

TEMA 1.- INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

INDICE:

1. ¿Qué es un sistema operativo?	
1.1.- Introducción.....	Pág.2
1.2.- Visión usuario del sistema operativo.....	Pág.3
1.3.- Visión programador del sistema operativo.....	Pág.3
2. Historia y evolución de los SO	
2.1.- 1º Generación: Tubos de Vacío y Paneles de conexión.....	Pág.4
2.2.- 2º Generación: Transistores y Sistemas Batch.....	Pág.4
2.3.- 3º Generación: Circuitos integrados y multiprogramación.....	Pág.6
2.4.- 4º Generación: Ordenadores personales.....	Pág.7
3. Componentes de un SO	
3.1.- Administración de procesos.....	Pág.8
3.2.- Gestor de Memoria.....	Pág.8
3.3.- Sistemas de ficheros.....	Pág.9
3.4.- Administración de E/S.....	Pág.9
4. Estructura de un SO	
4.1.- Estructura monolítica.....	Pág.9
4.2.- Estructura jerárquica.....	Pág.9
4.3.- Estructura Cliente-Servidor.....	Pág.10
4.4.- Estructura orientada a objetos.....	Pág.11
5. Tipos de SO	
5.1.- Según la utilización de recursos.....	Pág.12
5.2.- Según la interactividad.....	Pág.12
5.3.- Según el número de usuarios.....	Pág.12
5.4.- Según el tipo de aplicaciones.....	Pág.13

BIBLIOGRAFIA:

- [MORERA'95] , capítulos 1 y 2
- [TANENB'98], capítulo 1
- [STALLI'97], capítulo 2

1.- ¿QUÉ ES UN SISTEMA OPERATIVO?

1.1 INTRODUCCIÓN

TAREAS DEL SISTEMA OPERATIVO

- Poner una capa de software por encima del hardware para ahorrar (es amigable).
- Aprovechar el hardware lo máximo posible.

El Sistema Operativo es una capa de software sobre el hardware que se encarga de gestionar todos los elementos del sistema y que presenta al usuario una interface (o máquina virtual) más fácil de entender y de programar.

Tabla 1 Definición de Sistema Operativo

Para ubicar el sistema operativo, hacemos una distribución del hardware y del software:

En la parte más baja ubicaremos el hardware donde: en primer lugar, tenemos dispositivos físicos, microprogramables (cargados en ROM, se encargan de traducir el lenguaje básico), lenguaje máquina. En la parte más alta ubicamos el software: el sistema operativo, justo encima encontramos compiladores, editores e interpretes de comandos, y por último procesadores de textos, juegos, fotos, vídeo, etc..

PROCESADOR DE TEXTOS	JUEGOS	GRÁFICOS	ETC...	SW APLICACIÓN
COMPILADORES	EDITORES	INTERPRETE DE COMANDOS		SW
SO				SISTEMA
LENGUAJE MÁQUINA				HW
MICROPROGRAMAS (ROM)				
DISPOSITIVOS FÍSICOS				

Tabla 2 Distribución del hardware y de software

FUNCIONES DEL SISTEMA OPERATIVO

- + Hacer de interface con el usuario.
- + Facilitar la E/S (facilita la comunicación con los periféricos).
- + Permitir compartir el HW y los datos (un servidor deja compartir su hw y sus datos con otros pc's).
- + Protección (por ejemplo, que un usuario borre los datos de otro).
- + Planificar la distribución de recursos, es decir, debemos distribuir el procesador para que varios usuarios puedan trabajar con el simultáneamente, la planificación es cómo se reparten los recursos.
- + Recuperarse de los errores.
- + Organización de datos (en el sentido de localizar rápidamente la información)
- + Manejo de comunicaciones de la red.

Según las funciones del so, podemos hacer una clasificación de sistemas operativos:

- a) Como Administrador de recursos.
- b) Como Protección.

c) Como Máquina virtual, que ofrece un interface.

OBJETIVOS DE SISTEMA OPERATIVOS

- 1.- Facilitar el trabajo al usuario.
- 2.- Repartir correctamente los recursos.

SO MULTIUSUARIOS Y MULTITAREA

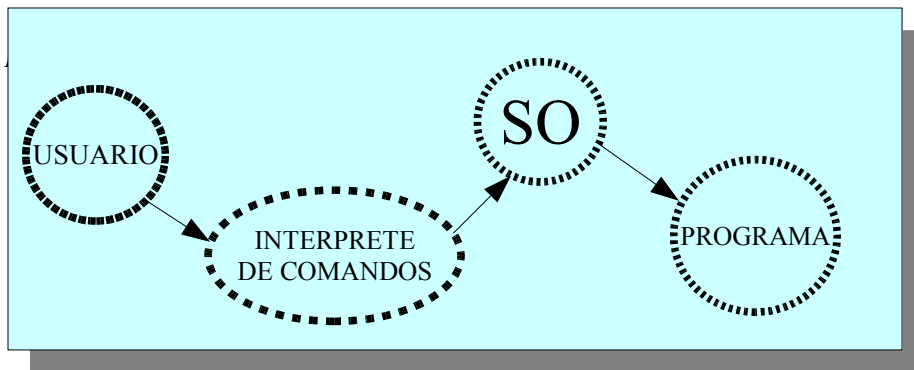
Multiusuario: Permite que varios usuarios estén al mismo tiempo trabajando con el SO.

Multitarea: Es capaz de hacer varias tareas a la vez.

1.2 VISIÓN DEL USUARIO

El usuario del SO ve al SO a través del interprete de comandos.

Interprete de comandos: es una capa que se coloca justo encima del SO y que sirve de traductor entre las órdenes del usuario y las acciones que entiende el ordenador.



1.3 VISION DEL PROGRAMADOR

Si queremos hacer programas y compilarlos, ES NECESARIO el SO, ya que, cuando programamos, hay una serie de instrucciones que son necesarias y no tienen que estar en el programa usado para programar (por ejemplo: internamente el 'Printf' llama a una orden 'write', del so, que si que sabe como sacar las cosas por pantalla, es decir, es interpretada por el so).

Las órdenes de SO (básicas) se llaman System Calls (S.C) o llamadas al sistema

Conseguiremos dos cosas:

- *Independencia del hardware*: cuando un programa fuente se puede ejecutar en varios SO. También es llamado *Portabilidad del software*.

2.- HISTORIA Y EVOLUCIÓN DE LOS SO

Vamos a ver como ha ido evolucionando el hardware.

Hasta los años 40 no se empezó a introducir el tratamiento de la información.

2.1.- 1ª GENERACIÓN: TUBOS DE VACÍO Y PANELES DE CONEXIÓN (1945 - 1955).

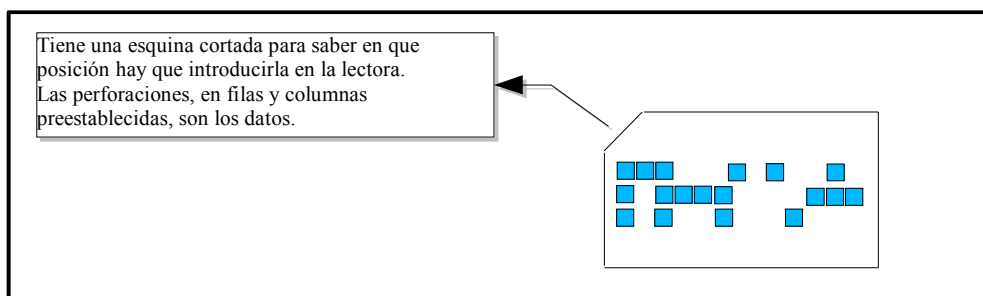
El desarrollo viene dado por la necesidad de, en la 2ª Guerra Mundial, descifrar códigos del bando enemigo.

Estas máquinas estaban compuestas por tubos de vacío, los cuales formaban la circuitería, junto a los paneles de conexión. Los paneles de conexión se programaban de una forma diferente a la actual; al ser los ordenadores inmensamente grandes, se tenía que entrar en la sala y puentear las conexiones manualmente para conseguir el resultado deseado. A la hora de detectar un error se debía repasar las conexiones una a una. Esta tarea, lógicamente, requería los conocimientos del creador de la máquina, por lo que, tanto el usuario como el creador eran, necesariamente, la misma persona.

Este método de programación antes descrito se denomina, procesamiento en serie (ya que las tareas se hacen una detrás de otra).

En tiempos de paz, estas máquinas, básicamente, se dedicaban a realizar operaciones complejas del tipo seno y coseno. Pero trabajar usando estas máquinas presentaba muchos problemas, como son: Equipamiento costoso y tareas mal desarrolladas, lo que provoca un pésimo aprovechamiento del trabajo.

La exigencias dieron pie a una mejora. Esta mejora fue la aparición de las Tarjetas Perforadas (1º dispositivo de entrada)



Dibujo 1 Tarjeta perforada

Ahora los programas son leídos por una lectora de tarjetas perforadas, para luego ser volcados en memoria, que funciona junto con unos datos que también introducidos por tarjetas perforadas. Por lo que nace el cargador, es decir el primer software de sistema.

CARGADOR: Módulo que recoge la información del lector de tarjetas y la vuelca a la memoria, es decir, ofrece unas rutinas de E/S que controlan los periféricos y las tarjetas perforadas. (Nos encontramos ante el principio de los SO).

2.2.- 2º GENERACIÓN: TRANSISTORES Y SISTEMAS BATCH (1955 - 1965).

La segunda Generación comienza con la aparición del transistor. Se empiezan a crear máquinas más fiables, de esta forma, separamos al fabricante, del usuario, y del mantenimiento.

Comienza la venta de ordenadores a Gobiernos, multinacionales, etc.

Aparecen los primeros lenguajes de alto nivel, como es FORTRAM.

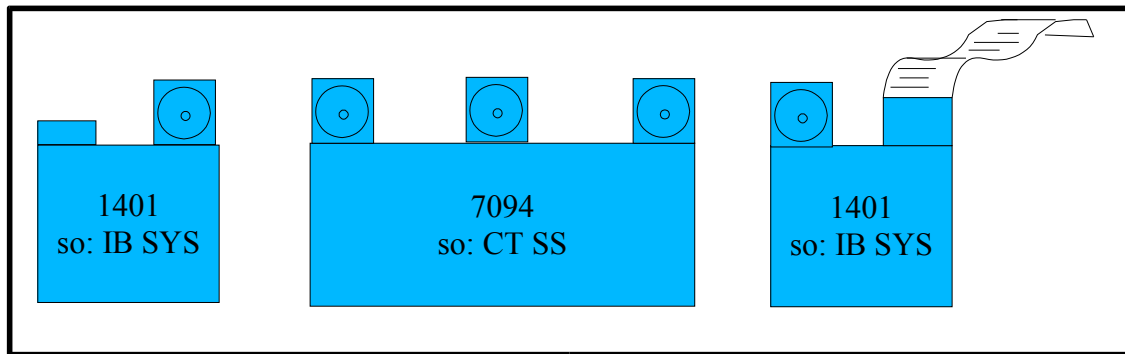
Se automatizan los procesos.

Todo esto, aun así, sigue teniendo un procesamiento en serie muy lento, con la máquina mucho tiempo parada, por lo tanto ofrecían un rendimiento muy bajo.

PROCESAMIENTO POR LOTES y SISTEMAS BATCH: se originaron a causa del bajo rendimiento que ofrecían los ordenadores. Básicamente consistía en agrupar un conjunto de trabajos similares para ejecutarlos conjuntamente, y perder así el menor tiempo posible.

Se ideó este sistema para aprovechar el rendimiento:

Utilizamos 3 máquinas distintas; dos de bajo coste y una grande de mucha capacidad de cómputo. A la primera de las pequeñas le colocamos un lector de tarjetas y una unidad de cintas de salida, a la grande le colocamos tres unidades de cinta, dos de entrada y una de salida, y por último, en la otra pequeña, colocamos una unidad de cintas de salida y una impresora.



Esquema 2

Funcionará de la siguiente forma: El primer 1401 se encarga de procesar la información de las tarjetas y grabarla en la cinta; el 7094 se encarga de leer la información de la cinta grabada por el 1401, y ejecutar el programa que introducimos en la primera lectora de cintas usando el programa de la segunda lectora, el cual interpreta las ordenes del programa, por ultimo los resultados del algoritmo se grababan en la cinta de salida; el segundo 1401 se encargaba de captar esa cinta de resultados e imprimirla.

El único problema es que no sabemos con exactitud dónde comienza el programa y dónde acaban los datos, debido a esto apareció un lenguaje llamado JCL (lenguaje de control de trabajos), el cual incorpora una serie de tarjetas con información para controlar la ejecución. Para poder interpretar ese lenguaje de órdenes necesitamos un SO, llamado FORTRAN MONITOR SISTEM (FMS), el cual controlará la impresora, las cintas, las tarjetas, etc.

De esta manera hemos ganado rendimiento, pero han surgido nuevos problemas:

1. Hemos perdido eficacia ya que no controlamos el tiempo de respuesta.
2. No se pueden depurar los errores.
4. Existe una gran diferencia de velocidad entre los periféricos y el resto del sistema.

Debido a esto último aparecieron nuevos dispositivos: (usaremos de ejemplo una impresora)

1. El DMA, el cual le ahorra al procesador accesos a memoria.
2. El buffer. El procesador tiene que mandar carácter a carácter a la impresora, y esperar a que lo imprima para enviarle el siguiente. Para evitar esta pérdida de tiempo, el procesador envía 20 caracteres al buffer y este se encarga de ir enviando a la impresora carácter a carácter.

3. Spooling. Su función es sustituir un dispositivo lento por otro más rápido que haga la misma función, es decir, Se deja toda la información a imprimir en un espacio de memoria y es el buffer el que se encarga de ir enviando la información allí guardada.

Spooling de disco: Se guarda en un disco y un proceso se encarga de mandarle a la impresora cuando lo vaya pidiendo.

2.3.- 3ª GENERACIÓN : CIRCUITOS INTEGRADOS Y MULTIPROGRAMACIÓN (1965 - 1980).

- Mayor fiabilidad con un menor tamaño, gracias a los circuitos impresos y los chips.
- Nueva gama de ordenadores.

Existían dos gamas de ordenadores: · IBM 1401 (+lento, +barato), usa aplicaciones de gestión

· IBM 7094 (+rápido, +caro), usa aplicaciones científicas

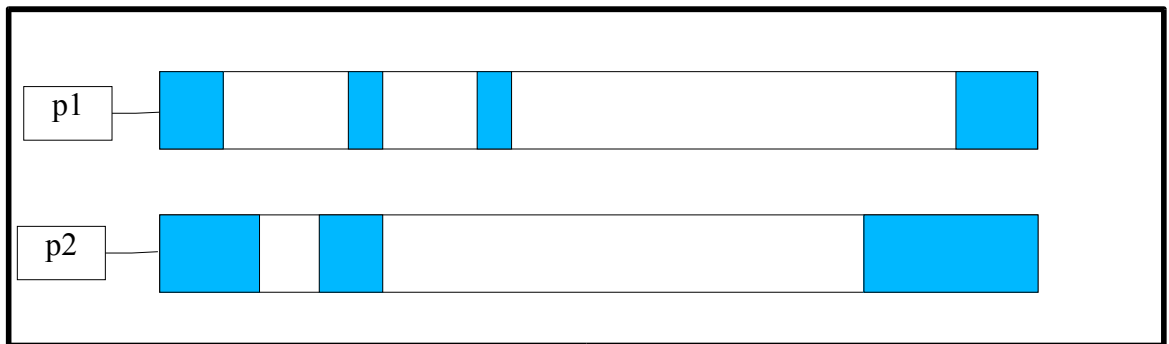
IBM creo uno nuevo que serviría para todo, de modo que creo la gama IBM360 (donde se encontraban ordenadores de muy diferentes prestaciones y precios), había la necesidad de implementar un SO que funcionará en toda la gama, nació así el SO que se llamo OS\360.

Este SO tubo muchos errores, los cuales se fueron parcheando, parches que a su vez producían nuevos errores, de este modo fue creciendo hasta convertirse en un SO muy complejo, dando pie a los SO actuales.

- Técnicas de optimización del rendimiento y eficacia.

A) Multiprogramación:

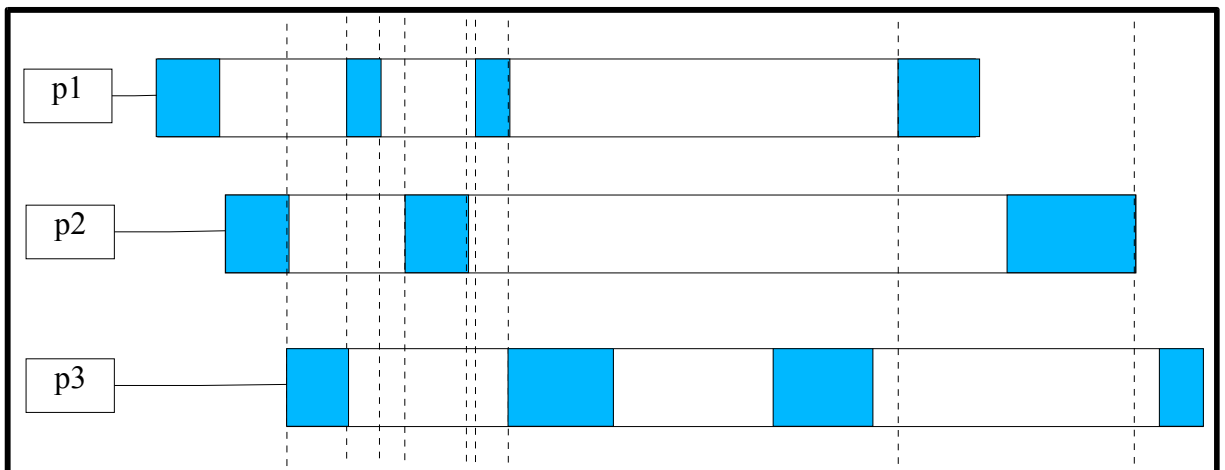
-Nos fijamos en un programa y su forma de trabajar:



Esquema 3 Cronograma sin multiprogramación

El tiempo que tardaran en ejecutarse los dos programas será: $P1+P2 = n+m$

- Mediante la multiprogramación: utiliza los tiempos muertos para realizar otras tareas.



Esquema 4 Cronograma con multiprogramación

El tiempo total sería $n+m+x$, siendo x menor que m y m menor que n .

Al número de procesos que se pueden iniciar se llama *grado de multiprogramación*.

(en este ejemplo es igual a 3).

B) Concurrencia. Cuando (con un solo procesador) el comienzo de un proceso está entre el comienzo y el término de otro proceso.

C) Paralelismo. Cuando (con dos o más procesadores) tenemos la posibilidad de hacer mas de una cosa a la vez. Dependiendo del nº de paralelismo tendremos esa misma cantidad en

procesadores.

D) Tiempo Compartido. Se utiliza cuando en So Multiusuarios queremos que todos tengan la misma respuesta del sistema. Y funciona:

Asignamos una unidad de tiempo, que llamaremos *rodaja* y se lo asignamos a todos los procesos, cuando hayamos llegado al último volveremos a empezar por el primero. Así la respuesta podrá ser rápida o lenta, pero en todo caso siempre será la misma.

El primer So que aplica esta técnica es el MULTICS (de el aparecerá posteriormente UNIX).

E) Sistemas Empotrados. Es un ordenador integrado en un elemento de ingeniería mayor, en estos sistemas no importa el rendimiento, sino la eficacia, por lo que tienen que ser sistemas de tiempo real.

2.4.- 4ª GENERACIÓN: ORDENADORES PERSONALES (1980 - ????)

· Características:

- Baja el coste de HW (muchísimo)
- Explosión del SW (existe software para todo)
- Expansión de las comunicaciones.

· Aparecen: los SO en red y los SO distribuidos.

- SO en red: Trabajan bajo redes donde esta es percibida por el usuario.
- SO distribuido: trabajan con redes entre varias máquinas donde esta no es percibida por el usuario (existe paralelismo).

3.- COMPONENTES DE UN SISTEMA OPERATIVO

Los componentes de un sistema operativo se distribuyen según la siguiente jerarquía:

USUARIOS						
PROGRAMAS DE		SHELL 1			SHELL 2	
USUARIO	WIN 32			POSIX		
	Gestor de procesos	Gestor de Memoria	Gestor de E/S	Gestor de archivos	Seguridad y protección	Comunicación y sincronización
	NÚCLEO (kernel)					
HARDWARE						

Tabla 3 Figura jerárquica de los componentes de un SO

1.- Núcleo: se encarga de los tareas básicas como son:

- Las interrupciones en general y la interrupción del reloj en particular.
- La gestión del procesador.
- La gestión básica de la memoria.

Si el núcleo cae, cae todo el sistema.

2.- Gestor de procesos: Gestiona los programas en ejecución (procesos)

3.- Gestor de memoria: Gestiona la memoria de tal forma, que si tenemos dos o más sucesos, alguno se guarda en memoria para que el cambio de proceso sea más rápido.

4.- Gestor de E/S: Gestionar las E/S es fundamental.

5.- Gestor de archivos: Gestiona los archivos de tal forma que nos permita recoger la información ordenadamente. Influye en:

- La fragmentación del disco.
- La búsqueda de archivos.

6.- Seguridad y protección: (en entornos multitarea y multiusuario) Su objetivo es que los usuarios no interfieran entre ellos ni con el sistema en todos los accesos, ya sean internos o externos.

7.- Comunicación y sincronización: Se basa en la cooperación y comunicación de los procesos entre si, para resolver problemas.

8.- WIN32 y POSIX: Las API (Application Program Interface) son el conjunto de llamas al sistema que soporta el sistema operativo. Las API mas conocidas son:

- WIN32 (estándar para los windows de 32 bits). Las funciones tienen:
 - Nombres largos, en mayúsculas y minúsculas.
- POSIX (para unix) estándar POSIX (para Linux). Las funciones:
 - Tienen nombres cortos, en minúsculas.
 - Devuelven un valor entero.

9.- Shell 1

10.- Shell 2

11.- Programas de usuario.

3.1.- GESTOR DE PROCESOS

El sistema debe planificar que proceso se va a ejecutar en cada momento.

Proceso: programa que está ejecutándose o en condiciones de ejecutarse.

Hilo: Las partes de un proceso.

3.2.- ADMINISTRADOR DE MEMORIA

Monoprogramadas	Multiprogramadas
Todos los procesos almacenados son de un único programa.	Cada proceso tiene su memoria .
Residentes	No residentes
Los procesos se quedan cargados hasta el final de su ejecución.	No se quedan cargados hasta le final.
Inmóvil	Móvil
Permanece en la misma posición de memoria hasta que finaliza.	Reubicación dinámica.
Contiguo	No contiguo
Se almacena en posiciones consecutivas de memoria.	Paginación – segmentación.
Entero	No entero
Se carga la totalidad del código.	No se carga en su totalidad.

3.3.- ADMINISTRADOR DE E/S

Debe ser capaz de reconocer dispositivos y poder ser instalados.

3.4.- SISTEMAS DE FICHEROS

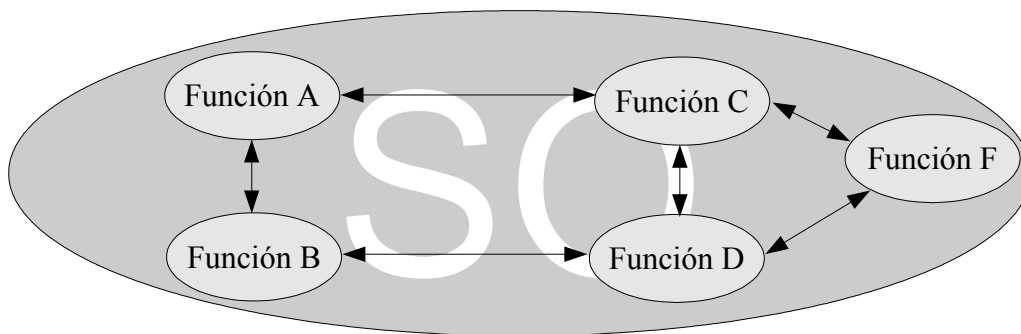
El SO debe gestionar el espacio libre y el ocupado, así como saber que lo ocupa. Se estructuran los archivos en directorios, y permite compartir información y proteger los datos.

4.- ESTRUCTURA DE LOS SISTEMAS OPERATIVOS

Es decir, cómo debemos diseñar un sistema operativo.

4.1.- ESTRUCTURA MONOLÍTICA (un solo conjunto o programa)

Se basa en un conjunto de funciones que se forman un gran programa, donde cada una realiza una función específica, lo que provoca que haya un cooperación entre las funciones. De modo que que una función depende de la ejecución previa de otras para poder ejecutarse.



Esquema 5 Estructura Monolítica.

Inconvenientes: A pesar de funcionar bastante bien, tiene un gran problema a la hora de hacer alguna modificación ya que implicaría la modificación de demasiadas funciones.

SO que siguen esta estructura:

- MS-dos.
- Unix.
- Linux.
- Windows (anteriores a NT).

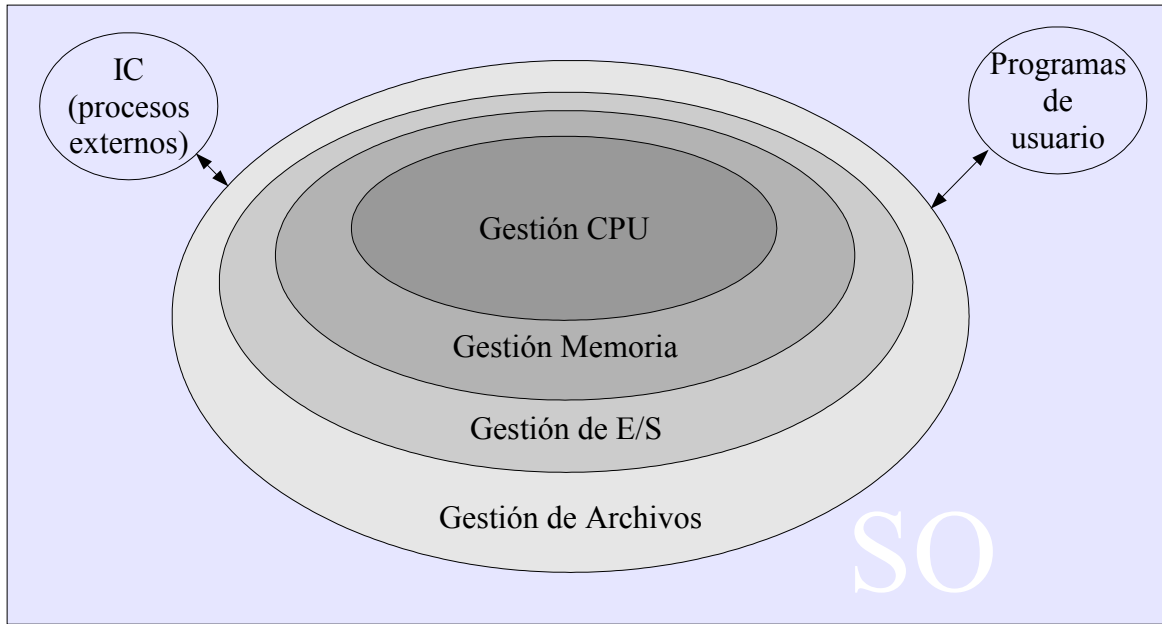
4.2.- ESTRUCTURA JERÁRQUICA.

Es una arquitectura modular, donde tendremos módulos básicos y otros que se apoyan en estos. Es lleva a la creación del TRAP

TRAP: (interface entre una capa y otra) forma en la que una capa utiliza la funcionalidad de una capa jerárquicamente inferior.

Una función llama a otra más interna, nunca al revés.

No se puede saltar entre capas, debemos ir recorriendo una a una.



Esquema 6 Estructura Jerárquica.

Ventajas: Mantenimiento mucho más fácil.

SO que siguen esta estructura:

- OS/2.

4.3.- ESTRUCTURA CLIENTE-SERVIDOR

Esta estructura está en auge. Se desarrolla con rapidez, y se puede ejecutar en casi cualquier ordenador.

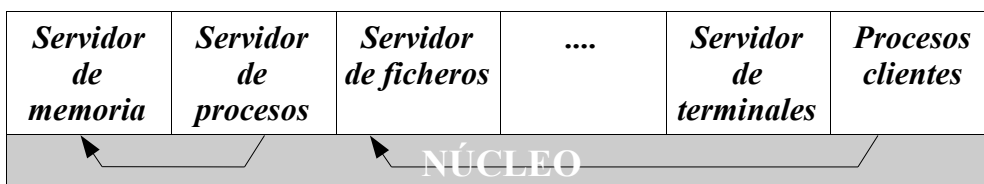
Es un SO de propósito general.

A) MICRONÚCLEO: el núcleo es lo más pequeño posible, es decir, realiza pocas funciones pero con mucha rapidez. Las funciones del micronúcleo son:

- Críticas en el tiempo: no pueden ejecutarse de otra forma.
- Manejo del procesador mediante funciones del núcleo.
- De uso muy común y general.

B) RESTO DE COMPONENTES: El núcleo hace de capa fundamental que da soporte al resto de funciones, las cuales están repartidas entre procesos que pueden ser clientes o servidores. Por lo que tendremos:

1. Servidor de memoria: Da servicio a aquellas funciones de memoria que necesiten un hueco en memoria.
2. Servidor de procesos: Se encarga de decidir que procesos va a ser el siguiente en ejecutarse.
3. Servidor de ficheros: Se encarga de buscar archivos y encontrar huecos para guardarlos.
4. Procesos clientes:



Esquema 7 Estructura Cliente-Servidor

En esta estructura, gracias al núcleo, los procesos se comunican con los servidores. Esta comunicación se establece a través del paso de mensajes. Un servidor también puede ser cliente de otro servidor.

Ventajas:

- Mayor modularidad (implica un fácil mantenimiento).
- Rápido (Implica que el núcleo es muy eficiente).
- Fiable (El fallo de una parte no implica la caída del resto del sistema).

Sistemas operativos con esta estructura:

- Windows 2000
- Minix
- DMOEBA

C) CREACIÓN DE UN SO DISTRIBUIDO A PARTIR DE CLIENTE-SERVIDOR:

Para ello replicamos el núcleo en distintas máquinas y a través de una red transmitimos los LPC y RPC:

- LPC (Llamadas a procedimientos locales): Son utilizadas por servidor y clientes para comunicarse entre si.
- RPC (Llamadas a procedimientos remotos): Son utilizadas por las distintas máquinas para comunicarse entre si.

Desventajas:

- La configuración de las E/S es aún mas complicada que en otros diseños.

4.4.- ESTRUCTURA ORIENTADA A OBJETOS

Su funcionamiento es mediante objetos, es decir, los componentes del sistema operativo son objetos.

Objeto: Es un registro al que, aparte de los datos que lo componen, se le añade procedimientos que estén relacionados directamente con esos datos.

Sistemas operativos que utilizan este diseño:

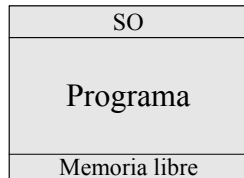
- Windows NT

5.- TIPOS DE SISTEMAS OPERATIVOS

5.1.- SEGÚN LA UTILIZACIÓN DE RECURSOS.

Ordenados de menor a mayor utilización de recursos, tenemos:

- **Monoprogramados:** Son aquellos que solo permiten un programa en ejecución. No tienen gestión de memoria, ya que toda la memoria libre está destinada al programa a ejecutar.



Esquema 8 Distribución de la memoria principal en un SO monoprogramado

- **Multiprogramados:** Permiten que haya más de un proceso que pueda estar ejecutándose (mediante concurrencia).
 - Multitarea apropiativa: Si un proceso en ejecución es quitado del procesador para poner otro.
 - Multitarea cooperativa (no apropiativa): Un proceso en ejecución deja al procesador solamente cuando ha terminado.

La memoria se reparte entre todos los procesos que intentan ejecutarse.



Esquema 9 Distribución de la memoria principal en un SO multiprogramado

- **Multiprocesamiento:** Son aquellos donde existen varios procesadores, lo cual implica la existencia de paralelismo.

5.2.- SEGÚN LA INTERACTIVIDAD.

Ordenados de menor a mayor interactividad, tenemos:

- **Procesamiento por lotes:** Son sistemas lentos, grandes y pesados, que no requieren una respuesta rápida. Consiste en agrupar todos los datos del mismo tipo con la finalidad de ejecutarlos a la vez. Interacción nula, no interviene ningún usuario.
- **Tiempo compartido:** Sistemas donde se reparte el procesador entre todas las tareas que tenemos en el sistema. Interacción alta, se utilizan sesiones.
- **Tiempo real:** Sistemas que imponen fuertes restricciones de tiempo a la respuesta ante ciertos eventos. Interacción con elementos mecánicos y electrónicos.

5.3.- SEGÚN EL NÚMERO DE USUARIOS.

Ordenados de menor a mayor número de usuarios, tenemos:

- Monousuarios (pueden ser multiprogramados)
- Multiusuarios

5.4.- SEGÚN EL TIPO DE APLICACIONES.

Tenemos:

- Propósito general: como son los sistemas operativos conocidos.
- Propósito específico: diseñados para manejar algo en concreto.