CPU a través de los canales de E/S de distinto tipo. Un canal no es simplemente un cable que une la unidad periférica con la CPU, sino que está constituido, además, por un procesador de E/S que puede trabajar simultáneamente con la CPU y que permite, en definitiva, el trabajo de multiprogramación. A su vez, un canal puede realizar de a una operación por vez, mediante los canales selectores, o más de una, mediante los canales multiplexores.

b. Administración de ciertas unidades:

Si consideramos que cada programa que se está ejecutando puede solicitar distintas operaciones de E/S, será el monitor quién se encargará de analizarlas. Lo primero que debe hacer es verificar el tipo de canal que va ser utilizado para la operación, el estado en que se encuentra y si existen operaciones pendientes para ese canal. Lo más simple sería que el canal estuviera libre y que no existiera ninguna operación pendiente. En este caso el monitor le daría al procesador del canal las instrucciones necesarias para que efectúe la operación solicitada. Sin embargo, es probable que el canal se encuentre en uso. Si el canal es multiplexor, o sea que admite operaciones simultáneas, controla si todos los subcanales están siendo usados, si es así, no admite la operación, estos tipos de subcanales contienen un buffer⁹, un registro de dirección del dispositivo y las señales de control, que le indican al sistema operativo el estado en que se encuentra el dispositivo. Si es un canal selector, o sea que admite una sola operación, tampoco aceptará la operación hasta tanto no termine la que se esté ejecutando. En estos casos el monitor registrará la operación en una cola de espera que normalmente no tiene prioridades, o sea que se ejecutarán en el orden que fueron ingresadas.

Cuando el canal ejecute la interrupción indicando la terminación de una operación, el monitor tomará la primera operación de la lista de pendientes (cola de espera) y le dará al procesador del canal las instrucciones necesarias para efectuarla.

Cada canal conecta a la CPU con un dispositivo periférico, cada dispositivo tiene diferentes características, no podemos comparar una unidad de disco con una impresora, podemos decir entonces que existen dispositivos de acceso compartido y otros de acceso dedicado.

Los dispositivos de acceso directo permiten el acceso compartido entre distintos programas, cómo por ejemplo el disco, donde se pueden almacenar gran cantidad de archivos, pudiéndose ejecutar distintos programas aparentemente al mismo tiempo, necesitando alternativamente distintos archivos, ya que los brazos móviles de la unidad de disco permiten el acceso directo, o sea grabar o leer archivos.

En cambio, si seria una impresora, al ser una unidad de acceso secuencial, la podrían utilizar alternativamente varios programas, no se obtendría listados diferenciados, sino que se imprimiría líneas intercaladas de los listados de los diferentes programas.

⁹ Peterson, J. L., Silberschatz, A., *Sistemas Operativos: Conceptos fundamentales*, Editorial Reverté S. A., Barcelona, España, 1989, pág. 13.



Los dispositivos de acceso directo permiten el acceso compartido entre distintos programas. En cambio si el dispositivo es de acceso secuencial el monitor asigna el dispositivo a un programa, y no asignará otro hasta que este no termine, ya que se tratará de un dispositivo de acceso dedicado.

Ejemplo:

1) Impresora:

El problema planteado en el uso de una impresora por varios programas puede traer serios inconvenientes en el aprovechamiento de una computadora. Debemos considerar que mientras se imprime un listado todos los demás programas se encontrarán en la memoria, lo que implica un gran desaprovechamiento de este recurso del sistema operativo, sobre todo si tenemos en cuenta que la impresora es uno de los dispositivos más lentos, por lo menos comparado con las unidades de almacenamiento.

Es así como una gran cantidad de sistemas operativos (no todos) cuentan con una facilidad denominada Spooling ¹⁰ (Simultaneous Peripheral Operation On-Line / Operación Periférica Simultánea en Línea) que es aplicable a todos los dispositivos lentos.

El sistema es bien sencillo y consiste en que cada vez que un programa requiere el uso de la impresora, abre un archivo asignado a la impresora. El sistema operativo mediante el monitor abre un archivo en una unidad de acceso directo (espacio de memoria virtual).

Todo lo que contenga un spooling ya ha sido procesado por la CPU y esta esperando que sea enviado hacia el dispositivo correspondiente.

Además cada vez que un programa solicita una operación de impresión, el monitor desvía la impresión y la graba en el archivo correspondiente.

Es decir que el Spooling hace independizar totalmente los conceptos de impresora física e impresora lógica. La computadora tendrá todas las impresoras virtuales necesarias, independientemente de las impresoras virtuales.

Puede haber dos posibilidades del uso del Spooling, que sea parte del sistema operativo o que sea un programa aparte que requerirá un espacio en memoria para su ejecución.

2) Unidades de disco:

Las unidades de discos son soportes de acceso compartido, ya que distintos usuarios pueden estar utilizándolos aparentemente al mismo tiempo, en consecuencia, la capacidad de almacenamiento debe ser convenientemente administrada para lograr un uso eficiente.

¹⁰ Peterson, J. L., Silberschatz, A., *Sistemas Operativos: Conceptos fundamentales*, Editorial Reverté S. A., Barcelona, España, 1989, pág. 17.



El problema básico de la administración de los espacios en las unidades de discos está dado por la asignación del espacio a los archivos. La lectura o modificación de los registros de un archivo no implica problemas de administración de espacio, ya que en el directorio se encuentran los datos necesarios para conocer donde se encuentra el archivo.

- c. Independencia de Unidades:** Esta es una característica de algunos sistemas operativos por lo cual los programas son independientes de las unidades físicas. Esto quiere decir que el programa no hace referencia a una unidad periférica en particular sino a un tipo de unidad periférica. La asignación en este caso podrá ser manual, o sea, indicar en que unidad periférica se grabará o leerá un archivo (unidad de disco o cinta), o en que impresora se emitirán los listados, o si será función del monitor determinar la unidad periférica a utilizar.
- 4. Administrador de Interrupciones / Administrador de los Dispositivos de E/S / Interrupt Handler.**

Este componente también llamado rutina de atención de interrupciones, realiza una de las funciones más importantes del sistema operativo, ya que él es el responsable del manejo de las distintas interrupciones (eventos) que se generen durante la ejecución de los procesos en un sistema de computación:

- Ñ Inicialización o finalización de un proceso de E/S.
- Ñ Finalización del tiempo asignado por la CPU para un programa.
- Ñ Mensaje por ocurrencia de un error.
- Ñ Falta de página de memoria.
- Ñ Pedido de servicio desde un periférico.
- Ñ Mal funcionamiento del hardware.
- Ñ Llamadas voluntarias del componente monitor.
- Ñ Interrupciones externas por teclas especiales.

Existen seis clases de Interrupciones¹¹:

- a. Interrupciones SVC (Supervisor call, llamadas al supervisor / monitor):** Son iniciada por el proceso en ejecución que ejecute la instrucción SVC. Una SVC es una petición generada por el usuario de un servicio particular del sistema, como realizar una operación de E/S, obtener más memoria o comunicarse con el operador del sistema.
- b. Interrupciones de E/S:** Son iniciadas por hardware de E/S. Estas interrupciones le indican a la CPU el cambio de estado de un canal o un dispositivo.

¹¹ Deitel, H. M. (1993); *Introducción a los Sistemas Operativos: Segunda Edición*; Addison-Wesley Iberoamerica S.A.; Wilmington, Delaware, E.U.A.; página 62.



- c. **Interrupciones externas:** Son causadas por diversos eventos, incluyendo la expiración de un ciclo de procesamiento, la pulsación de una tecla o la recepción de una señal procedente de otro procesador en un sistema de multiprocesadores.
- d. **Interrupciones de reinicio:** Ocurre cuando se pulsa el botón de reinicio del sistema o cuando en sistema de múltiples procesadores recibe la señal de reinicio.
- e. **Interrupciones de verificación del programa:** Son causadas por una amplia clase problemas que pueden ocurrir cuando se ejecutan las instrucciones en lenguaje de máquina de un programa. Algunos de los problemas relacionados con esta interrupción pueden ser por ejemplo, la presencia de datos con un formato erróneo, intento de hacer referencia de un espacio de memoria que exceda los límites de la memoria real, intento de hacer referencia a un recurso que esté protegido, etc. Algunos sistemas ofrecen a los usuarios la opción de especificar las rutinas que deben ejecutarse cuando debe ocurrir una interrupción de verificación del programa.
- f. **Interrupciones de verificación de la máquina:** Son ocasionadas por el mal funcionamiento del hardware.

Las clases de interrupciones pueden ser de distinto tipo:

1. HARDWARE:

a. INTERNAS:

- i. Memoria RAM: Errores de direccionamiento (Dirección inexistente, espacio ocupado por otro programa).
- ii. CPU: Decodificación de un código de operación inexistente y por operaciones aritméticas (división por cero).

b. EXTERNAS:

- i. Dispositivos externos¹².

2. SOFTWARE:

- a. **S.O.:** Portabilidad (Más fáciles de usar pero más lentas porque dependen del Bios).
- b. **BIOS:** Velocidad (Son más específicas, su mayor inconveniente es que varían según el tipo de tecnología o fabricante del hardware).

¹² Ver SO-ANEXO II-IRQ



5. Administrador de Trabajos a ser Ejecutados / Administrador de Colas / Scheduler:

Es parte integrante del ejecutivo y su función es la de administrar los requerimientos de un procesamiento, siendo responsable de dos actividades:

- 1)** Asignación de tiempos de CPU a los procesos.
- 2)** Establecimiento y mantenimiento de los trabajos a ser ejecutados que representan a las tareas que están esperando por la asignación de los recursos de la computadora.

Una cola es una estructura de almacenamiento en el cual el primer pedido se satisface primero.

En los sistemas operativos grandes hay una cola de programas esperando por cada recurso, incluyendo a la CPU. En algunos equipos se pueden usar colas adicionales para establecer las prioridades o categorías para las tareas que estén en ejecución, de este modo, a algunas tareas se le asignará preferencialmente recursos que a otras.

La ejecución de un programa tiene 3 requisitos básicos, que tenga la CPU libre, que este primero en la cola de espera y que exista memoria disponible para poder ser ejecutado.

En los equipos grandes, el sistema operativo es el que maneja la ejecución de los requerimientos, cuando no puede satisfacerlos, los almacenará en las unidades de disco correspondientes.

Cuando el monitor procesa una interrupción por finalización de un programa, activa los componentes necesarios para analizar sintácticamente el requerimiento pendiente, lo busca en la biblioteca de ejecutables, si existe alguno, lo trae a la memoria asignada por el administrador de memoria, y el requerimiento es ejecutado, si no existiera ninguno queda pendiente hasta la aparición del primer requerimiento.

En los equipos grandes este componente puede no formar parte del sistema operativo, o sea que puede ser residente o transitorio, y en este último caso es traído a la memoria por el monitor al procesar una interrupción de fin de programa.

En los equipos medianos los programas se ejecutan bajo el control del sistema operativo y necesitan una composición determinada de datos, procedimientos y estructuras de superposición (overlaying) como requerimiento. Generalmente, estos programas se instalan y almacenan en archivos de programas.

Cuando un programa es activado, el ejecutivo separa al programa en segmentos de datos y segmentos de procedimientos, y los carga en las áreas de memoria disponibles. De esta forma un programa compite con otros, mientras un programa se está ejecutando, otros esperan su reactivación.



El orden prioridades de cada programa puede ser asignado externamente. En las mini computadoras, tipo PC, si su uso es con usuario único, este componente no cumple ninguna función, porque el usuario es quién decide el orden de la ejecución de los programas, en el caso que sea multiusuario, la ejecución de los programas se ejecutaran por orden de llegada.

La forma de administrar los trabajos a ser ejecutados por parte de este componente puede ser en serie o en paralelo.

Cuando un sistema operativo realiza los procesos en serie, quiere decir que es un sistema que trabaja con multiprogramación del tipo Batch, en este caso cuenta con una rutina conocida como Job Control (Controlador de trabajos) que planifica la secuencia de los trabajos de forma que varios programas compartan la memoria útil en un momento dado, teniendo cada uno de ellos una partición de memoria asignada y controla la ejecución sucesiva de los mismos atendiendo ciertos niveles de prioridad, recursos de E/S disponibles, tiempos límites asignados a ciertos programas, etc.

El algoritmo que sigue un Job Control para planificar la secuencia de trabajos, generalmente no maneja prioridades, o sea, que los administra por orden de llegada, controla la cola de trabajos en serie y los multiprograma.

En el caso de que este componente administre los trabajos a ser ejecutados en paralelo, estaríamos hablando de que el sistema operativo es de tiempo compartido.

En estos tipos de sistemas generalmente los programas son atendidos con la misma prioridad, se le asigna a cada usuario que este conectado a la computadora principal un tiempo mínimo de procesamiento (milésimas de segundo), teniendo cada uno de ellos la sensación de que están trabajando solos. La CPU sólo ejecuta por vez a un programa, pero dada que la velocidad de rotación de los distintos programas es muy alta, el usuario no percibe un retraso apreciable en la ejecución de su programa.

Existen dos sistemas de cómo compartir los tiempos de procesos, el primero de ellos consiste en dar a todos los usuarios, porciones iguales de tiempo, el segundo sistema consiste en repartir el tiempo en forma distinta con cada programa según el tipo de operaciones que este requiera.

Con el procesamiento paralelo los programas son ejecutados sin abandonar la cantidad de veces el control de ejecución, se dice que los programas avanzan tarea por tarea, de ahí el nombre de la rutina que forma parte de este componente es el de Controlador de Tareas (Task Control).

Como vemos este componente se encarga de administrar las distintas colas de trabajos de los distintos recursos, y aunque en el caso del procesamiento paralelo parecería que no existieran, recordemos que en la mayoría de los casos, cada recurso sólo puede ser utilizado por un programa por vez.



METODOS DE ASIGNACION DE LA CPU

Los distintos tipos de colas de trabajo se generaron a partir de la existencia de varios programas compitiendo por el uso de un mismo recurso, siendo la CPU el recurso más importante a ser administrado por un sistema operativo, existen distintos métodos de asignación de la CPU para multiprogramación que son utilizados por el componente administrador de trabajos a ser ejecutados cuando una interrupción le comunica al sistema operativo la existencia de un evento que debe ser analizado, es por ello que frente a una interrupción, cesa la ejecución de un programa que se encuentra en estado de ejecución pasando al estado de preparado y que una vez procesada la interrupción por el sistema operativo, se continuará ejecutando el programa que corresponda de acuerdo al método de asignación que utilice.

METODO POR CICLOS DE TIEMPO CONSTANTE (Orden de llegada): Con este método, la lista es rastreada a intervalos constantes. En este caso el ordenamiento de la lista no es significativo ya que no existen privilegios. Comienza a ejecutarse el primero que llega. Comenzando con un programa, la lista es rastreada:

- 1) Cuando el programa termina.
- 2) Cuando el programa no puede continuar su ejecución.
- 3) Cuando el contador de tiempos genera una interrupción.

El rastreo de una lista recomienza en el programa siguiente, volviendo al principio, después del último.

Sólo puede ser seleccionado un programa que esté listo para ejecutarse, si todos se encuentra en suspenso, continuará rastreando la lista hasta que se produzca una interrupción de E/S que cambie el estado de un programa, de detenido a preparado.

Este método no permite que una actividad de mucho tiempo de proceso no monopolice en algún momento al procesador.

Este método es aplicable a procesos en tiempo real o donde se mezclan modalidades de procesos en batch o en tiempo real.

METODO DE TRABAJAR HASTA TERMINAR (Hasta Terminar): Este método se caracteriza porque en el ordenamiento de la lista no existen privilegios. Comenzando siempre por el primer programa que requiera ser ejecutado. La lista es rastreada:

- 1) Cuando el programa termina.
- 2) Cuando el programa no puede continuar su ejecución.

A este método se lo conoce comúnmente con el nombre de trabajar hasta terminar ya que un programa cesa su ejecución: cuando la ha terminado, cuando se detiene por alguna causa, por ejemplo: por una operación de E/S, pero **no** por haber terminado su ciclo de tiempo como ocurre en el método por ciclos de tiempo constante.

Este método permite que un programa de mucho tiempo de proceso monopolice en algún momento el uso del procesador a expensas de los programas que tengan mayor cantidad de operaciones de E/S. Este método es aplicable en procesos batch.



METODO DE E/S (Por Prioridades): Este método se caracteriza por el criterio de ordenamiento de la lista de programas, ya que éstos se ordenan de acuerdo a prioridades decididas exteriormente

Es decir que se dará mayor privilegio a los programas con mayor cantidad de E/S y menor tiempo de proceso.

De esta manera, quedará tiempo de CPU para los últimos programas de la lista. Cuando un programa de mayor prioridad queda listo para ejecutarse, éste será el programa que se continuará ejecutando dado que la lista comienza a rastrearse desde el inicio.

Existe una variante de mejoramiento de este método que consiste en efectuar un reordenamiento de la lista de programas realizando la reasignación de prioridades de tal forma que el programa que ha insumido más tiempo de procesador desde el último ordenamiento, tenga la menor prioridad. Este reordenamiento se efectúa en forma automática.

Este método no permite que un programa monopolice el uso de la CPU a expensas de los programas de gran cantidad de operaciones de E/S.

El método de E/S es aplicable en procesos batch y en aquellos donde se mezclan modalidades batch o en tiempo real.

METODO DE CICLOS DE TIEMPO POR NIVELES (Por Privilegios): Con este método, la lista es rastreada a intervalos que no son regulares, comenzando con una actividad:

- 1) Cuando este programa termina.
- 2) Cuando este programa no puede continuar.
- 3) Cuando el contador de tiempos genera una interrupción.

Para la confección de la lista, se determinan varios niveles del **1 a n**, y a cada nivel se lo caracteriza por una duración determinada del ciclo de procesamiento. Es decir, que la lista de programas se encuentra dividida en varios niveles y la duración del ciclo de tiempo es mayor a medida que aumenta el nivel. Por ejemplo, el ciclo de tiempo en el nivel **1** será menor que en el nivel **n**.

Los programas se asignan a los niveles de acuerdo a su tamaño, desde el punto de vista del uso de la CPU.

Este método da prioridad a los programas cortos. Los largos son atendidos con menor frecuencia, pero por períodos más largos.

Este método se adecua las modalidades de procesamiento en tiempo real y de accesos múltiples.

**RESUMEN DE LOS MÉTODOS DE ASIGNACIÓN**

Método	Cuando	Desde donde	Tipo de Cola	Privilegios	Depende
Ciclo de tiempo constante:	1. Fin del Prg. 2. Detención Prg. 3. Interrupción (Fin de ciclo).	Desde el programa siguiente.	Orden de llegada.	No existen.	Sistema Operativo
Ciclo de tiempo por niveles:	1. Fin del Prg. 2. Detención Prg. 3. Interrupción (Fin de ciclo).	Desde el principio de la lista.	Por privilegios.	Por niveles (Asignación automática).	Sistema Operativo
Hasta Terminar:	1. Fin del Prg. 2. Detención Prg.	Desde el programa siguiente.	Hasta terminar.	No existen.	Usuario
E/S:	1. Fin del Prg. 2. Detención Prg. 3. Interrupción (Cambio de Estado).	Desde el principio de la lista.	Por prioridades.	Por niveles definidos externamente.	Usuario

6. Administrador de Datos / File System.

Existen dos funciones del sistema operativo para la administración de los datos:

- a. Manejo de la organización de los archivos.
- b. Manejo de los registros físicos.

a. Manejo de la organización de los archivos:

Si bien los programas establecen la organización de cada archivo, no son las instrucciones las que detallan las actividades necesarias para leer o grabar un registro, tan sólo establecerán que se debe leer o grabar de un registro indicando a que archivo pertenece.

Esta función representa la interfase entre el programa y la rutina que maneja las unidades periféricas a través del procesador del canal. En algunos sistemas operativos es cumplida por el monitor. En otros, existen distintos subprogramas (bibliotecas) destinados a manejar cada organización de archivos.

b. Manejo de los Registro Físicos:

La unidad de transferencia entre las unidades periféricas y la CPU es el registro físico. Sin embargo el programa, cuando lee o graba, lo hace por registro lógico.



Generalmente el registro físico y el registro lógico no coinciden, ya que un registro físico puede contener varios registros lógicos, se necesitará una rutina que se encargue de leer un registro físico y de ir pasándole al programa de a un registro lógico, por cada vez que éste lo requiera y una vez completado el registro físico, solicitar su grabación física. La ventaja de agrupar varios registros lógicos en un registro físico, significan: Ahorro de soporte (el registro ocupa menor espacio) y ahorro de tiempo, debido a que la de un registro búsqueda podrá ser más rápida, ya que habrá varios registros en una misma dirección física.

“Un registro es una unidad natural de datos. Distinguiremos como un registro físico (a una línea de impresión, un carácter de un teclado o bloque en una unidad de almacenamiento) y cómo registro lógico (a una línea en el ingreso de datos, una instrucción o una tabla).

Los registros lógicos están definidos por la aplicación; los registros físicos se definen por la naturaleza del dispositivo de E/S.

La agrupación y desagrupación en bloques, por medio del software, pueden realizar las conversiones entre los registros lógicos de una aplicación y los registros físicos del dispositivo.

Los registros son las unidades de datos utilizados en el buffering”.

Un registro es un área de almacenamiento temporario donde se colocan los datos e instrucciones para poder ser procesados.

Los buffers contienen los registros que ya han sido leídos pero todavía no procesados por el recurso al cual pertenecen, o procesados por la CPU pero que aún no han sido enviados hacia el recurso correspondiente.

La diferencia del uso de buffering con el spooling, es que el buffering puede realizar el intercambio de a un trabajo por vez y el spooling puede realizar el intercambio con varios trabajos a la vez.

7. Librerías / Bibliotecas / Library Manager.

Las librerías son colecciones de programas que pueden ser suministrados junto con el sistema operativo (esto no quiere decir que formen parte) o desarrolladas por los usuarios. Las librerías de sistema constan de tres partes principales: la biblioteca del SISTEMA, que incluye traductores de lenguaje, subrutinas y rutinas utilitarias como el debugger, linker, editor de textos; la librería del USUARIO, esta incluye programas, escritos por el usuario y los archivos de datos; y los DIRECTORIOS, que son índices que le indican al sistema operativo en donde se encuentra almacenado un archivo.

Las librerías no son estrictamente parte del sistema operativo, pero el programa controla su uso es en realidad un componente del sistema operativo que detecta un pedido por un programa de la librería, localizándolo de manera que el administrador de periféricos pueda ser llamado para poder leer o escribir en él periférico que corresponda.



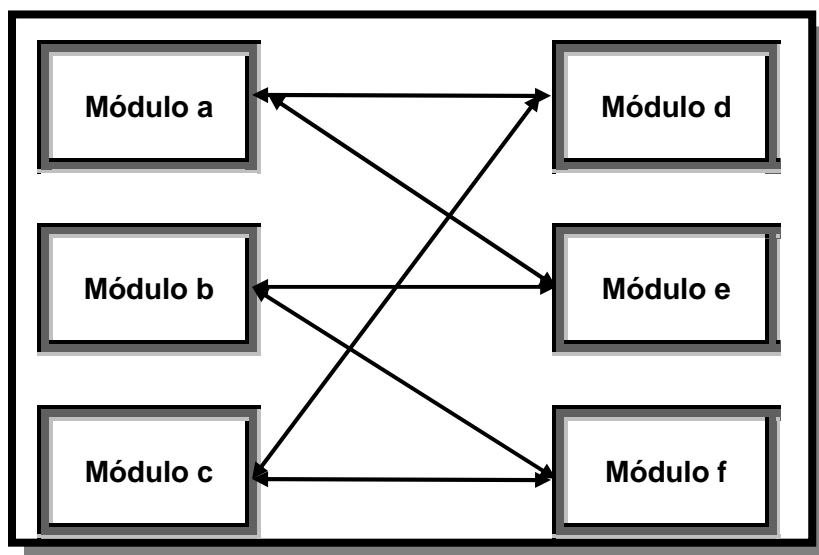
Las librerías cumplen la función de brindarle información a los programas, tanto para su creación como para su ejecución. Una librería puede formar parte de un programa (**internas o estáticas**), o sea que se las incorpora dentro de él, como ocurre con los archivos de encabezado que se utilizan en el lenguaje C., o pueden ser compartidas por varios programas que requieren de información adicional para poder continuar con su ejecución (**externas o dinámicas**), como ocurre por ejemplo con las librerías dinámicas de Windows (.DLL).

TIPOS DE ESTRUCTURAS DEL NÚCLEO ¹³

1. Monolítica:

Esta estructura se identifica con los primeros sistemas operativos que estaban formados fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra.

Estructura Monolítica



Las características fundamentales de este tipo de estructura son:

- Ñ Construcción del programa final en base de módulos compilados separadamente que se unen a través del linkeador.
- Ñ Buena definición de los parámetros de enlace entre las distintas rutinas existentes, que puede provocar mucho acoplamiento (intercambio).
- Ñ Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc.

¹³ Alcalde E., Morera J., Pérez J. A., Campanero (1992), *Introducción a los Sistemas Operativos: Serie: Informática de Gestión*; Aravaca, Madrid; Mc Graw-Hill, Interamericana de España S.A.U...; Capítulo 3, páginas 33 a 35.



- Ñ Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones.

2. Jerárquica:

A medida que fueron creciendo las necesidades de los usuarios y se fueron perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, del sistema operativo, donde una parte del sistema contenía subpartes y esto estaba organizado en forma de niveles.

Se dividió el sistema operativo en pequeñas partes, de tal forma que cada una de ellas estuviera perfectamente definida y con un claro interfase con el resto de elementos.

Se constituyó una estructura jerárquica o de niveles en los sistemas operativos, el primero de los cuales fue denominado THE (Technische Hogeschool, Eindhoven), de Dijkstra, que se utilizó con fines didácticos. Se puede pensar también en estos sistemas como si fueran "multicapa". Multics y Unix caen en esa categoría.

Sistema Jerárquico THE

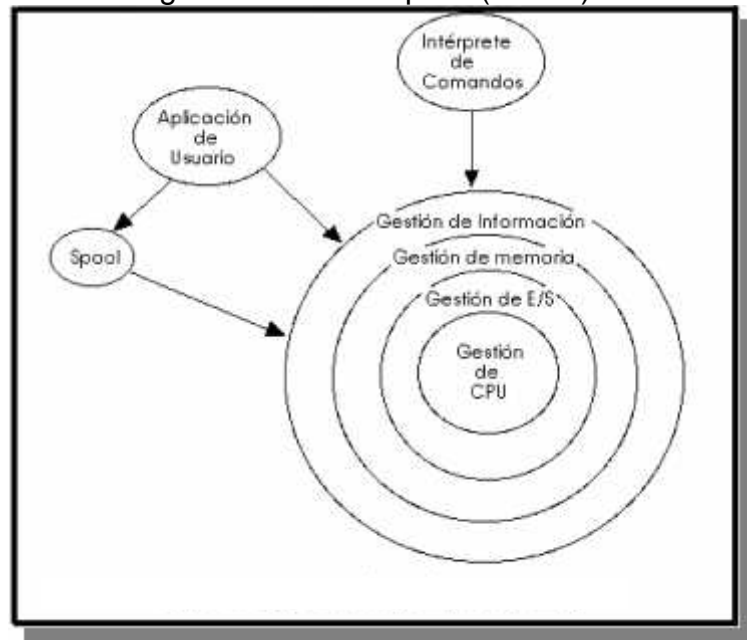
Capa 5	Usuario
Capa 4	Archivos
Capa 3	Entrada / Salida
Capa 2	Comunicaciones
Capa 1	Memoria
Capa 0	Gestión de la CPU
Capa 1	Hardware

En la siguiente estructura se basan prácticamente la mayoría de los sistemas operativos actuales. Otra forma de ver este tipo de sistema es la denominada "de anillos concéntricos".

En el sistema de anillos, cada uno tiene una apertura, conocida como puerta o trampa, por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos indeseados desde las capas más externas. Por lo tanto las capas más internas serán más privilegiadas que las externas.

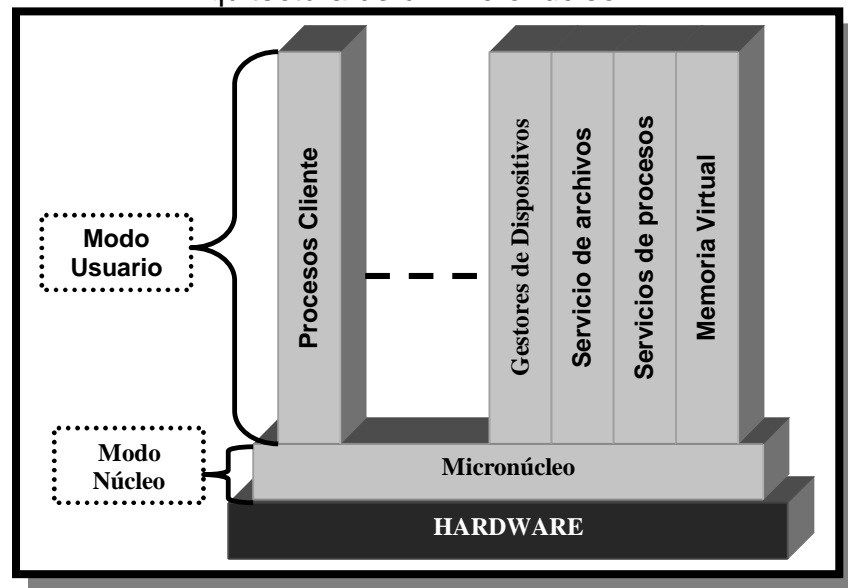


Organización Jerárquica (anillos) ¹⁴



3. Arquitectura Micronúcleo

Arquitectura de un Micronúcleo



La mayor parte de los sistemas operativos, hasta hace poco tiempo, se caracterizaban por un gran núcleo monolítico. Este tipo de estructura del núcleo incluía planificadores, sistemas de archivos, comunicación entre procesos, controladores de dispositivos, gestión de memoria y demás funciones, generales de todo sistema operativo. Normalmente un núcleo monolítico está implementado como un único proceso, con todos sus componentes compartiendo un mismo espacio de direcciones.

¹⁴ <http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/capitulo2.html>

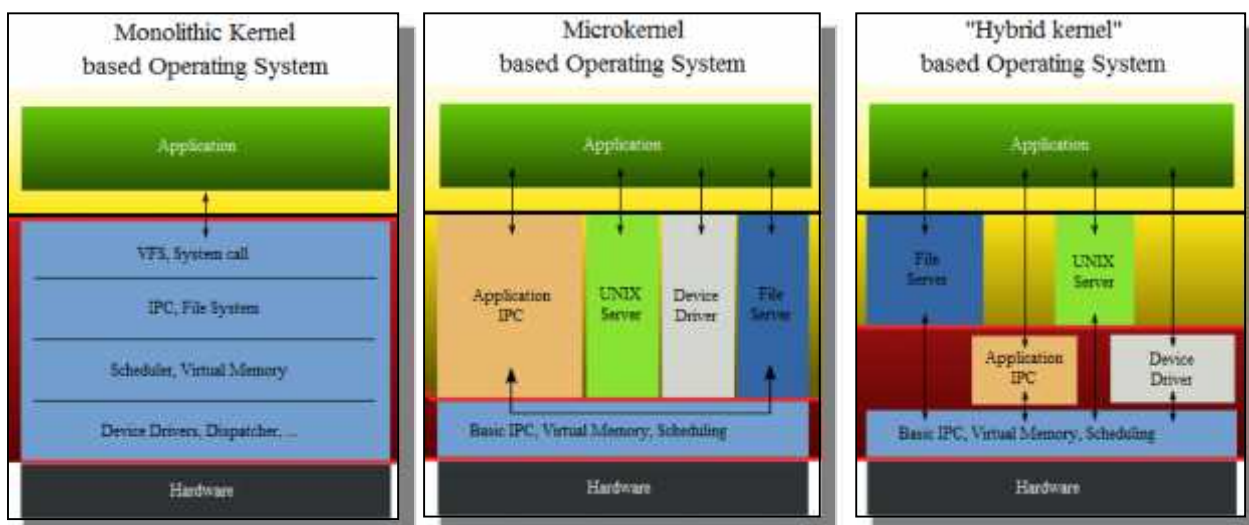


La arquitectura micronúcleo, le asigna las funciones esenciales al núcleo, en donde se incluyen espacios de direcciones, comunicación entre procesos, y planificación básica. Otras funciones del sistema operativo las proporcionan procesos que se ejecutan a modo usuario y que el micronúcleo los trata como a cualquier otra aplicación. Las aplicaciones y los servicios menos importantes se construyen sobre el micronúcleo y se ejecutan a modo usuario, aunque la línea divisoria de lo que esta adentro o afuera del micronúcleo varía de un diseño a otro, una de las características en común es que muchas de las rutinas que tradicionalmente formaron parte del núcleo de los sistemas operativos ahora son subsistemas externos que interactúan con el núcleo y con otros subsistemas, estos incluyen controladores de dispositivos, los sistemas de archivos, el manejo de la memoria virtual, los sistemas de ventanas y los servicios de seguridad.

4. Núcleo Híbrido

Son fundamentalmente una arquitectura micronúcleo que tiene en la parte del núcleo algunas de las rutinas como en la arquitectura monolítica que permite que las funciones del núcleo se ejecuten más rápidamente, dado que no tiene que invocarlas en los espacios de usuario en donde se encuentran almacenados los distintos objetos de la arquitectura micronúcleo.

Tipos de Arquitecturas del Núcleo de los Sistemas Operativos modernos¹⁵



¹⁵ http://es.wikipedia.org/wiki/N%C3%BAcleo_h%C3%ADbrido



CARACTERÍSTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS¹⁶

Con el paso de los años, se ha producido una evolución gradual de la estructura y capacidades de los sistemas operativos. Sin embargo se vienen produciendo nuevos elementos de diseño en los nuevos sistemas operativos debido a la incorporación permanente de nuevas tecnologías a nivel de hardware que se vienen produciendo en algunos de los dispositivos que lo componen, como por ejemplo, computadoras con más de un procesador, los incrementos en la velocidad en los distintos aspectos vinculados con el procesamiento y transferencia de la información entre sus componentes internos, alta velocidad de transferencia de datos en las redes de comunicación, incremento en la capacidad de almacenamiento y variedad en las características de los nuevos dispositivos. A nivel del software, el diseño de los sistemas operativos, están influenciados por las aplicaciones multimediales, el aumento en el uso de Internet y el crecimiento del uso de los sistemas operativos Cliente-Servidor.

El porcentaje de cambios de los sistemas operativos, requiere no solamente de las modificaciones y mejoras en las arquitecturas ya existentes, sino en nuevas formas de organización. Dentro de las variantes que se han implementado podemos encontrar:

- a. Multihilos.
- b. Multiprocesamiento simétrico.
- c. Sistemas operativos distribuidos.
- d. Diseño orientados a objetos.
- a. **Multihilos**

Los multihilos son una técnica por la cual un proceso, ejecutando una aplicación, se divide en hilos que pueden ejecutarse concurrentemente. Esta es una capacidad que poseen los sistemas operativos de mantener varios hilos de ejecución dentro de un mismo proceso. El concepto tradicional que indica que para un sistema operativo un proceso ejecuta un solo hilo, la idea de hilo no existe, dado que cada proceso genera un solo ciclo de proceso. (Ver Procesos).

b. Multiproceso Simétrico

Esta técnica surgió a partir de la existencia de varios procesadores para un sistema de computación que define un tipo de arquitectura del hardware que deberá ser administrada por un sistema operativo que debe permitir que todos los procesadores puedan ejecutar las mismas funciones (de aquí el termino de simétrico), compartir los mismos recursos y estar interconectados por el bus de comunicaciones correspondiente permitiendo de esta forma la ejecución paralela de procesos. (Ver Sistemas Multiusuarios)

¹⁶ Stallings, William (2005); *Sistemas Operativos: Quinta Edición*; Madrid; Pearson Educación S.A.; página 177.



c. Sistemas Operativos Distribuidos

Este tipo de sistemas operativos está asociado a la forma de trabajo de los sistemas Cliente/Servidor que permite la administración individual de cada procesador y el tratamiento simétrico (anónimo) de todos los procesadores que compongan el sistema.

d. Diseño Orientado a Objetos

Esta es una de las características principales de los sistemas operativos actuales que permite agregar nuevos objetos al núcleo sin romper su integridad, dado que cada objeto es independiente.

El concepto fundamental del diseño orientado a objetos es el objeto. Un objeto es una unidad de software independiente que contiene un conjunto de variables (datos) y métodos (procedimientos) relacionados.

Todo objeto representa algo, que puede ser físico o lógico, que contiene variables que expresan la información que se conoce del objeto y métodos que incluyen procedimientos cuya ejecución influyen en los valores del objeto, y posiblemente en lo que este representa.

FORMA DE TRABAJO

1. Máquina Virtual:

Se trata de un tipo de sistemas operativos que presentan una interfaz para cada proceso, mostrando una máquina que parece idéntica a la máquina real.

Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de los sistemas: la multiprogramación y la máquina extendida.

El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.

Máquina Virtual ¹⁷

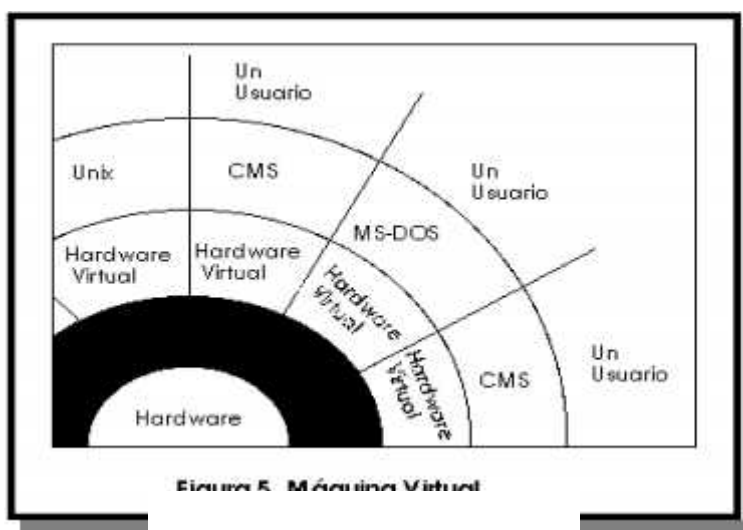


Figure 5. Máquina Virtual

¹⁷ <http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/capitulo2.html>



El núcleo de estos sistemas operativos se denomina monitor virtual y tiene como misión llevar a cabo la multiprogramación, presentando a los niveles superiores tantas máquinas virtuales como se soliciten. Estas máquinas virtuales no son máquinas extendidas, sino una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario.

2. Cliente-Servidor:

Es el tipo más reciente de sistemas operativos, denominado Cliente-Servidor, que puede ser ejecutado en la mayoría de las computadoras, ya sean grandes o pequeñas.



Este sistema sirve para toda clase de aplicaciones por lo tanto, es de propósito general y cumple con las mismas actividades que los sistemas operativos convencionales.

El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser ejecutados tanto en los servidores como en las estaciones de trabajo de los clientes. Por ejemplo, un programa de aplicación que se está ejecutando en un cliente, llama al servidor correspondiente para acceder a un archivo o realizar una operación de E/S sobre un dispositivo determinado. A su vez, un cliente puede actuar como servidor para otro. Este paradigma ofrece gran flexibilidad en cuanto a los servicios posibles en el sistema final, ya que el núcleo provee solamente funciones muy básicas de memoria, E/S, archivos y procesos, dejando a los servidores proveer la mayoría de los recursos, que el usuario final o programador puede usar. Estos servidores deben tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware.

¹⁸ <http://www.geocities.com/SiliconValley/8195/noscs.html>



SERVICIOS QUE OFRECE

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final y tiene en cuenta el número de usuarios (monousuarios y multiusuarios), el número de tareas (monotareas y multitareas) y por el número de procesadores (monoproceso y multiproceso). Este tipo de clasificación permite la selección del tipo de sistema operativo adecuado para un sistema de computación que puede estar formado por una sola computadora o por varias.

Sistemas Operativos por Servicios ¹⁹

Clasificación	Tipos
Cantidad de Usuarios:	Monousuario
	Multiusuario
Cantidad de Tareas:	Monotarea
	Multitarea
Cantidad de Procesadores:	Monoprocesador
	Multiprocesador

1. Monousuarios:

Los sistemas operativos monousuarios son aquellos que soportan de a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales se identifican con estos tipos de sistemas operativos.

2. Multiusuarios:

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones.

No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

3. Monotareas:

Los sistemas del tipo monotarea son aquellos que sólo permiten realizar una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos están compartiendo una misma tarea (aplicación) a la vez.

¹⁹ <http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/capitulo2.html>



4. Multitareas:

Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias tareas al mismo tiempo. Por ejemplo, puede estar editando el código fuente de un programa durante su depuración mientras compila otro programa, a la vez que está recibiendo correo electrónico en un proceso en background (segundo plano). Es común encontrar en ellos interfaces gráficas orientadas al uso de menús y el ratón, lo cual permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

5. Monoproceso:

Un sistema operativo monoproceso es aquél que es capaz de manejar solamente de a un procesador por vez, de manera que si la computadora tuviese más de uno le sería inútil. El ejemplo más típico de este tipo de sistemas es el DOS y Mac OS.

6. Multiproceso:

Un sistema operativo multiproceso se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente. Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá para distribuir la carga de trabajos a los demás procesadores, que reciben el nombre de esclavos. Cuando se trabaja de manera simétrica, los procesos o partes de ellos (subprocesos) son enviados indistintamente a cualquiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

FORMA DE OFRECER SUS SERVICIOS

Esta clasificación también se refiere a una visión externa vinculada con los sistemas multiusuarios²⁰ y a la forma que cada usuario accede a los servicios en un sistema operativo multiusuario. Existen de dos tipos: Punto a Punto (Sistemas Distribuidos) y con Servidor Dedicado (Redes).

1. Punto a punto:

Este tipo de sistema operativo está relacionado con los sistemas distribuidos en donde cada puesto de trabajo debe tener su propio sistema operativo para poder comunicarse y poder compartir los distintos recursos que forman parte de un sistema multiusuario. Todos los sistemas operativos de Microsoft salvo el DOS, cumplen con estas características, dada su forma de trabajo que permiten que un puesto de trabajo pueda trabajar de forma dependiente o no del computador principal. IBM tiene en las distintas versiones del OS/2 esa misma posibilidad al igual que Linux y de las distintas versiones de los sistemas operativos de Apple. Los sistemas operativos para grupos de trabajo y los sistemas Cliente-Servidor son los típicos sistemas multiusuarios que se los identifican como las redes Windows.

²⁰ Ver SO-ANEXO VI-SISTEMAS MULTIUSUARIOS



2. Con servidor dedicado:

Este tipo de sistema operativo se relaciona directamente con los sistemas de red (redes reales) en donde la existencia de un computador principal es obligatoria dada la modalidad de trabajo de los distintos puestos, en donde todas sus actividades dependerán exclusivamente del servidor. Los puestos de trabajo podrán ser inteligentes (poseen recursos propios) o no (no poseen recursos propios), si son inteligentes dependerá de las aplicaciones su dependencia, sino lo son, al no poseer recursos propios no podrán realizar ningún tipo de actividad. En este caso el sistema operativo que debe estar instalado en el computador principal debe reunir las condiciones necesarias como administrador de recurso, en este grupo encontramos por ejemplo a Unix, Linux, Novell, las versiones de Windows NT, 2000, 2003, XP y Vista.

Existen sistemas operativos que pueden trabajar en cualquiera de los sistemas multiusuarios existentes, la variedad surge a partir del tipo de configuración que posea dicho sistema.

En los sistemas multiusuarios el concepto de multiprocesamiento y multiprogramación están relacionados directamente con el tipo de sistema, en un sistema de red, al existir una dependencia de los puestos de trabajo con respecto al servidor, la existencia de multiprogramación es un estándar dentro de su modalidad de trabajo, en cambio en los sistemas distribuidos, el multiprocesamiento es una de sus características principales.

MULTIPROGRAMACION Y MULTIPROCESAMIENTO

La ejecución de varios programas, tiene potencial significado para mejorar el rendimiento total del sistema y la utilización de recursos con respecto a los procesos en **serie** y en **paralelo**. Este potencial se utiliza mediante una clase de sistemas operativos que multiplique los recursos de la computadora al trabajar con varios programas activos, estos sistemas operativos son los que trabajan con varios programas (**multiprogramación**) y el que trabaja con varias computadoras (**multiprocesamiento**).

MULTIPROGRAMACIÓN

La **multiprogramación** consiste en el manejo casi simultáneo de dos o más programas independientes, intercalando o empalmando su ejecución y compartiendo sus tiempos.

El control de la ejecución de la intercalación o empalme, lo realiza el sistema operativo, siempre que estos adopten esta modalidad de trabajo.

Significa entonces que por medio de la **multiprogramación**, se efectúa la administración en paralelo de dos o más programas que residen simultáneamente en la memoria de la computadora.



Características generales:

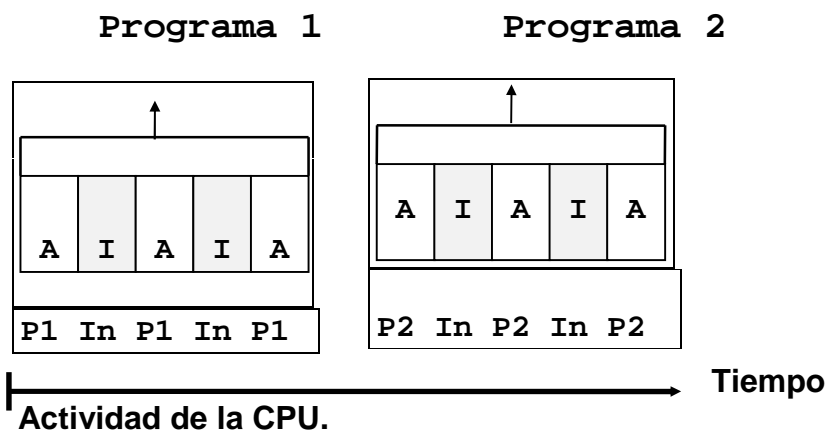
- **INTERCALACION:** Es cuando más de un programa se encuentra cargado en la computadora en condiciones de **ejecutarse, ejecutándose o demorados**, pero todos compartiendo tiempos de procesador y asignaciones de memoria.
- **INSTANTANEIDAD:** Porque simula trabajar, como si existiera un sólo programa cargado, de ejecución instantánea.
- **INDEPENDENCIA:** Se trata de distintos programas, con distinta asignación de memoria y de dispositivos de E/S.

De esta forma con reglas preestablecidas y dando las prioridades de los distintos programas se puede optimizar la aplicación en una computadora.

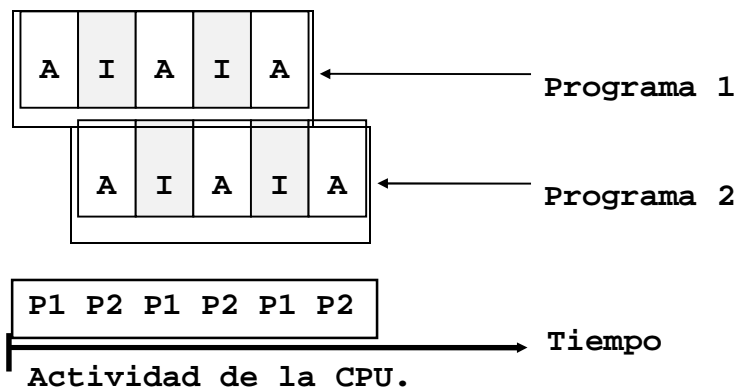
Con la ejecución intercalada de programas, o **multiprogramación** como se le llama a este modo de operaciones, puede producir ganancias significativas de rendimiento. Si realizaríamos la ejecución de los procesos en forma secuencial la CPU estaría inactiva por períodos de tiempo, aunque existiera una cola de espera de los procesos a ser ejecutados.

Por ejemplo:

a) Ejecución secuencial



b) Ejecución Multiprogramada





El objetivo primario de la multiprogramación es el aprovechamiento de la computadora, permitiendo, a través del software de base que lo soporta, que varios programas o diferentes aplicaciones se estén ejecutando intercalándose compartiendo tiempos. Paralelamente a este objetivo se debe desarrollar otro, cual es evitar la interferencia mutua entre dichos programas o aplicaciones. El cumplimiento de este último objetivo se encuentra en estrecha relación con las direcciones de memoria asignadas y las disponibles.

La multiprogramación se puede plantear de dos formas diferentes, la real cuando existe un servidor (red real) y todas las aplicaciones se ejecutan desde, o virtual, cuando las aplicaciones de un mismo sistema se pueden ejecutar en los distintos puestos de trabajo (sistema distribuido).

MULTIPROCESAMIENTO

Hasta aquí hemos desarrollado el tema, indicando la posibilidad de trabajar compartiendo tiempo de proceso y casi en paralelo, efectuando el aprovechamiento de la CPU con varios programas y un sólo procesador.

Si queremos ampliar el concepto pensando en un trabajo real en paralelo y a una ejecución simultánea de programas nos obliga necesariamente a utilizar más de un procesador. De manera que el problema consistirá en asignar una cantidad de procesadores n a una cantidad de programas m , donde generalmente $n < m$, pero sabiendo que $n > 1$. En este caso estamos en presencia del **multiprocesamiento**. El problema consistirá, cómo en **multiprogramación**, en la asignación de un procesador a los distintos programas sabiendo que contamos con más de un procesador.

Cuando un programa termina o se detiene, uno de los procesadores asociados queda disponible y la lista de programas se podrá rastrear de acuerdo a las características que posea el sistema operativo con el que estamos trabajando.

Estos sistemas comparten además de la CPU, la memoria y los periféricos. Se pueden utilizar de diferentes maneras, la más común, es asignar a cada procesador una tarea específica, un procesador central controla el sistema, y los demás tienen tareas específicas, o le piden instrucciones al procesador principal. Este esquema define una relación maestro / esclavo (Ver la clasificación de los sistemas operativos por la forma de trabajo **Cliente–Servidor**).

El multiprocesamiento en los sistemas multiusuarios se lo relaciona con los sistemas distribuidos. En éstas, varias computadoras independientes pueden comunicarse, intercambiar archivos e información y cada computadora tiene su propio sistema operativo, y realiza sus operaciones independientemente.

Las características generales de este tipo sistemas corresponden a las características generales de todos los sistemas operativos y a las que se relacionan con la multiprogramación que plantea como características la instantaneidad y la independencia de los procesos.

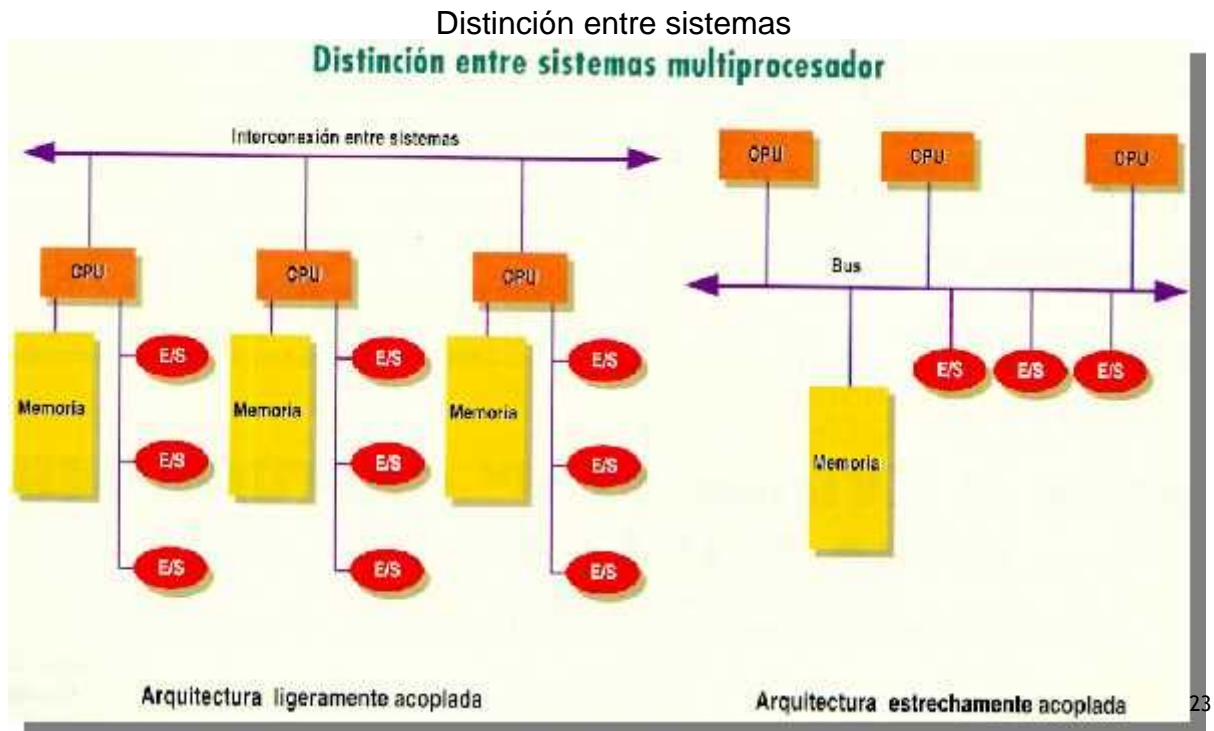


SISTEMAS DISTRIBUIDOS

Existen dos tipos de sistemas distribuidos que sus principales diferencias pasan exclusivamente por el tipo de comunicación entre procesos²¹, estos son el sistema de memoria compartida (fuertemente acoplado) y el sistema de mensajes (débilmente acoplado)²².

Memoria compartida (multiprocesamiento fuertemente o estrechamente acoplado): Utiliza un almacenamiento único compartido por los diferentes procesadores y un sólo sistema operativo que controla los procesadores y el hardware del sistema, precisa que los procesos compartan algunas variables. Los programadores de los sistemas de aplicación deben facilitar la comunicación entre procesos, el sistema operativo sólo tiene que permitir la utilización de memoria compartida.

Sistema de mensajes (multiprocesamiento débilmente o ligeramente acoplado): Implica conectar dos o más sistemas de computación independientes mediante un enlace de comunicación. Cada sistema tiene su propio sistema operativo y su almacenamiento. Los sistemas pueden funcionar en forma independiente y se pueden comunicar entre sí, si es necesario. Cada sistema puede tener acceso a los archivos de otros, y en algunos casos los sistemas pueden enviar tareas a los procesadores que no estén muy cargados para equilibrar la carga de trabajo. La comunicación entre los procesadores se realiza por medio de la transferencia de mensajes y la responsabilidad de facilitar la comunicación es del sistema operativo.



²¹ Ver PROCESOS, página 94.

²² Deitel, H. M. (1993); *Introducción a los Sistemas Operativos: Segunda Edición*; Addison-Wesley Iberoamerica S.A.; Wilmington, Delaware, E.U.A.; página 340.

²³ <http://www.consulintel.es/html/Tutoriales/Articulos/risc.html>



Estos dos sistemas podrían utilizarse conjuntamente. Un sistema de memoria compartida está orientado a una aplicación específica, en cambio la función de un sistema de mensajes consiste en permitir que los procesos se comuniquen sin necesidad de compartir variables.

CLUSTER (AGRUPACIONES)

Una de las áreas más recientes de desarrollo de los sistemas operativos para los sistemas distribuidos esta asociada a creación de agrupaciones que permite que uno o más sistemas de computación interconectados trabajen juntos creando la ilusión de ser una única computadora.

Existen cuatro beneficios que se pueden lograr con los cluster, estos beneficios también se los puede considerar como objetivos o requisitos para su diseño:

Escalabilidad absoluta: Es posible crear un gran cluster que supere incluso la potencia de la mayor de las máquinas. Un cluster puede tener decenas o incluso centenas de máquinas, cada una de ellas en un procesador.

Escalabilidad incremental: Un cluster se configura de tal manera que sea posible añadir nuevos sistemas al cluster en pequeños incrementos. De esta forma, un usuario puede comenzar con sistema pequeño y expandirlo según sus necesidades, sin tener que hacer grandes actualizaciones en las que un pequeño sistema debe ser reemplazado por uno mayor.

Alta disponibilidad: Ya que cada nodo del cluster es una computadora en sí mismo, el de uno de los nodos no significa perdida del servicio. En muchos productos, el sistema operativo maneja automáticamente la tolerancia a fallos.

Relación precio/prestaciones: A través del uso de bloques de construcción es posible hacer un cluster con igual o mayor poder de procesamiento que una gran máquina, con mucho menor costo.

Un cluster es una red LAN en donde cada computadora puede trabajar de forma independiente. Cada computadora tiene instalada una capa de software middleware²⁴ que permite la operación del cluster. Esta capa del sistema operativo proporciona una única imagen del sistema al usuario, independientemente del puesto de trabajo que este utilizando.

Los clusters proporcionan los siguientes servicios o funciones:

- **Un único punto de entrada:** El se identifica para el cluster no para una computadora.
- **Una única jerarquía de archivos:** Los usuarios ven una única estructura de directorios.
- **Un único punto de control:** Existen un servidor.

²⁴ Conjunto de servicios que ofrecen los sistemas operativos que permiten que las aplicaciones funcionen en una red. Esto incluye a los protocolos y las librerías API (Interfaz de Programas de Aplicación) propias del sistema operativo que permite que las aplicaciones invoquen sus servicios.



- **Una única red virtual:** De un puesto de trabajo se puede acceder a cualquier otro del cluster.
- **Un único espacio en memoria:** La memoria compartida distribuida permite que los programas compartan las mismas variables.
- **Un único sistema de control de trabajos:** Esto permite que los programas de usuarios puedan ejecutarse en cualquiera de los puestos de trabajo sin necesidad de indicar en cual se realizará.
- **Una única interfaz de usuario:** Todos los puestos de trabajo tendrán el mismo tipo interfaz.
- **Un único espacio de E/S:** Se pueden compartir los recursos sin necesidad de saber cual es su ubicación física.
- **Un único espacio de procesos:** Todos los procesos tendrán un identificador común que permitirán que la comunicación entre procesos se pueda realizar entre los puestos de trabajo.
- **Puntos de control:** Esta función permite salvar el estado de un proceso y los resultados que este genere, para permitir de esta forma recuperarlos después de un fallo del sistema.
- **Migración de procesos:** Esta función permite una buena distribución de los datos a procesar.

Los últimos cuatro puntos de la lista aumentan la disponibilidad de los cluster, los restantes sirven para proporcionar una imagen única del sistema.

DIFERENTES SISTEMAS OPERATIVOS FUNCIÓN DE LOS COMPONENTES

1. Sistemas Operativos Monousuario

Los usuarios no están compitiendo por los distintos recursos de la computadora. Los componentes tienen las siguientes responsabilidades:

- Ñ **Monitor:** Coordina todas las acciones del sistema operativo y carga y arranca los programas pedidos por el usuario.
- Ñ **Scheduler:** Normalmente no existe dado que un usuario en la terminal toma todas las decisiones tales como que programa quiere y en que orden se ejecutan.
- Ñ **Interrupt Handler:** Permanece del mismo tamaño independientemente del número de usuarios que haya. El pedido sobre el interrupt handler es una función de la configuración del hardware y no del sistema operativo.



- Ñ **Device Handler:** Uno por cada uno de los periféricos. Son programas relativamente estándar de un sistema operativo a otro. Cuando comienza un device handler se le indica cuantas palabras de información se transferirán, la dirección del buffer en memoria, la ubicación del periférico (si corresponde) y en que sentido se realizará la transferencia. El device handler realiza la transferencia y devuelve el control al ejecutivo.
- Ñ **Storage Allocator:** Es relativamente simple dado que el mapa de memoria siempre mostrará al sistema operativo y un programa del usuario. El usuario tiene asignada toda la memoria no usada por el sistema operativo. En algunos sistemas se incluye la capacidad de overlaying para manejar programas muy grandes.
- Ñ **Library Manager:** Administra archivos que contienen programas y datos y están todos disponibles para el usuario del sistema.

La principal ventaja es que tiene una operativa rápida y la mayor desventaja es el ineficiente uso del tiempo de CPU comparado con los sistemas operativos más grandes y complejos mientras el programa del usuario realiza una operación de E/S la CPU está muy desocupada dado que trabaja mucho más rápidamente que cualquier periférico. El tiempo ocioso de la CPU se reduce en sistemas operativos más grandes a través de la capacidad denominada multiusuario o multiprogramación.

La multiprogramación permite que dos o más programas sean ejecutados concurrentemente por la CPU. Cada programa corre en áreas de memoria separadas llamadas particiones y usan a la CPU alternativamente.

Sólo un programa se ejecuta en cualquier instante dado, porque solamente hay una CPU. Pero mientras un programa está esperando por un periférico otro programa puede ejecutarse, de esta forma prácticamente la CPU y los periféricos nunca están libres y frecuentemente, ambos están activos al mismo tiempo en consecuencia la multiprogramación permite un eficiente uso de los recursos de la computadora.

2. Sistemas de Tiempo Compartido

Los sistemas operativos de tiempo compartido multiprograman para varios usuarios en línea. Los recursos de la computadora están compartidos concurrentemente a través de los usuarios permitiendo a cada usuario interconectarse con el sistema como si él fuera el único usuario.

El sistema operativo administra automáticamente los recursos comunes. En estos de sistemas sus componentes son responsables de:

- Ñ **Monitor:** Es más complejo que un sistema de usuario único, la diferencia está dada en que ahora hay muchos usuarios que coordinar. Este carga y arranca los programas para cada usuario en respuesta a comandos desde el teclado.



- Ñ **Scheduler:** Es mucho más complejo que en un sistema de usuario único, todos los programas de usuario normalmente llamados tareas se les permite ejecutar durante un período máximo denominado tiempo de proceso. Al final del tiempo de proceso o cuando la tarea requiere una operación de E/S el programa es retenido y se ejecuta otro. El estatus del programa que se ha interrumpido se salva para permitir su continuación cuando llegue su turno de ejecución. La determinación de la finalización del tiempo que tenga asignado se logra mediante un reloj interno que interrumpe a la CPU y señala la finalización del período. Algunos sistemas operativos de tiempo compartido permiten niveles de prioridad de manera de poder dar a algunos usuarios o programas acceso preferencial tanto a la CPU como a los periféricos.
- Ñ **Storage Allocator:** Es más complejo. La existencia de muchos usuarios implica un mapa de memoria mucho más complejo y los usuarios deben estar protegidos. Adicionalmente la mayoría de los sistemas operativos de tiempo compartido permite más programas de usuarios de los pueden ser almacenados en memoria al mismo tiempo, de esta forma la técnica de intercambio swapping es muy usada. En consecuencia, un programa de un usuario podrá residir en memoria solamente cuando está siendo ejecutado, en cualquier otro momento podrá ser almacenado en un medio de almacenamiento auxiliar esperando por su próximo turno.
- Ñ **Library Manager:** En un sistema de tiempo compartido no solamente provee el acceso a archivos en los medios auxiliares de almacenamiento sino que además limita el acceso a los archivos a los usuarios autorizados. Esto requiere que los usuarios ingresen un número de identificación de usuario y una password que pueden ser verificados contra la restricción de acceso a cada archivo. Estas restricciones normalmente están registradas en el directorio de archivos junto con el nombre de cada archivo y alguna otra información.
- Ñ **Interrupt y Device Handler:** Básicamente son similares a los de un sistema de Usuario único. Los sistemas operativos de Tiempo Compartido están diseñados para operaciones en línea.

Ventajas:

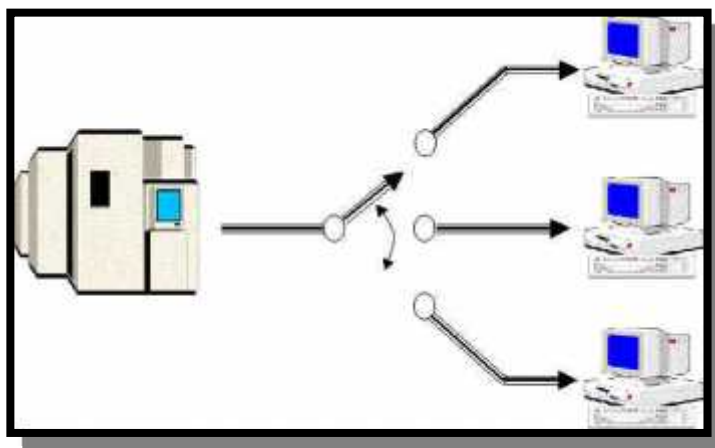
- 1) Cada usuario tiene el beneficio de una computadora dedicado por una fracción del costo.
- 2) Si los sistemas de hardware y software son adecuados, el usuario no se entera de la existencia de los demás usuarios.
- 3) Un eficiente uso de la CPU y otros recursos.
- 4) Los programas pueden ser desarrollados interactivamente.
- 5) Los archivos de datos pueden ser actualizados en línea.



Desventajas:

- 1) Una parte importante de la memoria principal es utilizada por el ejecutivo y otras partes del sistema operativo.
- 2) Cada programa tomará más tiempo de ejecución que en un sistema de usuario único.
- 3) Se requiere una mayor inversión para cosas tales como CPU más rápidas, múltiples terminales para varios usuarios, más memoria principal y dispositivos de almacenamiento auxiliar más grandes y rápidos.

Sistema Operativo de Tiempo Compartido ²⁵



3. Sistemas de Tiempo Real

Los sistemas de tiempo real están diseñados para interactuar con máquinas, las cuales operan más rápidamente que la gente en una terminal. Ciertas aplicaciones tales como procesos de control industrial o monitoreo de pacientes en terapia intensiva utilizan este tipo de sistemas.

Estos sistemas están diseñados para proporcionar una rápida respuesta a programas o dispositivos de cierta prioridad. El término tiempo real (real time) significa que el sistema responde a eventos externos durante el tiempo en que ellos están ocurriendo realmente. Esto es, la computadora puede procesar información y responder lo suficientemente rápido para controlar eventos mientras ellos están ocurriendo.

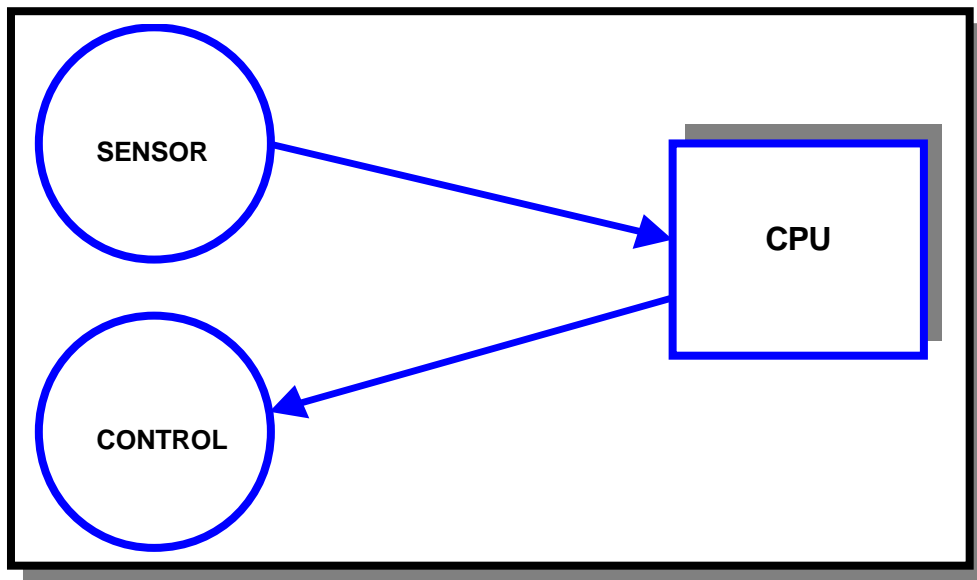
Los sistemas de tiempo real permiten que más de un programa pueda operar concurrentemente. En este caso, las prioridades de los programas son definitivamente establecidos para indicar que programa recibe la más rápida respuesta.

²⁵ <http://itver.edu.mx/so1/clas.html>

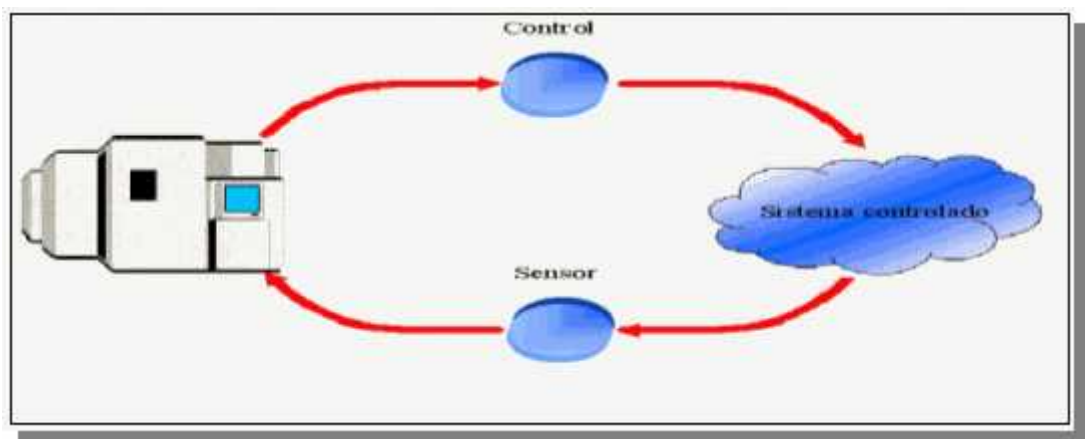


“Un sistema de tiempo real se utiliza frecuentemente como dispositivo de control en una aplicación especializada. Unos sensores proporcionan datos al ordenador. El ordenador tiene que analizar los datos y ajustar en lo posible los controles para modificar las entradas del sensor”.

Sistema de Operativo de Tiempo Real



Las responsabilidades de los componentes del sistema operativo de tiempo real son similares a aquellas de tiempo compartido excepto que el manejo de interrupciones y atenciones de periféricos es mucho más crítica.



De esta forma se tiene un gran cuidado para asegurar que el Interrupt Handler y los Device Handler sean eficientes y permanezcan residentes en memoria en todo momento. Por ejemplo: monitoreo de instrumental medico, proceso de control industrial, etc. La principal ventaja es la rápida respuesta a estímulos externos.



- Ñ **Monitor:** Multiprograma las tareas que atiende otras máquinas.
- Ñ **Scheduler:** Planifica la ejecución basada en prioridades que han sido dadas de acuerdo a la naturaleza del tiempo crítico de la máquina que será atendida por el programa o tarea.
- Ñ **Storage Allocator:** Hace intercambio de tareas que no son de tiempo crítico y asigna espacio permanente a las tareas que necesitan iniciarse en menos tiempo del que se necesita para reingresar un nuevo programa en memoria.
- Ñ **Library Manager:** Opera rápidamente para referenciar archivos y hacer que la información este disponible.