UD2
INSTALACIÓN DE SISTEMAS OPERATIVOS

Contenido

1. Sistemas Operativos.

El sistema operativo es un conjunto de programas que se encarga de gestionar los recursos hardware y software del ordenador, por lo que actúa como una interfaz entre los programas de aplicación del usuario y el hardware puro.

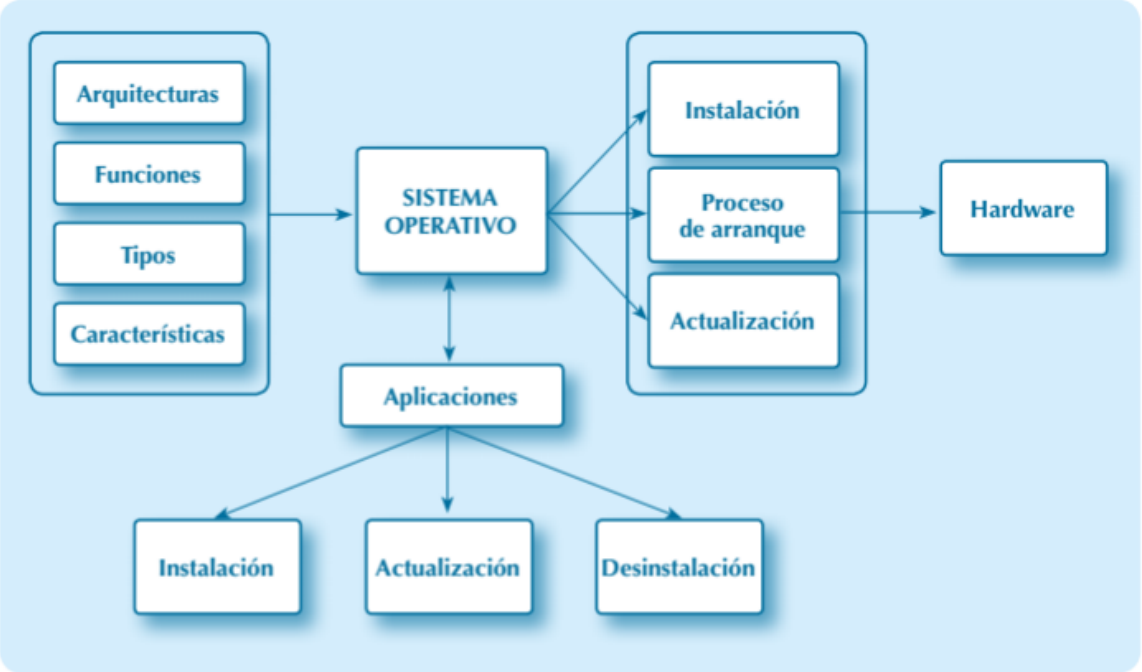


ILUSTRACIÓN 1 MAPA CONCEPTUAL

1.1. Concepto y objetivos de los sistemas operativos.

Los principales objetivos de los sistemas operativos son:

- Abstraer al usuario de la complejidad del hardware: El sistema operativo hace que el ordenador sea más fácil de utilizar.
- Eficiencia: Permite que los recursos del ordenador se utilicen de la forma más eficiente posible. Por ejemplo, se deben optimizar los accesos a disco para acelerar las operaciones de entrada y salida.
- Permitir la ejecución de programas: Cuando un usuario quiere ejecutar un programa, el sistema operativo realiza todas las tareas necesarias para ello, tales como cargar las instrucciones y datos del programa en memoria, iniciar dispositivos de entrada/salida y preparar otros recursos.

- Acceder a los dispositivos entrada/salida: El sistema operativo suministra una interfaz homogénea para los dispositivos de entrada/salida para que el usuario pueda utilizar de forma más sencilla los mismos.
- Proporcionar una estructura y conjunto de operaciones para el sistema de archivos.
- Controlar el acceso al sistema y los recursos: en el caso de sistemas compartidos, proporcionando protección a los recursos y los datos frente a usuarios no autorizados.
- Detección y respuesta ante errores: El sistema operativo debe prever todas las posibles situaciones críticas y resolverlas, si es que se producen.
- Capacidad de adaptación: Un sistema operativo debe ser construido de manera que pueda evolucionar a la vez que surgen actualizaciones hardware y software.
- Gestionar las comunicaciones en red: El sistema operativo debe permitir al usuario manejar con facilidad todo lo referente a la instalación y uso de las redes de ordenadores.
- Permitir a los usuarios compartir recursos y datos: Este aspecto está muy relacionado con el anterior y daría al sistema operativo el papel de gestor de los recursos de una red.

¿Sabes cómo han ido cambiando los sistemas operativos desde sus inicios? Resulta interesante conocer la evolución histórica que han sufrido los sistemas operativos para comprender mejor las características que explicaremos más adelante.

Evolución histórica de los sistemas operativos.

La evolución que vamos a ver de los sistemas operativos está estrechamente relacionada con los avances en la arquitectura de los ordenadores que se produjo de cada generación.

Primera generación (1945-1955)

En un principio no existían sistemas operativos, se programaba directamente sobre el hardware. Los programas estaban hechos directamente en código máquina y el control de las funciones básicas se realiza mediante paneles enchufables.

La mayoría de los programas usaban rutinas de E/S y un programa cargador (automatizaba la carga de programas ejecutables en la máquina) esto constituía una forma rudimentaria de sistema operativo.

2ª Generación (1955-1965)

La programación se realizaba en lenguaje ensamblador y en FORTRAN sobre tarjetas perforadas. Otro aspecto importante de esta generación es el procesamiento por lotes,

en el cual mientras el sistema operativo está ejecutando un proceso, éste último dispone de todos los recursos hasta su finalización. La preparación de los trabajos se realiza a través de un lenguaje de control de trabajos conocido como JCL. El sistema operativo residía en memoria y tenía un programa de control que interpretaba las tarjetas de control, escritas JCL. Dependiendo del contenido de la tarjeta de control el sistema operativo realizaba una acción determinada. Este programa de control es un antecedente de los modernos intérpretes de órdenes.

Procesamiento Fuera de línea (Offline)

Como mejora del procesamiento por lotes surgió el procesamiento fuera de línea (offline), en el cual las operaciones de carga de datos y salida de resultados de un proceso podían realizarse de forma externa y sin afectar al tiempo que el procesador dedicaba a los procesos. A esto ayudó la aparición de las cintas magnéticas y las impresoras de líneas.

Ejemplos de sistemas operativos de la época son FMS (Fortran Monitor System) y IBSYS.

3ª Generación (1965-1980)

En relación con los sistemas operativos, la característica principal de esta generación fue el desarrollo de la multiprogramación y los sistemas compartidos. En los sistemas multiprogramados se cargan varios programas en memoria simultáneamente y se alterna su ejecución. Esto maximiza la utilización del procesador. Como evolución aparecen los sistemas de tiempo compartido donde el tiempo del procesador se comparte entre programas de varios usuarios pudiendo ser programas interactivos.

Algunos de los sistemas operativos de esta generación son OS/360, CTSS, MULTICS y UNIX.

4ª Generación (1980 - hoy)

Aparecen los ordenadores personales y ejemplos de sistemas operativos de los primeros ordenadores personales son MS DOS, desarrollado por Microsoft, Inc., para el IBM PC y MacOS de Apple Computer, Inc. Steve Jobs, cofundador de Apple, apostó por la primera interfaz gráfica basada en ventanas, iconos, menús y ratón a partir de una investigación realizada por Xerox. Siguiendo esta filosofía aparecería MS Windows. Durante los 90 apareció Linux a partir del núcleo desarrollado por Linus Torvalds.

Los sistemas operativos evolucionan hacia sistemas interactivos con una interfaz cada vez más amigable al usuario. Los sistemas Windows han ido evolucionando, con diferentes versiones tanto para escritorio como para servidor (Windows 3.x, 98, 2000, XP, Vista, 7, 10, 11, Windows Server 2003, 2008, 2022, etc), al igual que lo han hecho

Linux (con multitud de distribuciones, Ubuntu, Debian, RedHat, Mandrake, etc) y los sistemas Mac (Mac OS 8, OS 9, OS X, Mac OS X 12 "Monterey" , entre otros).

Un avance importante fue el desarrollo de redes de ordenadores a mediados de los años 80 que ejecutan sistemas operativos en red y sistemas operativos distribuidos. En un sistema operativo en red los usuarios tienen conocimiento de la existencia de múltiples ordenadores y pueden acceder a máquinas remotas y copiar archivos de un ordenador a otro. En un sistema distribuido los usuarios no saben dónde se están ejecutando sus programas o dónde están ubicados sus programas, ya que los recursos de procesamiento, memoria y datos están distribuidos entre los ordenad ores de la red, pero todo esto es transparente al usuario.

Existen sistemas operativos integrados, para una gran diversidad de dispositivos electrónicos, tales como, teléfonos móviles, PDAs (Personal Digital Assistant, Asistente Digital Personal u ordenador de bolsillo), otros dispositivos de comunicaciones e informática y electrodomésticos. Ejemplos de este tipo de sistemas operativos son PalmOS, WindowsCE, Android OS, etc.

Trabajo historia GNU/Linux

https://www.qe2computing.com/sistemas/gnu-linux/historia/

Historia MS-DOS

https://www.geeknetic.es/Guia/97/MS-DOS-I-La-historia-del-sistema-operativo.html

1.2. Tipos de sistemas operativos.

Ahora vamos a clasificar los sistemas operativos en base a su estructura, servicios que suministran y por su forma.

Tipos de sistemas operativos		
Por estructura	**Por sus servicios**	**Por su forma**
Monolíticos	Monousuario	Sistema operativo en red
Jerárquicos	Multiusuario	Sistema operativo distribuido
Máquina Virtual	Monotarea	
Microkernel o Cliente-Servidor	Multitarea	
Híbridos	Monoprocesador	
	Multiprocesador	

ILUSTRACIÓN 2 TIPOS DE SO

Sistemas operativos por su estructura.

Monolíticos: Es la estructura de los primeros sistemas operativos, consistía en un solo programa desarrollado con rutinas entrelazadas que podían llamarse entre sí. Por lo general, eran sistemas operativos hechos a medida, pero difíciles de mantener.

Jerárquicos: Conforme las necesidades de los usuarios aumentaron, los sistemas operativos fueron creciendo en complejidad y funciones. Esto llevó a que se hiciera necesaria una mayor organización del software del sistema operativo, dividiéndose en partes más pequeñas, diferenciadas por funciones y con una interfaz clara para interoperar con los demás elementos. Un ejemplo de este tipo de sistemas operativos fue MULTICS.

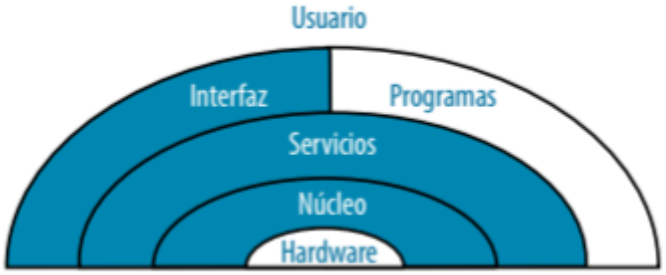


ILUSTRACIÓN 3 SO JERÁRQUICOS

Microkernel o Cliente Servidor: El modelo del núcleo de estos sistemas operativos distribuye las diferentes tareas en porciones de código modulares y sencillas. El objetivo es aislar del sistema, su núcleo, las operaciones de entrada/salida, gestión de memoria, del sistema de archivos, etc. Esto incrementa la tolerancia a fallos, la seguridad y la portabilidad entre plataformas de hardware. Un ejemplo es AIX.

Leer:

https://www.technologyuk.net/computing/computer-software/operating-systems/operating-system-architecture.shtml

Máquina Virtual: El objetivo de los sistemas operativos es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes. Presentan una interfaz a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estas máquinas no son máquinas extendidas, son 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. VMware y VM/CMS son ejemplos de este tipo de sistemas operativos.

Kernel Híbrido: Se considera una evolución que aúna las arquitecturas monolítica y microkernel, persiguiendo las ventajas de ambas. Consiste en un diseño microkernel, pero con una implementación monolítica, que consigue una gran estabilidad y un significativo rendimiento (como ventajas de ambos modelos, respectivamente). A diferencia de los sistemas microkernel, los sistemas híbridos añadirían en su espacio kernel los drivers de dispositivos y todo lo relativo a la comunicación entre procesos, como servicios fundamentales para ejecutar en modo supervisor.

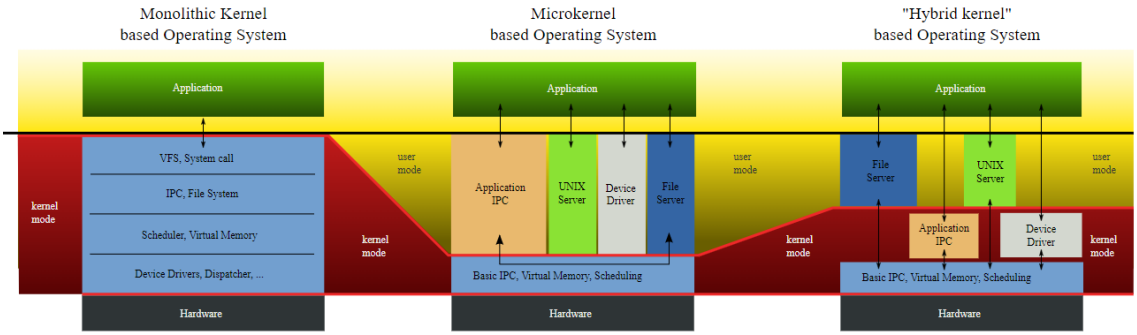


ILUSTRACIÓN 4 COMPARATIVA DE ESTRUCTURAS

Trabajo arquitecturas de los SO actuales. Buscar tipo de kernel de Windows, Linux y MacOS.

https://www.genbeta.com/a-fondo/como-es-el-kernel-de-windows-y-cuales-son-sus-diferencias-con-el-de-linux

Sistemas operativos por sus servicios.

Según el número de usuarios:

Monousuario: Son aquellos que soportan a un usuario a la vez, sin importar el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Ejemplos de sistemas operativos de este tipo son MS DOS, Microsoft Windows 9x y ME, MAC OS, entre otros.

Multiusuario: Son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varios terminales conectados al ordenador 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 puede ejecutar cada usuario simultáneamente. Algunos ejemplos serán UNIX, GNU/Linux, Microsoft Windows Server o MAC OS X.

Según el número de tareas:

Monotarea: Sólo permiten una tarea a la vez por usuario. Se puede dar el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios simultáneamente pero cada uno de ellos puede ejecutar sólo una tarea en un instante dado. Ejemplos de sistemas monotarea son MS DOS, Microsoft Windows 3.x y 95 (estos últimos sólo simulan la multitarea).

Multitarea: Permite al usuario realizar varias tareas al mismo tiempo. Algunos ejemplos son MAC OS, UNIX, Linux, Microsoft Windows 98, 2000, XP, Vista y 7.

Según el número de procesos:

Monoprocesador: Es aquel capaz de manejar sólo un procesador, de manera que si el ordenador tuviese más de uno le sería inútil. MS DOS y MAC OS son ejemplos de este tipo de sistemas operativos.

Multiprocesador: Un sistema operativo multiprocesador se refiere al número de procesadores del sistema, éste es más de uno y el sistema operativo es capaz de utilizarlos todos para distribuir su carga de trabajo. Estos sistemas trabajan de dos formas: simétricamente (los procesos son enviados indistintamente a cualquiera de los procesadores disponibles) y asimétricamente (uno de los procesadores actúa como maestro o servidor y distribuye la carga de procesos a los demás).

Según el tiempo de respuesta:

De tiempo real: requieren unos plazos en su ejecución o tiempos de respuesta.

https://www.youtube.com/watch?v=F321087yYy4

Sistemas de tiempo compartido: requieren de la participación del usuario.

Por lotes, batch o no interactivos: se suministra un conjunto de tareas al sistema operativo con características similares, y este se encarga de ejecutarlas en serie y sin la intervención del usuario. Ejemplos: realización de facturas agrupadas, tareas de cómputo en investigación…

Según la interfaz de usuario:

Textuales: emplean un repertorio de comandos que se introducen en el sistema de forma escrita a través de un terminal de órdenes.

Gráficos: usan un conjunto de ventanas, botones y desplegables gráficos donde se representan los diferentes volúmenes, unidades y sistemas de ficheros de

forma muy intuitiva. Además, los programas lanzados presentan una vista gráfica. El manejo se realiza con un dispositivo de entrada/salida, como un ratón, y destaca por su fácil utilización. Este sistema emplea muchos más recursos que el textual a nivel de procesador, memoria e incluso, en algunos casos, se necesita de manera casi obligada un adaptador gráfico.

Según la forma de acceder a los servicios.

Sistemas operativos cliente o de escritorio. Se encargan de realizar el procesamiento de la información, la gestión de los procesos, de la memoria, dispositivos de E/S de una sola computadora. Por tanto, este tipo de sistema operativo es el normalmente empleado en un hogar o pequeña oficina, así como en entornos empresariales en el ámbito de un servicio de directorio en una red distribuida.

Sistemas operativos en red: Se encargan de gestionar la red, los usuarios y los recursos de una red de computadoras en general, de forma centralizada mediante un servidor o varios como réplicas o extensiones del primero. Es en el servidor donde se instala este sistema operativo. El resto de equipos de la red (con sistemas operativos cliente) se conectan al servidor (de forma consciente) formando parte del sistema e interactuando con él. Su principal objetivo es el intercambio de información centralizada. Sin embargo, el servidor puede resultar un cuello de botella si cae o si se deteriora la transferencia de información. Destacan por su seguridad y robustez en la administración general del sistema y la gestión de la información que gestionan frente a los sistemas operativos de escritorio.

Sistemas operativos distribuidos: A diferencia de los anteriores, actúan varios computadores de manera transparente al usuario, de forma que da la sensación que este interactúa solo con uno de ellos. Por tanto, permiten emplear los recursos de varias computadoras en paralelo.

https://www.youtube.com/watch?v=NYBKXzl5bWU

https://setiathome.ssl.berkeley.edu/

1.3. Versiones de los sistemas operativos más utilizados.

Vamos a realizar un trabajo de investigación sobre versiones de sistemas operativos actuales:

Para S.O. de Microsoft, Apple y basados en GNU/Linux (2 distribuciones) obtener la siguiente información:

- Versiones actuales para escritorio, móvil y servidor.
- Características que el fabricante destaca de cada versión.

La página web https://distrowatch.com/ aglutina mucha información y permite comparar multitud de distribuciones GNU/Linux, BSD y Solaris.

¿Cuáles son las 5 distribuciones más populares actualmente, según el ranking ofrecido por esta página? (Average rating (>100 votes)) Establece una comparativa, indicando: tipo de sistema operativo, en qué sistema operativo está basado y para qué plataformas o entornos (seguridad, sobremesa, servidores).

En la web https://www.top500.org/ encontramos una lista actualizada de los quinientos sistemas de computación más potentes. Realiza una tabla con los cinco primeros, donde se indique:

- Localización: ciudad y país.
- Fabricante.
- El número de núcleos.
- El número de operaciones máximas en TFlops.
- El sistema operativo empleado.

2. Funciones de los Sistemas operativos.

2.1. Gestión de procesos.

Entre las principales tareas del sistema operativo está la de administrar los procesos del sistema. ¿A qué nos referimos cuando hablamos de procesos?

Un proceso en un programa en ejecución. Un proceso simple tiene un hilo de ejecución (o subproceso), en ocasiones, un proceso puede dividirse en varios subprocesos. **Un hilo es básicamente una tarea que puede ser ejecutada en paralelo con otra tarea**. Por lo que los hilos de ejecución permiten a un programa realizar varias tareas a la vez.

Un proceso, en un instante dado, puede estar en uno de los tres **estados** siguientes:

- **Listo:** Los procesos en estado listo son los que pueden pasar a estado de ejecución si el planificador del sistema operativo los selecciona, esto es, cuando llegue su turno (según el orden de llegada o prioridad).
- **En ejecución**: Los procesos en estado de ejecución son los que se están ejecutando en el procesador en un momento dado.
- **Bloqueado:** Los procesos que se encuentran en estado bloqueado están esperando la respuesta de algún otro proceso para poder continuar con su ejecución, por ejemplo, una operación de entrada/salida.

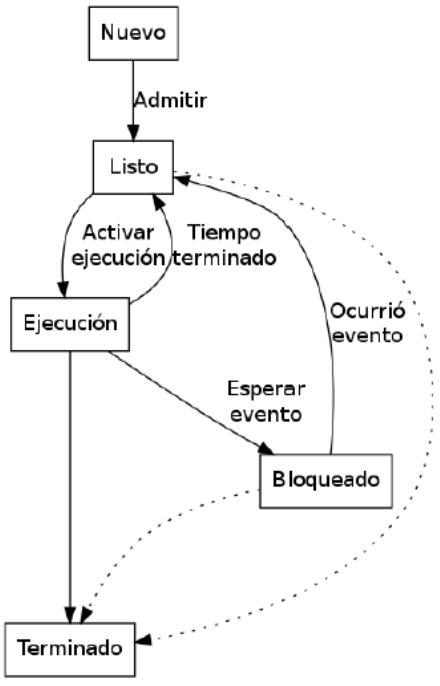


ILUSTRACIÓN 5 DIAGRAMA TRANSICIÓN DE ESTADOS DE UN PROCESO.

El sistema operativo sigue la pista de en qué estado se encuentran los procesos. El **planificador de procesos** (process scheduler, en inglés) es la parte del sistema operativo que se encarga de seleccionar a qué proceso se asigna el recurso procesador y durante cuánto tiempo.

Los procesos pueden comunicarse entre sí o ser independientes. En el primer caso, los procesos necesitarán sincronizarse y establecer una serie de mecanismos para la comunicación.

2.1.1. Planificación del procesador.

En la planificación del procesador se decide cuánto tiempo de ejecución se le asigna a cada proceso del sistema y en qué momento.

El sistema operativo almacena en una tabla denominada **tabla de control de procesos** la información relativa a cada proceso que se está ejecutando en el procesador. Ésta es:

- Identificación del proceso.
- Identificación del proceso padre.
- Información sobre el usuario y grupo que lo han lanzado.
- Estado del procesador. El contenido de los registros internos, contador de programa, etc. Es decir, el entorno volátil entorno volátil del proceso.
- Información de control de proceso.
- Información del planificador.
- Segmentos de memoria asignados.
- Recursos asignados.

Cada elemento de esta tabla se conoce con el nombre de descriptor de proceso **o bloque de control de proceso (BCP ó PCB-Process Control Block)** y constituye la representación de un proceso para el sistema operativo.

Una **estrategia de planificación debe buscar** que los procesos obtengan sus turnos de ejecución de forma apropiada , con un buen rendimiento y minimizando la sobrecarga (overhead) del planificador mismo. En general, se buscan los siguientes objetivos:

- **Ser justo.** Debe tratarse de igual manera a todos los procesos que compartan las mismas características, y nunca postergar indefinidamente uno de ellos.
- **Maximizar el rendimiento.** Dar servicio a la mayor parte de procesos por unidad de tiempo.
- **Ser predecible.** Un mismo trabajo debe tomar aproximadamente la misma cantidad de tiempo en completarse independientemente de la carga del sistema.
- **Minimizar la sobrecarga.** El tiempo que el algoritmo pierda en burocracia debe mantenerse al mínimo, dado que éste es tiempo de procesamiento útil perdido.

- **Equilibrar el uso de recursos**. Favorecer a los procesos que empleen recursos subutilizados, penalizar a los que peleen por un recurso sobre utilizado causando contención en el sistema.
- **Evitar la postergación indefinida**. Aumentar la prioridad de los procesos más viejos, para favorecer que alcancen a obtener algún recurso por el cual estén esperando.

La carga de trabajo de un sistema informático a otro puede variar considerablemente, esto depende de las **características de los procesos**. Nos podemos encontrar:

- Procesos que hacen un uso intensivo de la CPU.
- Procesos que realizan una gran cantidad de operaciones de Entrada/Salida.
- Procesos por lotes, procesos interactivos, procesos en tiempo real.
- Procesos de menor o mayor duración.

En función de cómo sean la mayoría de los procesos habrá algoritmos de planificación que den un mejor o peor rendimiento al sistema.

2.1.2. Planificación apropiativa y no apropiativa.

La **planificación no apropiativa** (en inglés, no preemptive) es aquélla en la que, cuando a un proceso le toca su turno de ejecución, ya no puede ser suspendido. Este esquema tiene sus problemas, puesto que, si el proceso contiene ciclos infinitos, el resto de los procesos pueden quedar aplazados indefinidamente.

La **planificación apropiativa** (en inglés, preemptive) supone que el sistema operativo puede arrebatar el uso de la CPU a un proceso que esté ejecutándose. En la planificación apropiativa existe un reloj que lanza interrupciones periódicas en las cuales el planificador toma el control y se decide si el mismo proceso seguirá ejecutándose o se le da su turno a otro proceso.

En ambos enfoques de planificación se pueden establecer distintos algoritmos de planificación de ejecución de procesos.

Algoritmos de planificación

Algunos de los algoritmos para decidir el orden de ejecución de los procesos en el sistema son los siguientes:

Primero llegado, primero servido (FCFS)

Es el más sencillo, pero el más ineficaz. Como su nombre indica First In First Out, los procesos se ejecutan según el orden de llegada; es decir, el primero en llegar es el primero en salir.

Ejemplo:

TAREAS	Orden de llegada	Tiempo de Servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Ejecución	A	A	A	B	B	B	B	B	B	C	C	C	C	D	D	D	D	D	E	E
Ciclos de CPU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Esto se puede representar en un diagrama de Gantt

E									L	L	L	L	L	L	L
D							L	L	L	L	L	L	L	X	X
C					L	L	L	L	L	X	X	X	F		
B			L	X	X	X	X	X	F						
A	X	X	F												
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

E	L	L	L	X	F										
D	X	X	F												
C															
B															
A															
0	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Algoritmo rotatorio (Round Robin)

Un modo sencillo de reducir la penalización que sufren los trabajos cortos con FIFO es considerar la apropiación dependiente de un reloj. La más simple de estas políticas se denomina planificación por turno rotatorio (RR, Round Robin).

Periódicamente, se genera una interrupción de reloj. Cuando se genera la interrupción, el proceso que está en ejecución se sitúa en la cola de Listos y se selecciona el siguiente trabajo, según un FIFO.

Esta técnica se conoce también como fracciones de tiempo, puesto que cada proceso recibe una fracción de tiempo antes de ser expulsado.

Con la política del turno rotatorio, la cuestión principal del diseño es la longitud del quantum de tiempo o fracción (q) que se va a usar.

Ejemplo:

TAREAS	Orden de llegada	Tiempo de Servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Veamos el modo en que el algoritmo de RR ejecuta los 5 procesos.

a) Si el quantum = 1.

Ejecución	A	A	B	A	B	C	B	D	C	B	E	D	C	B	E	D	C	B	D	D
Ciclos de CPU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

b) Si el quantum = 4

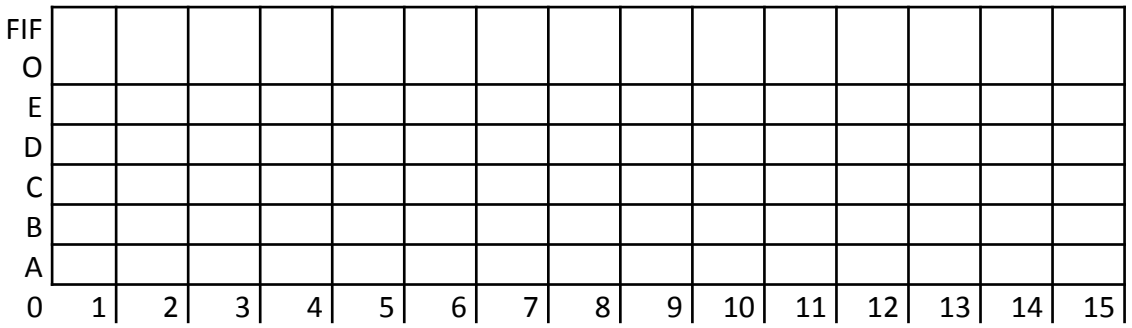
Ejecución	A	A	A	B	B	B	B	C	C	C	C	D	D	D	D	B	B	E	E	D
Ciclos de CPU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

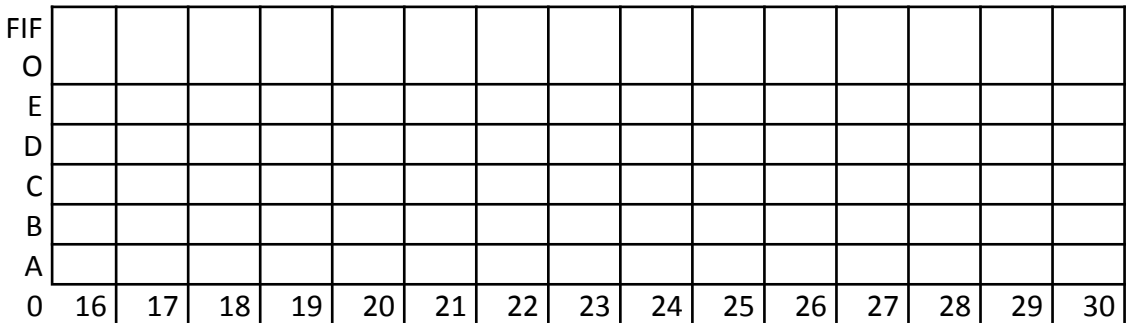
Con quantum = 1 y en caso de empate priorizamos el proceso nuevo.

FIFO			A	B	C	B	DC	CB	BED	EDC	DCB	CBE	BED	EDC	DCB
E									L	L	X	L	L	L	F
D							L	X	L	L	L	X	L	L	L
C					L	X	L	L	X	L	L	L	X	L	L
B			X	L	X	L	X	L	L	X	L	L	L	X	L
A	X	X	L	F											
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

FIFO															
	CB	BE	E												
E															
D	X	L	L	X	X										
C	L	F													
B	L	L	F												
A															
0	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Vamos a hacerlo con el quantum = 4 y en caso de empate priorizamos el proceso nuevo.





Algoritmo primero proceso más corto (SJF)

Es otra forma de reducir el sesgo favorable al proceso más largo inherente al FIFO. Esta es una política no preferente en la que se selecciona el proceso con menor tiempo esperado de ejecución. Así pues, un proceso corto saltará a la cabeza de la cola, sobrepasando a trabajos largos.

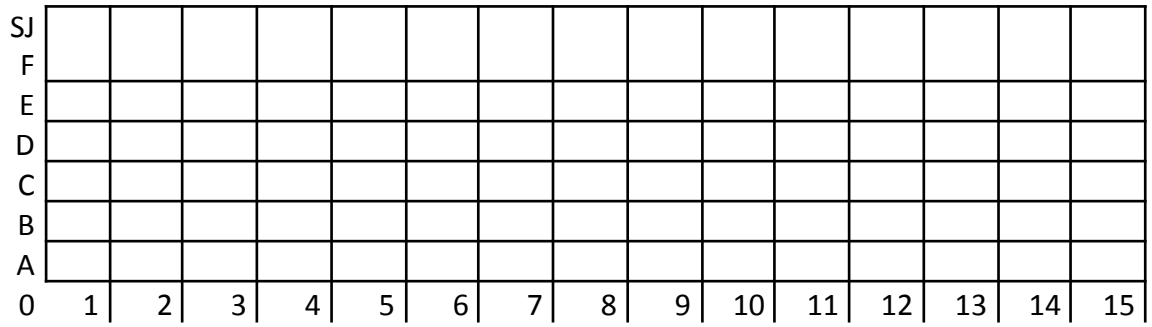
Una dificultad que plantea esta política es la necesidad de conocer o, por lo menos, estimar, el tiempo exigido en cada proceso.

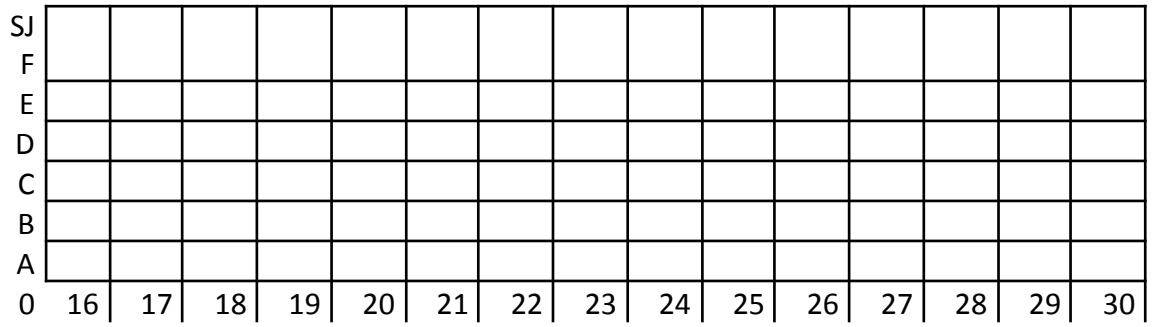
Ejemplo:

TAREAS	Orden de llegada	Tiempo de Servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Ejecución	A	A	A	B	B	B	B	B	B	E	E	C	C	C	C	D	D	D	D	D
Ciclos de CPU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

En caso de empate priorizamos el proceso más antiguo.





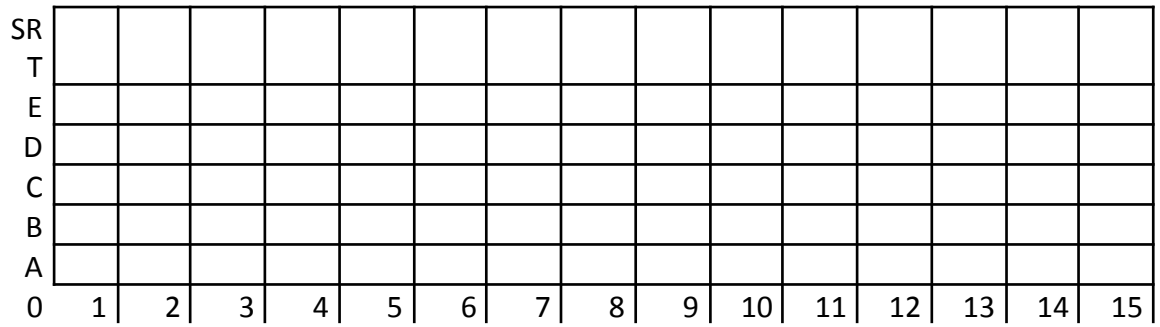
Algoritmo de Menor Tiempo Restante (SRT)

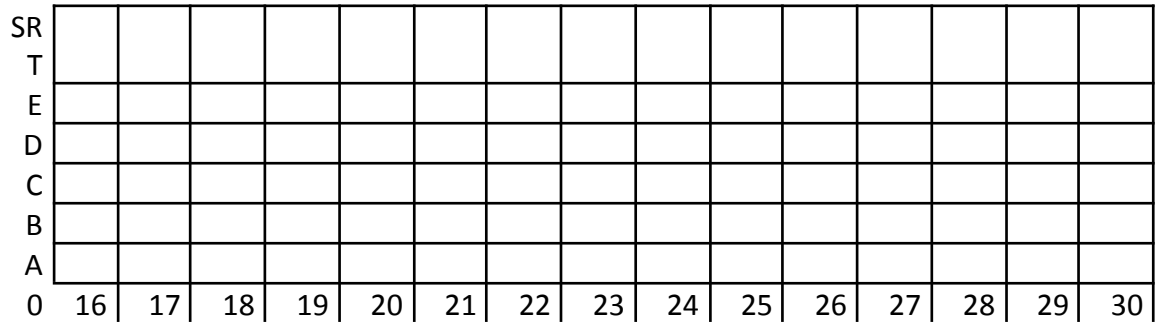
Este algoritmo permite asignar el tiempo de ejecución de forma prioritaria a procesos muy cortos para ejecutarlos en el menor tiempo posible. Si se está ejecutando un proceso más largo, el sistema operativo le quitará la ejecución de la CPU para asignársela al proceso más corto. De esta forma, el usuario del proceso corto obtendrá resultados en un tiempo mínimo, y el usuario del proceso largo casi no notará esta circunstancia.

Ejemplo:

TAREAS	Orden de llegada	Tiempo de Servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Ejecución	A	A	A	B	C	C	C	C	E	E	B	B	B	B	B	D	D	D	D	D
Ciclos de CPU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20





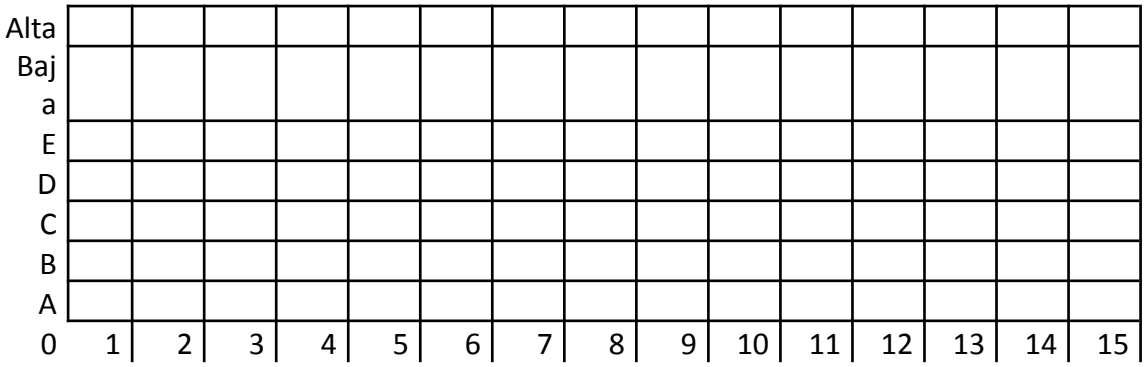
Algoritmo por prioridades o multinivel

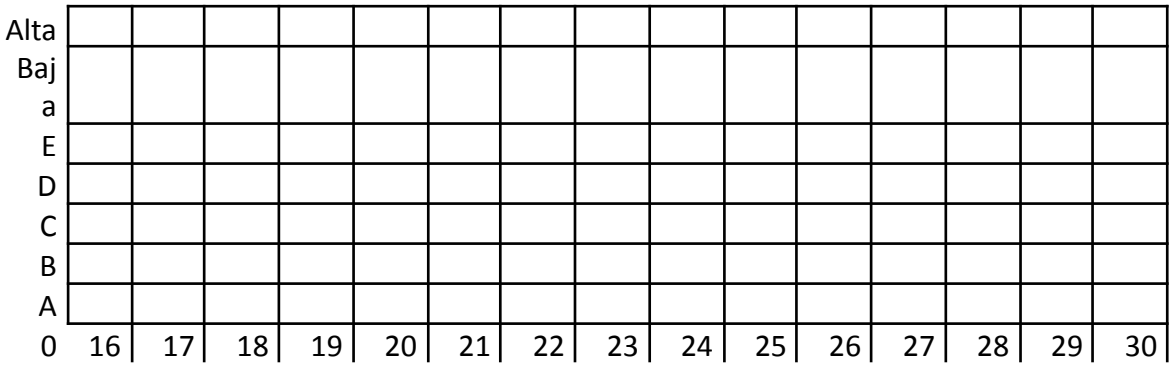
Es más complejo y eficaz. Asigna los tiempos de ejecución de la CPU según una lista de prioridades. En cada una de estas listas, el sistema operativo incluirá aquellos procesos a los que se haya asignado esa prioridad.

El tiempo de ejecución del procesador se irá destinando, en primer lugar, de forma secuencial a los procesos de mayor nivel. Terminados éstos, se ejecutarán los procesos del nivel inferior, y así sucesivamente, hasta llegar a los procesos del nivel más bajo.

Ejemplo: El planificador en este caso clasificará las tareas como prioridad alta y baja. Las tareas B y D tendrán prioridad alta. Las tareas A, C y E baja. El algoritmo será apropiativo entre niveles. Dentro de cada nivel se usará FCFS para el desempate.

TAREAS	Orden de llegada	Tiempo de Servicio
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2





Medida de rendimiento de los algoritmos de planificación

Los algoritmos de planificación tendrán distintas propiedades y favorecerán cierta clase de procesos. Es necesario definir criterios para poder evaluar los algoritmos de planificación:

- Utilización de CPU: Es el porcentaje de uso (en cuanto a ejecución de tareas de usuario o del sistema que son consideradas útiles) que tiene un procesador.
- Rendimiento: Es el número de procesos que ejecutaron completamente por unidad de tiempo (una hora p.ej.).
- Tiempo de servicio: Es el intervalo de tiempo que necesita un proceso de ejecución en la CPU más el tiempo de E/S. Ts
- Tiempo de espera: Es la suma de los intervalos de tiempo que un proceso estuvo en la cola de procesos listos. Te
- Tiempo de retorno: Es el intervalo de tiempo desde que un proceso es cargado hasta que este finaliza su ejecución. Tr = Ts + Te
- Tiempo de respuesta: Es el intervalo de tiempo desde que un proceso es cargado hasta que brinda su primera respuesta. Es útil en sistemas interactivos. Tre
- Índice de servicio: Es la proporción entre el tiempo de servicio y el tiempo de retorno. Is = Ts / Tr

Ejemplos:

Para los casos anteriores FCFS, RR, SJF, SRT, Multinivel vamos a calcular:

- Tiempo de servicio. Ts

- Tiempo de espera. Te
- Tiempo de retorno. Tr
- Tiempo de respuesta. Tre
- Índice de servicio. Is
- Las medias de cada uno de los parámetros anteriores

Primero en llegar primero en servir, FCFS

	Ts	Te	Tr	Tre	Is
A					
B					
C					
D					
E					
Media					

Round Robin, RR q=1

	Ts	Te	Tr	Tre	Is
A					
B					
C					
D					
E					
Media					

Round Robin, RR q=4

	Ts	Te	Tr	Tre	Is
A					
B					
C					
D					
E					
Media					

Trabajo más corto primero, SJF

	Ts	Te	Tr	Tre	Is
A					
B					
C					
D					
E					
Media					

Menor tiempo restante primero, SRT

	Ts	Te	Tr	Tre	Is
A					
B					
C					
D					
E					
Media					

Multinivel

	Ts	Te	Tr	Tre	Is
A					
B					
C					
D					
E					
Media					

3. Gestión de memoria.

En la gestión de procesos que el **recurso compartido es el procesador**. Sin embargo, para que un proceso se pueda ejecutar no sólo requiere tiempo de procesamiento sino también estar cargado en memoria principal. Ningún proceso se puede activar antes de que se le asigne el espacio de memoria que requiere. Así, **la memoria** se convierte en otro **recurso clave** que tendrá que **gestionar el sistema operativo** y la parte encargada de ello se denomina gestor de memoria.

La función principal del **gestor de memoria** es la de **asignar memoria principal a los procesos que la soliciten**. Otras funciones serán:

- Controlar las zonas de memoria que están asignadas y cuáles no.
- Asignar memoria a los procesos cuando la necesiten y retirársela cuando terminen.
- Evitar que un proceso acceda a la zona de memoria asignada a otro proceso.
- Gestionar el intercambio entre memoria principal y memoria secundaria en los casos en que la memoria principal está completamente ocupada, etc.

De este modo, la gestión de memoria va a tener que cubrir los siguientes requisitos:

- **Reubicación**: En un sistema multitarea la memoria va a estar compartida entre varios procesos, el gestor de memoria debe decidir qué zonas de memoria asigna a cada proceso y que zonas descarga.
- **Protección**: El gestor de memoria debe evitar que los procesos cargados en memoria interfieran unos con otros accediendo a zonas de memoria que no les corresponden, Para ello, se comprueba que las referencias a la memoria generadas por un proceso durante su ejecución sólo hacen referencia a la zona de memoria asignada a ese proceso y no acceden a zonas prohibidas, áreas de memoria donde estén otros procesos.
- **Control de memoria**: El sistema operativo, a través del gestor de memoria, tiene que controlar las zonas de memoria libres y las asignadas, además de saber las zonas de memoria que corresponden a cada proceso.
- **Controlar y evitar en lo posible casos de fragmentación de la memoria**: Existen dos tipos de fragmentación de la memoria principal, la fragmentación interna y la externa. La fragmentación interna sucede al malgastarse el espacio interno de una partición cuando el proceso o bloque de datos cargado es más pequeño que la partición. Por el contrario, la fragmentación externa sucede cuando la memoria externa a todas las particiones se divide cada vez más y van quedando huecos pequeños y dispersos en memoria difícilmente reutilizables.
- **Organización lógica y física**: En ocasiones la memoria principal no es suficiente para proporcionar toda la memoria que necesita un proceso o para almacenar todos los procesos que se pueden ejecutar. Entonces los procesos pueden ser intercambiados a disco y más tarde, si es necesario, vueltos a cargar en

memoria. Por lo que el gestor de memoria se encarga de gestionar la transferencia de información entre la memoria principal y la secundaria (disco).

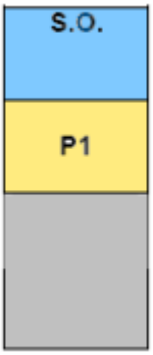
El sistema de **gestión de la memoria** que se use dependerá del ordenador y **sistema operativo** en particular que se tenga. Las opciones en la gestión de memoria se dividen en función del número de procesos albergados en memoria (monotarea/multitarea) y de si se utiliza memoria real o virtual.

Gestión de la memoria con memoria real y virtual

<table>
<tr><th>Memoria Real</th><th colspan="2">Memoria Real</th><th colspan="2">Memoria Virtual</th></tr>
<tr><td rowspan="5">Monotarea</td><td colspan="2">Multitarea</td><td colspan="2">Multitarea</td></tr>
<tr><td colspan="2">Particiones</td><td>Memoria virtual paginada</td><td>Memoria virtual segmentada</td></tr>
<tr><td>Fijas</td><td>Variables</td><td colspan="2" rowspan="2">Combinación</td></tr>
<tr><td>Paginación pura</td><td>Segmentación pura</td></tr>
<tr><td colspan="2">Relocalización</td><td colspan="2">Protección</td></tr>
</table>

3.1. Gestión de memoria en sistemas operativos monotarea.

En sus orígenes los sistemas operativos no incluían ningún gestor de memoria, y el programador tenía un control completo sobre el espacio total de memoria. La memoria real se utiliza para almacenar el programa que se esté ejecutando en un momento dado. Conforme los procesos se ejecutan secuencialmente a medida que van terminando los anteriores.

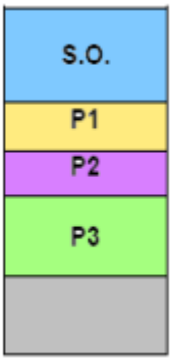


Se trata del esquema más sencillo, en cada momento la memoria alberga un solo proceso y reserva otra zona de la memoria para el sistema operativo. Por ello, se

necesita un mecanismo de protección para evitar accesos a la parte del sistema operativo de los procesos de usuario.

3.2. Gestión de memoria en sistemas operativos multitarea.

Actualmente la mayoría de los sistemas operativos son sistemas multitarea, en los que va a haber varios procesos simultáneamente en ejecución. Para que esto sea posible, todos estos procesos deberán estar también simultáneamente en memoria, pues ésta es una condición necesaria para que un proceso pueda ejecutarse. Por tanto, deberá haber mecanismos de gestión para distribuir la memoria principal entre todos estos procesos que quieren ejecutarse.



Intercambio o swapping

Como sabemos la memoria principal es un recurso limitado, por ello puede ocurrir que haya más procesos esperando a ser cargados en memoria que zonas libres en la misma. En estos casos, el gestor de memoria sacará de la memoria algunos procesos (bloqueados, suspendidos, que estén esperando a que finalice una operación de entrada/salida, etc.) y los llevará a un área de disco (memoria secundaria), conocida como área de intercambio o de swap. A esta operación se la denomina intercambio o swapping. Los procesos permanecerán allí hasta que existan huecos libres en memoria y puedan ser recuperados de disco y reubicados en memoria principal.

3.2.1. Asignación de particiones fijas.

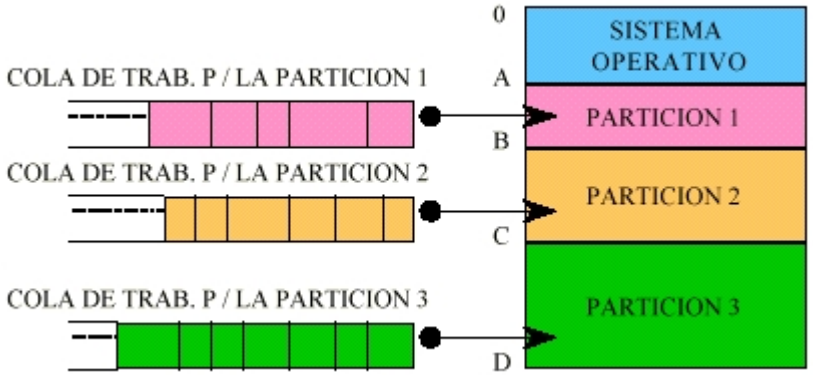
Hemos estudiado que el gestor de memoria necesita reservar un espacio de memoria para el sistema operativo y que el resto de la memoria queda para los procesos de usuarios. Cuando existen varios procesos que requieren ser cargados en memoria el gestor de memoria tiene que organizar el espacio para ubicarlos.

Hay varias alternativas, la primera de ellas es dividir el espacio de memoria en particiones fijas. Estas particiones podrán ser todas del mismo tamaño o tener distintos tamaños. Estas particiones se establecen de forma lógica por el sistema operativo y están predefinidas antes de que lleguen los procesos. El número de particiones se mantiene fijo en el tiempo, así como el tamaño de cada una de las particiones.

La gestión y asignación de particiones a los procesos se puede hacer siguiendo dos tipos de organización:

Una cola por partición.

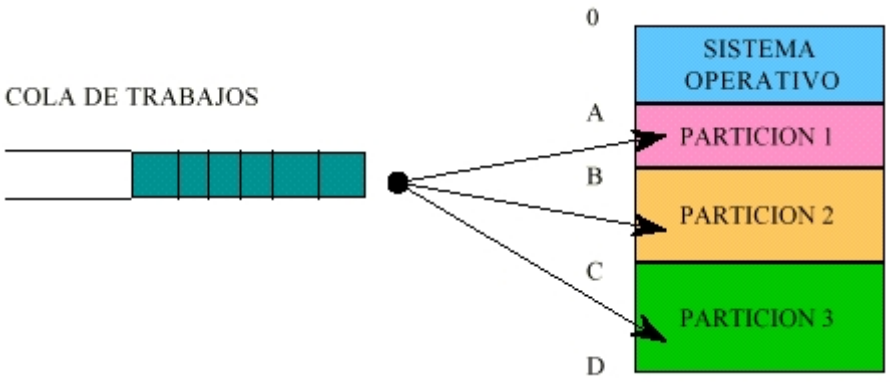
Se tiene una cola por cada partición y se coloca cada trabajo en la cola de la partición más pequeña en que quepa dicho trabajo, a fin de desperdiciar el menor espacio posible.



La planificación de cada cola se hace por separado y, como cada cola tiene su propia partición, no hay competencia entre las colas por la memoria. La desventaja de este método se hace evidente cuando la cola de una partición grande está vacía y la cola de una partición pequeña está llena.

Una única cola común a todas las particiones.

Se tiene una única cola común para todas las particiones. El sistema operativo decidirá en que partición se ubica cada proceso. En función de la disponibilidad de particiones y las necesidades del proceso en cuestión.



En ambos casos, utilización de una cola por partición o uso de una única cola para los procesos, el gestor de memoria establecerá mecanismos para impedir que un proceso pueda acceder a una zona de memoria que está fuera de la memoria correspondiente a la partición en la que se encuentra.

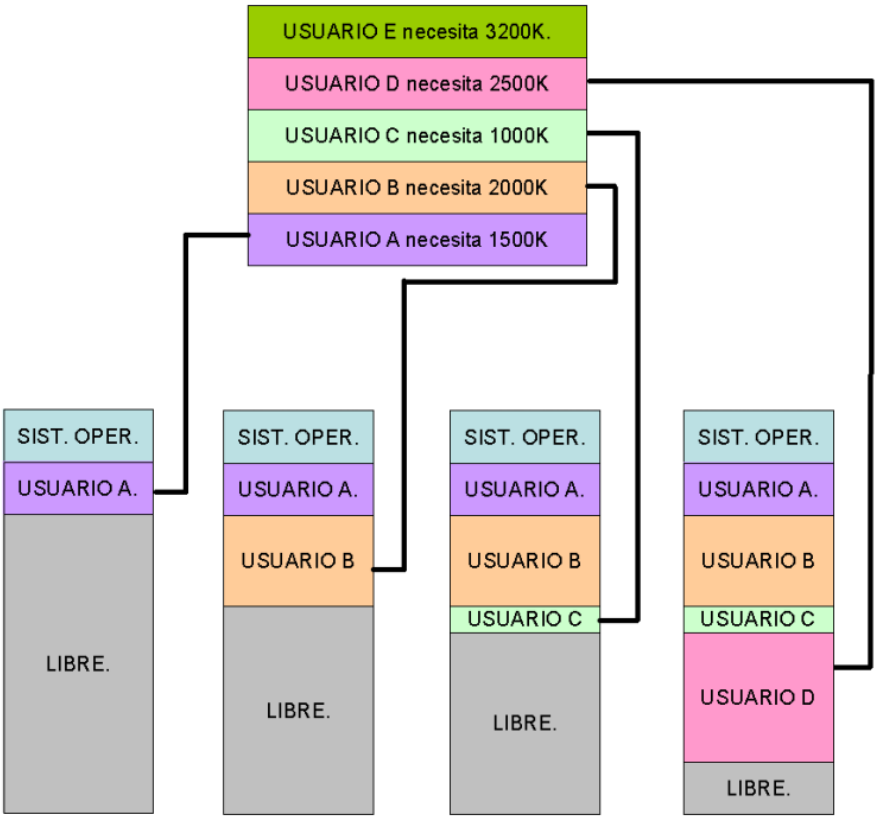
Además de esto, puede surgir el problema de la fragmentación, la cual se produce, cuando en la memoria hay áreas ocupadas intercaladas con áreas libres; es decir, cuando no hay una única área ocupada ni una única área libre.

3.2.2. Asignación de particiones variables.

Con la asignación de particiones fijas se tiene la desventaja de que no se aprovecha, con frecuencia, todo el tamaño de cada partición, ya que el proceso se adapta a los tamaños fijos ya preestablecidos en memoria. En este punto se plantea una segunda alternativa, la asignación de memoria a los procesos mediante particiones variables. La idea es crear las particiones dinámicamente, conforme llegan los procesos y en función de los tamaños de estos. Este método de gestión de memoria se conoce con el nombre de asignación de la memoria con particiones variables. Es una técnica más realista que aprovecha mejor el espacio de la memoria.

Este mecanismo se ajusta a la realidad de que el número y tamaño de los procesos varía dinámicamente y, por tanto, lo lógico es que no se está sujeto a un número fijo de particiones que pudieran ser muy grandes o demasiado pequeñas, con lo que se consigue un mejor uso de la memoria, aunque a costa de una mayor complejidad.

En la asignación de particiones variables, el sistema operativo debe llevar el control de qué partes de la memoria están disponibles y cuales libres.



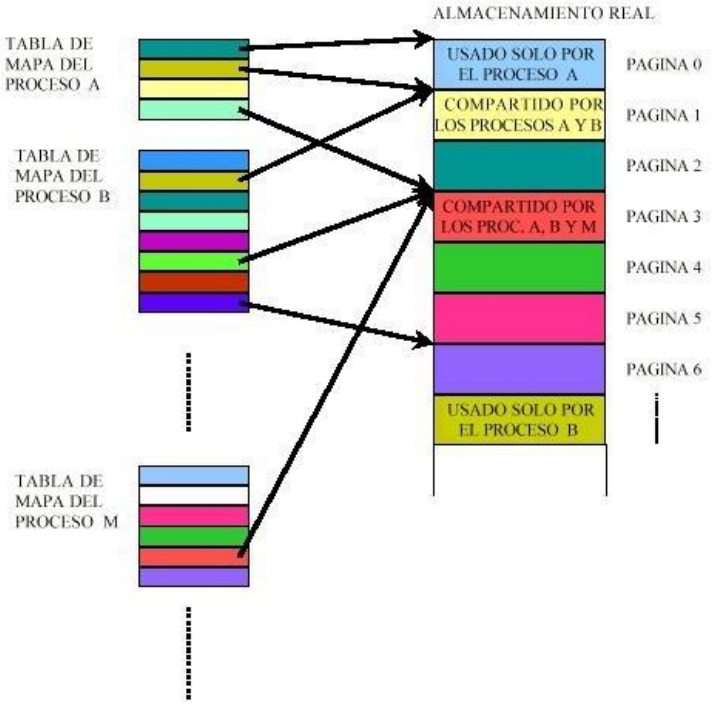
3.2.3. Memoria virtual.

Hasta este momento los procesos se cargaban enteros en la memoria, pero podría suceder que existan procesos grandes que no quepan en las particiones de la memoria y por tanto, no puedan ser cargados por completo en la memoria.

La memoria virtual da una solución a estos casos, ya que permite dividir los procesos en varias partes y cargar sólo algunas de ellas en memoria. La memoria virtual se basa en el uso de las técnicas de paginación o segmentación.

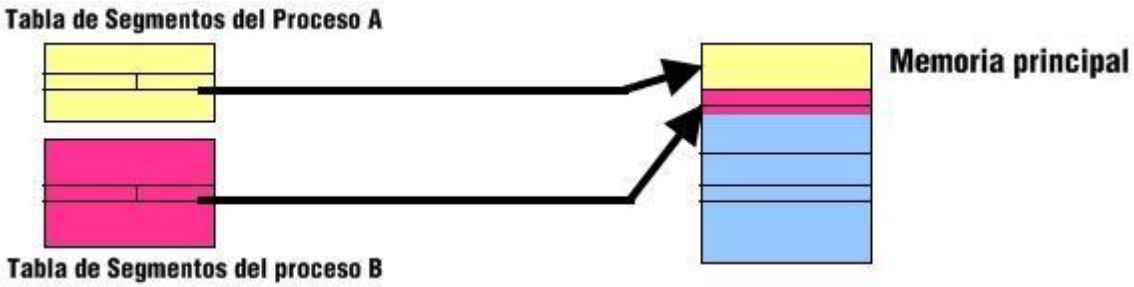
Paginación pura

La idea es la de dividir la memoria principal en un conjunto de particiones conocidas como “marcos de página” de igual tamaño. Cada proceso se divide a su vez en una serie de partes llamadas “páginas” del mismo tamaño que los marcos. El proceso se carga en memoria situando todas sus páginas en los marcos de página de la memoria, sin embargo, las páginas no tienen por qué estar contiguas en memoria. Como ventaja reduce la fragmentación externa de la memoria principal. Sin embargo, puede aparecer cierta fragmentación interna.



Segmentación pura

Cada proceso se divide en una serie de segmentos. La peculiaridad de estos segmentos es que su tamaño no tiene que ser el mismo y puede variar hasta un límite máximo. Un proceso se carga situando todos sus segmentos en particiones dinámicas que no tienen que estar contiguas en memoria. Este sistema reduce la fragmentación interna de la memoria principal.



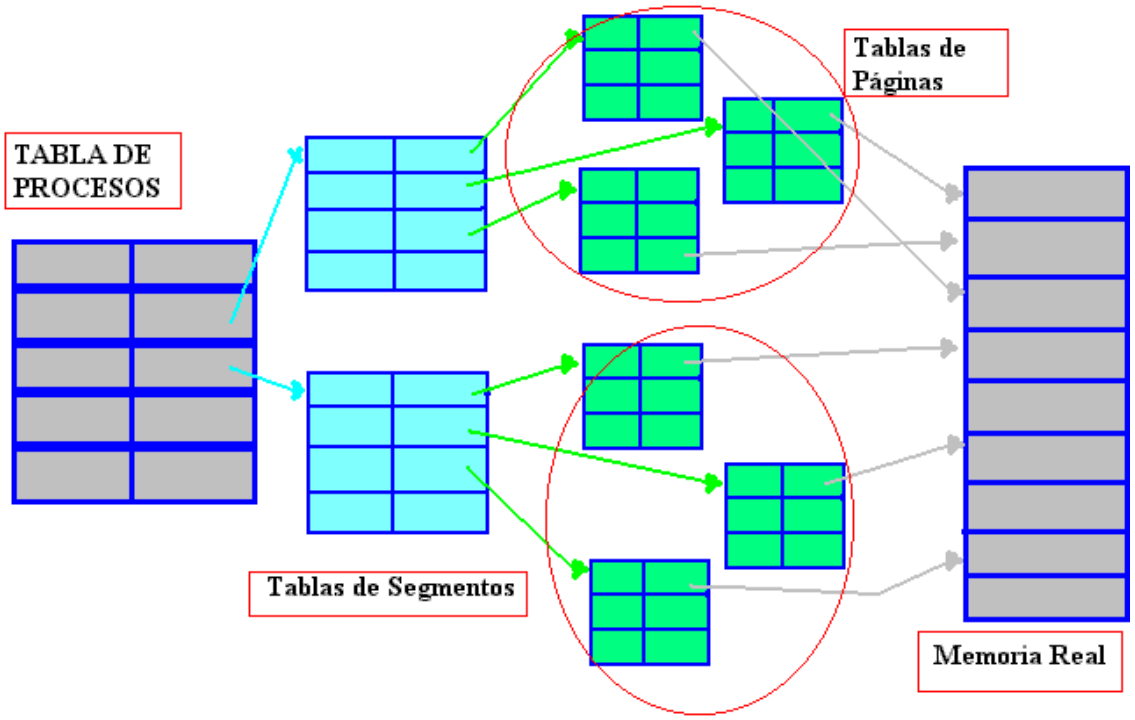
Como hemos comentado, no todas las partes de un proceso pueden estar cargadas en memoria en un instante determinado. Por ello, cuando un proceso haga referencia a un parte que no se encuentre asignada en memoria provocará un fallo de página o segmento, y el gestor de memoria traerá dicha parte del proceso de disco a memoria.

La utilización de las técnicas de paginación o segmentación por parte de la memoria virtual se conocen como:

Memoria Virtual Paginada: Sigue el funcionamiento de la paginación simple, pero no es necesario cargar todas las páginas de un proceso para que éste pueda ejecutarse. Las páginas que no se encuentren y se necesiten se traerán posteriormente a memoria de manera automática. Reduce la fragmentación

Memoria Virtual Segmentada: En este caso la operación sería la misma que en la segmentación simple, pero tampoco será necesario cargar todos los segmentos de un proceso. Si se necesitan más segmentos no asignados en memoria se traerán en el momento en que sean referenciados.

Combinación de las técnicas de segmentación y paginación: En la figura siguiente vemos el funcionamiento de la combinación de ambas técnicas.



Ejercicio 1. Sea un computador de 20 bits con memoria virtual paginada con páginas de 1 KB y un total de memoria física de 256 KB. Se pide, de forma razonada y breve:

a) ¿Cuál es el formato de la dirección virtual? Indique los campos y el número de bits de los mismos.

b) ¿Cuál es el número máximo de entradas de la tabla de páginas (de un nivel)?

c) ¿Cuántos marcos de página tiene la memoria principal?

Ejercicio 2. Considere un computador de 32 bits que dispone de un sistema de memoria virtual que emplea páginas de 16 KB y tiene instalada una memoria principal de 1 GB. Indique de forma razonada:

a) El formato de la dirección virtual.

b) El número máximo de páginas en este computador.

c) El número de marcos de página de este computador.

d) El tamaño del bloque que se transfiere entre disco y memoria principal cuando ocurre un fallo de página

4. Gestión de la entrada/salida (E/S).

Anteriormente, vimos que una de las funciones del ordenador era procesar la información, dicha información la obtiene y muestra a través de los periféricos. **La parte del sistema operativo que se encarga de este proceso es la gestión de la E/S (entrada/salida).** En la primera unidad estudiamos los periféricos y recordamos que se clasificaban en periféricos:

- De entrada
- De salida
- De entrada y salida

El sistema operativo hace que los dispositivos se conecten al sistema y realicen sus funciones de forma adecuada y eficiente. El sistema operativo abstrae de la complejidad y peculiaridad hardware de cada periférico para que las aplicaciones de usuario puedan hacer uso de los periféricos de una manera estandarizada y más sencilla. El sistema operativo actúa pues como intermediario entre ellos, gracias a los controladores de dispositivo.

¿Cómo pueden entenderse los programas de aplicación con los dispositivos periféricos? Hay multitud de tipos y fabricantes de periféricos, esto conlleva que tanto el sistema operativo como los fabricantes de periféricos deben estandarizar el acceso a los dispositivos utilizando lo que se denominan controladores de dispositivos (device drivers).

Un periférico siempre tiene dos partes: un controlador, se encarga de la comunicación con la CPU y un dispositivo mecánico, electromecánico o electromagnético. El controlador es un software, generalmente, suministrado por el fabricante del dispositivo o bien por el desarrollador del sistema operativo. De esta manera, estos controladores actúan como interfaz entre los programas y el hardware.

Las técnicas más utilizadas por los sistemas operativos para gestionar las entradas / salidas son dos:

Spools: Los datos de salida se almacenan de forma temporal en una cola situada en un dispositivo de almacenamiento masivo (spool), hasta que el dispositivo periférico requerido se encuentre libre .De este modo se evita que un programa quede retenido porque el periférico no esté disponible. El sistema operativo dispone de llamadas para añadir y eliminar archivos del spool. Se utiliza en dispositivos que no admiten intercalación, como ocurre en la impresora, ya que no puede empezar con otro hasta que no ha terminado.

Buffers: Es para dispositivos que pueden atender peticiones de distintos orígenes. En este caso. Los datos no tienen que enviarse completos, pueden enviarse porciones que el buffer retiene de forma temporal. También se utilizan para acoplar velocidades de distintos dispositivos. Así, si un dispositivo lento va a recibir información más rápido de lo que puede atenderla se emplea un buffer para retener temporalmente la

información hasta que el dispositivo pueda asimilarla. Esto ocurre entre una grabadora de DVD y el disco duro, ya que la primera funciona a una menor velocidad que el segundo.

Las distintas formas de funcionamiento de la E/S en los sistemas operativos según la intervención de la CPU tenemos:

E/S programada: la CPU tiene todo el protagonismo ya que inicia y lleva a cabo la transferencia. Esta técnica repercute en la velocidad de proceso del ordenador porque la CPU debe dejar todo lo que está haciendo para ocuparse del proceso de entrada/salida.

E/S por interrupciones: la CPU ejecuta la transferencia, pero el inicio es pedido por el periférico que indica así su disponibilidad. La CPU no pregunta a los dispositivos, sino que son estos los que la avisan cuando es necesario.

Acceso directo a memoria (DMA): la transferencia es realizada por un controlador especializado. Esta técnica acelera enormemente el proceso de la E/S y libera a la CPU de trabajo. Lo habitual es que los datos que se quieren escribir en el dispositivo o que son leídos del dispositivo provengan o vayan a la memoria del ordenador, pues bien, en este caso. la CPU inicia el proceso, pero luego este continúa sin necesitar a la CPU, con lo que se acelera mucho el proceso de entrada/salida y se libera a la CPU del proceso.

5. Gestión del sistema de archivos.

Esta parte del sistema operativo gestiona el servicio de almacenamiento, por lo que permite crear, modificar, borrar archivos y directorios y para ello utiliza generalmente una estructura jerárquica.

Cada sistema operativo utilizará su propio sistema de archivos, no obstante, las operaciones que se pueden realizar sobre el sistema de archivos son bastante similares. Así, todos los sistemas de archivos actuales utilizan los directorios o carpetas para organizar a los archivos.

El sistema de archivos es el software que provee al sistema operativo, a los programas de aplicación y a usuarios de las funciones para operar con archivos y directorios almacenados en disco proporcionando mecanismos de protección y seguridad.

Los objetivos más importantes en la implementación de un sistema de archivos son:

- Optimizar el rendimiento mediante un acceso rápido para recuperar la información contenida en archivos.
- Fácil actualización: Los cambios (añadir, borrar y modificar) no deben suponer una tarea complicada para el usuario y las aplicaciones.
- Economía de almacenamiento: Intentar que los archivos desperdicien la menor cantidad de espacio en disco posible. Es muy importante evitar la fragmentación de los discos.
- Mantenimiento sencillo: Evitar las operaciones complicadas a usuarios y programas, ocultando los detalles y proporcionando un acceso estandarizado a los archivos.
- Fiabilidad para asegurar la confianza en los datos: Deben proveer sistemas que aseguren que los datos escritos o leídos (entradas/salidas) sean correctos y fiables. También se debe minimizar o eliminar la posibilidad de pérdida o destrucción de datos.
- Incorporar mecanismos de seguridad y permisos: Esto es especialmente importante en sistemas de archivos de sistemas operativos multiusuario. Se debe poder proteger los archivos de un usuario del acceso de los demás usuarios. Por ejemplo, estableciendo permisos de escritura, lectura o ejecución.
- Control de concurrencia: Se debe controlar y asegurar el acceso correcto a los archivos por parte de varios usuarios a un tiempo, posiblemente bloqueando el archivo en uso hasta que termine la operación de modificación en curso.

6. Mecanismos de seguridad y protección.

El sistema operativo debe protegerse activamente a sí mismo y a los usuarios de acciones accidentales o malintencionadas. Cada vez es más necesaria la seguridad en los sistemas, ya que actualmente la mayoría de los ordenadores se encuentran conectados en red y el número de usuarios y recursos compartidos ha aumentado considerablemente.

Vamos a diferenciar entre seguridad y protección. Por seguridad nos referimos a una política donde se deciden qué accesos están permitidos, qué usuarios pueden acceder, en que forma y a qué recursos. Por otro lado, la protección hace referencia al mecanismo que se utiliza para llevar a cabo la política de seguridad.

Los requisitos que debe cumplir un sistema operativo son:

Confidencialidad : Los elementos del sistema sólo serán visibles por aquellos usuarios o grupos autorizados.

Integridad : Los elementos del sistema sólo serán modificados por los usuarios o grupos autorizados.

Disponibilidad : Los elementos del sistema sólo estarán disponibles para usuarios y grupos autorizados.

Para hacer frente a estas acciones el sistema operativo agrupa la seguridad según tres aspectos:

1. Seguridad en el uso de recursos y servicios y control de acceso : Utilizar un mecanismo de control de acceso a los recursos que tan sólo permita el acceso si existe el permiso correspondiente. Se establecerán políticas de permisos para acceder y operar con recursos y servicios.

2. Seguridad en el acceso al sistema : Asegurar que sólo entran los usuarios autorizados. Para ello podrán utilizarse un sistema de contraseñas eficaz con niveles de acceso diferentes.

3. Seguridad en el uso de redes : Evitar que se puedan producir escuchas y alteraciones en los datos que viajan por la red. Se aplicarán técnicas de cifrado y descifrado de las comunicaciones a través de la red.

7. Documentación y búsqueda de información técnica.

Todo software con una cierta complejidad suele venir acompañado de una documentación, ésta puede ser en formato digital o papel. Esta documentación toma forma en manuales, tutoriales y demás guías de referencia que sirven para mostrar al usuario cómo se implanta y utiliza una aplicación. A continuación, veremos los tipos de documentación nos podemos encontrar:

Manual de usuario (con distintos niveles: básico, intermedio, avanzado)

Manual de Instalación y Configuración del programa

Manual del Administrador

Guía de referencia rápida

Existe la posibilidad de buscar información adicional utilizando otros medios, como por ejemplo:

- Consulta al soporte técnico del desarrollador de software, vía web, email o teléfono.
- Consulta en foros de expertos.
- Consulta en bases de conocimiento.
- Consulta en FAQs (Frequently Asked Questions Preguntas Frecuentes).