Núcleo (informática)

*Para otros usos de este término, véase*[*Núcleo*](http://es.wikipedia.org/wiki/N%C3%BAcleo)*.*

En [informática](http://es.wikipedia.org/wiki/Inform%C3%A1tica), un **núcleo** o **kernel** (de la raíz [germánica](http://es.wikipedia.org/wiki/Idioma_alem%C3%A1n) *Kern*) es un [software](http://es.wikipedia.org/wiki/Software) que constituye la parte más importante del [sistema operativo](http://es.wikipedia.org/wiki/Sistema_operativo).[1](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#cite_note-0) Es el principal responsable de facilitar a los distintos programas [acceso seguro](http://es.wikipedia.org/wiki/Seguridad_inform%C3%A1tica) al [hardware](http://es.wikipedia.org/wiki/Hardware) de la [computadora](http://es.wikipedia.org/wiki/Computadora) o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo, lo que se conoce como [multiplexado](http://es.wikipedia.org/wiki/Multiplexaci%C3%B3n). Acceder al hardware directamente puede ser realmente complejo, por lo que los núcleos suelen implementar una serie de [abstracciones del hardware](http://es.wikipedia.org/wiki/Abstracci%C3%B3n_del_hardware). Esto permite esconder la complejidad, y proporciona una interfaz limpia y uniforme al hardware subyacente, lo que facilita su uso al [programador](http://es.wikipedia.org/wiki/Programador).

|  |
| --- |
| **Contenido** [[ocultar](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29)]* [1 Técnica](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#T.C3.A9cnica)
* [2 Generalidades](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Generalidades)
	+ [2.1 Funciones generalmente ejercidas por un núcleo](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Funciones_generalmente_ejercidas_por_un_n.C3.BAcleo)
	+ [2.2 Unix](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Unix)
* [3 Tipos de núcleo](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Tipos_de_n.C3.BAcleo)
	+ [3.1 Micronúcleos](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Micron.C3.BAcleos)
	+ [3.2 Núcleos monolíticos en contraposición a micronúcleos](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#N.C3.BAcleos_monol.C3.ADticos_en_contraposici.C3.B3n_a_micron.C3.BAcleos)
	+ [3.3 Núcleos híbridos (micronúcleos modificados)](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#N.C3.BAcleos_h.C3.ADbridos_.28micron.C3.BAcleos_modificados.29)
	+ [3.4 Exonúcleos](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Exon.C3.BAcleos)
* [4 Referencias](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Referencias)
* [5 Véase también](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#V.C3.A9ase_tambi.C3.A9n)
* [6 Enlaces externos](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#Enlaces_externos)
 |

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=1)]Técnica

Cuando se aplica voltaje al [procesador](http://es.wikipedia.org/wiki/Unidad_central_de_procesamiento) de un dispositivo electrónico, éste ejecuta un reducido código en [lenguaje ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador) localizado en una dirección concreta en la memoria [ROM](http://es.wikipedia.org/wiki/ROM) (dirección de reset) y conocido como *reset code*, que a su vez ejecuta una [rutina](http://es.wikipedia.org/wiki/Subrutina) con la que se inicializa el hardware que acompaña al procesador. También en esta fase suele inicializarse el controlador de las [interrupciones](http://es.wikipedia.org/wiki/Interrupci%C3%B3n). Finalizada esta fase se ejecuta el código de arranque (*startup code*), también código en lenguaje ensamblador, cuya tarea más importante es ejecutar el programa principal (*main()*) del [software de la aplicación](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_%28inform%C3%A1tica%29).[2](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#cite_note-programming_es-1)

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=2)]Generalidades





Una visión típica de la [arquitectura de computadores](http://es.wikipedia.org/wiki/Arquitectura_de_computadores) como una serie de capas de abstracción: [hardware](http://es.wikipedia.org/wiki/Hardware), [firmware](http://es.wikipedia.org/wiki/Firmware),[lenguaje de máquina](http://es.wikipedia.org/wiki/Lenguaje_de_m%C3%A1quina), núcleo, [software de sistema](http://es.wikipedia.org/wiki/Software_de_sistema) y [aplicaciones](http://es.wikipedia.org/wiki/Proceso_%28inform%C3%A1tica%29)[3](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#cite_note-2)

En informática, el núcleo es el programa informático que se asegura de:

* La comunicación entre los programas que solicitan recursos y el [hardware](http://es.wikipedia.org/wiki/Hardware).
* Gestión de los distintos programas informáticos (tareas) de una máquina.
* Gestión del hardware ([memoria](http://es.wikipedia.org/wiki/Memoria_%28inform%C3%A1tica%29), [procesador](http://es.wikipedia.org/wiki/CPU), [periférico](http://es.wikipedia.org/wiki/Perif%C3%A9rico), forma de almacenamiento, etc.)

La mayoría de las [interfases de usuario](http://es.wikipedia.org/wiki/Interfaz_de_usuario) se construyen en torno al concepto de núcleo. La existencia de un núcleo, es decir, de un único programa responsable de la comunicación entre el hardware y el programa informático, resulta de compromisos complejos referentes a cuestiones de resultados, seguridad y arquitectura de los procesadores. El núcleo tiene grandes poderes sobre la utilización de los recursos materiales (hardware), en particular, de la memoria.

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=3)]**Funciones generalmente ejercidas por un núcleo**

Los núcleos tienen como funciones básicas garantizar la carga y la ejecución de los procesos, las entradas/salidas y proponer una interfaz entre el espacio núcleo y los programas del espacio del usuario.

Aparte de las funcionalidades básicas, el conjunto de las funciones de los puntos siguientes (incluidos los pilotos materiales, las funciones de redes y sistemas de ficheros o los servicios) necesariamente no son proporcionados por un núcleo de sistema de explotación. Pueden establecerse estas funciones del sistema de explotación tanto en el espacio usuario como en el propio núcleo. Su implantación en el núcleo se hace en el único objetivo de mejorar los resultados. En efecto, según la concepción del núcleo, la misma función llamada desde el espacio usuario o el espacio núcleo tiene un coste temporal obviamente diferente. Si esta llamada de función es frecuente, puede resultar útil integrar estas funciones al núcleo para mejorar los resultados.

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=4)]**Unix**

Un núcleo Unix es un programa escrito casi en su totalidad en [lenguaje C](http://es.wikipedia.org/wiki/Lenguaje_C), con excepción de una parte del manejo de interrupciones, expresada en el [lenguaje ensamblador](http://es.wikipedia.org/wiki/Lenguaje_ensamblador) del procesador en el que opera. Las funciones del núcleo son permitir la existencia de un ambiente en el que sea posible atender a varios usuarios y [múltiples tareas en forma concurrente](http://es.wikipedia.org/wiki/Multitarea), repartiendo al procesador entre todos ellos, e intentando mantener en grado óptimo la atención individual.

El núcleo opera como asignador de recursos para cualquier proceso que necesite hacer uso de las facilidades de cómputo.

* Creación de procesos, asignación de tiempos de atención y sincronización.
* Asignación de la atención del procesador a los procesos que lo requieren.
* Administración de espacio en el sistema de archivos, que incluye: acceso, protección y administración de usuarios; comunicación entre usuarios y entre procesos, y manipulación de E/S y administración de periféricos.
* Supervisión de la transmisión de datos entre la memoria principal y los dispositivos periféricos.

Reside siempre en la memoria principal y tiene el control sobre la computadora, por lo que ningún otro proceso puede interrumpirlo; sólo pueden llamarlo para que proporcione algún servicio de los ya mencionados. Un proceso llama al núcleo mediante módulos especiales conocidos como llamadas al sistema.

Consta de dos partes principales: la sección de control de procesos y la de control de dispositivos. La primera asigna recursos, programas, procesos y atiende sus requerimientos de servicio; la segunda, supervisa la transferencia de datos entre la memoria principal y los dispositivos del ordenador. En términos generales, cada vez que algún usuario oprime una tecla de una [terminal](http://es.wikipedia.org/wiki/Terminal_de_computadora), o que se debe leer o escribir información del disco magnético, se interrumpe al procesador central y el núcleo se encarga de efectuar la operación de transferencia.

Cuando se inicia la operación de la computadora, debe cargarse en la memoria una copia del núcleo, que reside en el disco magnético (operación denominada bootstrap). Para ello, se deben inicializar algunas interfaces básicas de hardware; entre ellas, el reloj que proporciona interrupciones periódicas. El núcleo también prepara algunas estructuras de datos que abarcan una sección de almacenamiento temporal para transferencia de información entre terminales y procesos, una sección para almacenamiento de descriptores de archivos y una variable que indica la cantidad de memoria principal.

A continuación, el núcleo inicializa un proceso especial, llamado proceso 0. En [Unix](http://es.wikipedia.org/wiki/Unix), los procesos se crean mediante una llamada a una rutina del sistema (fork), que funciona por un mecanismo de duplicación de procesos. Sin embargo, esto no es suficiente para crear el primero de ellos, por lo que el núcleo asigna una estructura de datos y establece apuntadores a una sección especial de la memoria, llamada tabla de procesos, que contendrá los descriptores de cada uno de los procesos existentes en el sistema.

Después de haber creado el proceso 0, se hace una copia del mismo, con lo que se crea el proceso 1; éste muy pronto se encargará de "dar vida" al sistema completo, mediante la activación de otros procesos que también forman parte del núcleo. Es decir, se inicia una cadena de activaciones de procesos, entre los cuales destaca el conocido como despachador, o planificador, que es el responsable de decidir cuál proceso se ejecutará y cuáles van a entrar o salir de la memoria central. A partir de ese momento se conoce el número 1 como proceso de inicialización del sistema, init.

El proceso init es el responsable de establecer la estructura de procesos en Unix. Normalmente, es capaz de crear al menos dos estructuras distintas de procesos: el modo monousuario y el multiusuario. Comienza activando el intérprete del lenguaje de control [Shell de Unix](http://es.wikipedia.org/wiki/Shell_de_Unix) en la terminal principal, o consola del sistema, proporcionándole privilegios de "superusuario". En la modalidad de un solo usuario la consola permite iniciar una primera sesión, con privilegios especiales, e impide que las otras líneas de comunicación acepten iniciar sesiones nuevas. Esta modalidad se usa con frecuencia para revisar y reparar sistemas de archivos, realizar pruebas de funciones básicas del sistema y para otras actividades que requieren uso exclusivo de la computadora.

Init crea otro proceso, que espera a que alguien entre en sesión en alguna línea de comunicación. Cuando esto sucede, realiza ajustes en el protocolo de la línea y ejecuta el programa login, que se encarga de atender inicialmente a los nuevos usuarios. Si el nombre de usuario y la contraseña proporcionadas son correctos, entonces entra en operación el programa Shell, que en lo sucesivo se encargará de la atención normal del usuario que se dio de alta en esa terminal.

A partir de ese momento el responsable de atender al usuario en esa terminal es el intérprete Shell. Cuando se desea terminar la sesión hay que desconectarse de Shell (y, por lo tanto, de Unix), mediante una secuencia especial de teclas (usualmente. < CTL > - D). A partir de ese momento la terminal queda disponible para atender a un nuevo usuario.

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=5)]Tipos de núcleo

No necesariamente se necesita un núcleo para usar una computadora. Los programas pueden cargarse y ejecutarse directamente en una computadora «vacía», siempre que sus autores quieran desarrollarlos sin usar ninguna [abstracción](http://es.wikipedia.org/wiki/Abstracci%C3%B3n_%28informatica%29) del hardware ni ninguna ayuda del sistema operativo. Ésta era la forma normal de usar muchas de las primeras computadoras: para usar distintos programas se tenía que reiniciar y reconfigurar la computadora cada vez. Con el tiempo, se empezó a dejar en memoria (aún entre distintas ejecuciones) pequeños programas auxiliares, como el [cargador](http://es.wikipedia.org/wiki/Cargador_de_programas) y el [depurador](http://es.wikipedia.org/wiki/Depurador), o se cargaban desde [memoria de sólo lectura](http://es.wikipedia.org/wiki/Memoria_ROM). A medida que se fueron desarrollando, se convirtieron en los fundamentos de lo que llegarían a ser los primeros núcleos de sistema operativo.

Hay cuatro grandes tipos de núcleos:

* Los **núcleos monolíticos** facilitan abstracciones del hardware subyacente realmente potentes y variadas.
* Los **micronúcleos** (en inglés **microkernel**) proporcionan un pequeño conjunto de abstracciones simples del hardware, y usan las aplicaciones llamadas [servidores](http://es.wikipedia.org/wiki/Servidor) para ofrecer mayor funcionalidad.[4](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#cite_note-3)
* Los **núcleos híbridos** (*micronúcleos modificados*) son muy parecidos a los micronúcleos puros, excepto porque incluyen código adicional en el espacio de núcleo para que se ejecute más rápidamente.
* Los **exonúcleos** no facilitan ninguna abstracción, pero permiten el uso de [bibliotecas](http://es.wikipedia.org/wiki/Biblioteca_%28programa%29) que proporcionan mayor funcionalidad gracias al acceso directo o casi directo al hardware.

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=6)]**Micronúcleos**





Esquema del funcionamiento de un micronúcleo.

*Artículo principal:*[*Micronúcleo*](http://es.wikipedia.org/wiki/Micron%C3%BAcleo)

El enfoque micronúcleo consiste en definir una abstracción muy simple sobre el hardware, con un conjunto de primitivas o llamadas al sistema que implementan servicios del sistema operativo mínimos, como la [gestión de hilos](http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_hilos), el [espacio de direccionamiento](http://es.wikipedia.org/w/index.php?title=Espacio_de_direccionamiento&action=edit&redlink=1) y la[comunicación entre procesos](http://es.wikipedia.org/wiki/Comunicaci%C3%B3n_entre_procesos).

El objetivo principal es la separación de la implementación de los servicios básicos y de la política de funcionamiento del sistema. Por ejemplo, el proceso de bloqueo de [E/S](http://es.wikipedia.org/wiki/Entrada/salida) se puede implementar con un servidor en espacio de usuario ejecutándose encima del micronúcleo. Estos servidores de usuario, utilizados para gestionar las partes de alto nivel del sistema, son muy modulares y simplifican la estructura y diseño del núcleo. Si falla uno de estos servidores, no se colgará el sistema entero, y se podrá reiniciar este módulo independientemente del resto. Sin embargo, la existencia de diferentes módulos independientes origina retardos en la comunicación debido a la copia de variables que se realiza en la comunicación entre módulos.

Algunos ejemplos de micronúcleos:

* [AIX](http://es.wikipedia.org/wiki/AIX)
* La familia de micronúcleos [L4](http://es.wikipedia.org/wiki/L4_%28microkernel%29)
* El micronúcleo [Mach](http://es.wikipedia.org/wiki/GNU_Mach), usado en [GNU](http://es.wikipedia.org/wiki/GNU) [Hurd](http://es.wikipedia.org/wiki/Hurd) y en [Mac OS X](http://es.wikipedia.org/wiki/Mac_OS_X)
* [BeOS](http://es.wikipedia.org/wiki/BeOS)
* [Minix](http://es.wikipedia.org/wiki/Minix)
* [MorphOS](http://es.wikipedia.org/wiki/MorphOS)
* [QNX](http://es.wikipedia.org/wiki/QNX)
* [RadiOS](http://es.wikipedia.org/wiki/RadiOS)
* [VSTa](http://es.wikipedia.org/wiki/VSTa)
* [Hurd](http://es.wikipedia.org/wiki/Hurd)

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=7)]**Núcleos monolíticos en contraposición a micronúcleos**

*Artículos principales:*[*Núcleo monolítico*](http://es.wikipedia.org/wiki/N%C3%BAcleo_monol%C3%ADtico)*y*[*Micronúcleo*](http://es.wikipedia.org/wiki/Micron%C3%BAcleo)

Frecuentemente se prefieren los [núcleos monolíticos](http://es.wikipedia.org/wiki/N%C3%BAcleo_monol%C3%ADtico) frente a los micronúcleos debido al menor nivel de complejidad que comporta el tratar con todo el código de control del sistema en un solo espacio de direccionamiento. Por ejemplo, [XNU](http://es.wikipedia.org/wiki/XNU), el núcleo de Mac OS X, está basado en el núcleo Mach 3.0 y en FreeBSD, en el mismo espacio de direccionamiento para disminuir la latencia que comporta el diseño de micronúcleo convencional.

A principios de los [años 90](http://es.wikipedia.org/wiki/A%C3%B1os_1990), los núcleos monolíticos se consideraban obsoletos. El diseño de Linux como un núcleo monolítico en lugar de como un micronúcleo fue el tema de una famosa disputa[5](http://es.wikipedia.org/wiki/N%C3%BAcleo_%28inform%C3%A1tica%29#cite_note-4) entre [Linus Torvalds](http://es.wikipedia.org/wiki/Linus_Torvalds) y [Andrew Tanenbaum](http://es.wikipedia.org/wiki/Andrew_Tanenbaum). Los argumentos de ambas partes en esta discusión presentan algunas motivaciones interesantes.

Los núcleos monolíticos suelen ser más fáciles de diseñar correctamente, y por lo tanto pueden crecer más rápidamente que un sistema basado en micronúcleo, pero hay casos de éxito en ambos bandos. Los micronúcleos suelen usarse en robótica embebida o computadoras médicas, ya que la mayoría de los componentes del sistema operativo residen en su propio espacio de memoria privado y protegido. Esto no sería posible con los núcleos monolíticos, ni siquiera con los modernos que permiten cargar [módulos del núcleo](http://es.wikipedia.org/wiki/M%C3%B3dulo_de_n%C3%BAcleo).

Aunque [Mach](http://es.wikipedia.org/wiki/Mach_kernel) es el micronúcleo generalista más conocido, se han desarrollado otros micronúcleos con propósitos más específicos. [L3](http://es.wikipedia.org/w/index.php?title=L3_(microkernel)&action=edit&redlink=1) fue creado para demostrar que los micronúcleos no son necesariamente lentos. La familia de micronúcleos L4 es la descendiente de L3, y una de sus últimas implementaciones, llamada [Pistachio](http://es.wikipedia.org/w/index.php?title=Pistachio_(microkernel_L4)&action=edit&redlink=1), permite ejecutar Linux simultáneamente con otros procesos, en espacios de direccionamiento separados.

[QNX](http://es.wikipedia.org/wiki/QNX) es un sistema operativo que ha estado disponible desde principios de los [años 80](http://es.wikipedia.org/wiki/A%C3%B1os_1980), y tiene un diseño de micronúcleo muy minimalista. Este sistema ha conseguido llegar a las metas del paradigma del micronúcleo con mucho más éxito que Mach. Se usa en situaciones en que no se puede permitir que haya fallos de software, lo que incluye desde brazos robóticos en naves espaciales, hasta máquinas que pulen cristal donde un pequeño error podría costar mucho dinero.

Mucha gente cree que como Mach básicamente falló en el intento de resolver el conjunto de problemas que los micronúcleos intentaban subsanar, toda la tecnología de micronúcleos es inútil. Los partidarios de Mach afirman que ésta es una actitud estrecha de miras que ha llegado a ser lo suficientemente popular para que mucha gente la acepte como verdad.

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=8)]**Núcleos híbridos (micronúcleos modificados)**

*Artículo principal:*[*Núcleo híbrido*](http://es.wikipedia.org/wiki/N%C3%BAcleo_h%C3%ADbrido)

Los núcleos híbridos fundamentalmente son micronúcleos que tienen algo de código «no esencial» en espacio de núcleo para que éste se ejecute más rápido de lo que lo haría si estuviera en espacio de usuario. Éste fue un compromiso que muchos desarrolladores de los primeros sistemas operativos con arquitectura basada en micronúcleo adoptaron antes que se demostrara que los micronúcleos pueden tener muy buen rendimiento. La mayoría de sistemas operativos modernos pertenecen a esta categoría, siendo el más popular [Microsoft Windows](http://es.wikipedia.org/wiki/Microsoft_Windows). XNU, el núcleo de Mac OS X, también es un micronúcleo modificado, debido a la inclusión de código del núcleo de FreeBSD en el núcleo basado en Mach. [DragonFlyBSD](http://es.wikipedia.org/wiki/DragonFlyBSD) es el primer sistema BSD que adopta una arquitectura de núcleo híbrido sin basarse en Mach.

Algunos ejemplos de núcleos híbridos:

* [Microsoft](http://es.wikipedia.org/wiki/Microsoft) [Windows NT](http://es.wikipedia.org/wiki/Windows_NT), usado en todos los sistemas que usan el código base de Windows NT
* [XNU](http://es.wikipedia.org/wiki/XNU) (usado en [Mac OS X](http://es.wikipedia.org/wiki/Mac_OS_X))
* [DragonFlyBSD](http://es.wikipedia.org/wiki/DragonFlyBSD)
* [ReactOS](http://es.wikipedia.org/wiki/ReactOS)

Hay gente que confunde el término «núcleo híbrido» con los núcleos monolíticos que pueden cargar módulos después del arranque, lo que es un error. «Híbrido» implica que el núcleo en cuestión usa conceptos de arquitectura o mecanismos tanto del diseño monolítico como del micronúcleo, específicamente el paso de mensajes y la migración de código «no esencial» hacia el espacio de usuario, pero manteniendo cierto código «no esencial» en el propio núcleo por razones de rendimiento.

[[editar](http://es.wikipedia.org/w/index.php?title=N%C3%BAcleo_(inform%C3%A1tica)&action=edit&section=9)]**Exonúcleos**

*Artículo principal:*[*Exonúcleo*](http://es.wikipedia.org/wiki/Exon%C3%BAcleo)





Esquema básico de la interacción entre un Exonúcleo (Exokernel) con el[Software](http://es.wikipedia.org/wiki/Software) a través de [bibliotecas](http://es.wikipedia.org/wiki/Biblioteca_%28programaci%C3%B3n%29).

Los exonúcleos, también conocidos como sistemas operativos verticalmente estructurados, representan una aproximación radicalmente nueva al diseño de sistemas operativos.

La idea subyacente es permitir que el desarrollador tome todas las decisiones relativas al rendimiento del hardware. Los exonúcleos son extremadamente pequeños, ya que limitan expresamente su funcionalidad a la protección y el multiplexado de los recursos. Se llaman así porque toda la funcionalidad deja de estar residente en memoria y pasa a estar fuera, en [bibliotecas](http://es.wikipedia.org/wiki/Biblioteca_%28inform%C3%A1tica%29)dinámicas.

Los diseños de núcleos clásicos (tanto el monolítico como el micronúcleo) abstraen el hardware, escondiendo los recursos bajo una capa de abstracción del hardware, o detrás de los controladores de dispositivo. En los sistemas clásicos, si se asigna memoria física, nadie puede estar seguro de cuál es su localización real, por ejemplo.

La finalidad de un exonúcleo es permitir a una aplicación que solicite una región específica de la memoria, un bloque de disco concreto, etc., y simplemente asegurarse que los recursos pedidos están disponibles, y que el programa tiene derecho a acceder a ellos.

Debido a que el exonúcleo sólo proporciona una interfaz al hardware de muy bajo nivel, careciendo de todas las funcionalidades de alto nivel de otros sistemas operativos, éste es complementado por una «biblioteca de sistema operativo». Esta biblioteca se comunica con el exonúcleo subyacente, y facilita a los programadores de aplicaciones las funcionalidades que son comunes en otros sistemas operativos.

Algunas de las implicaciones teóricas de un sistema exonúcleo son que es posible tener distintos tipos de sistemas operativos (p.e. Windows, Unix) ejecutándose en un solo exonúcleo, y que los desarrolladores pueden elegir prescindir ó incrementar funcionalidades por motivos de rendimiento.

Actualmente, los diseños exonúcleo están fundamentalmente en fase de estudio y no se usan en ningún sistema popular. Un concepto de sistema operativo es [Nemesis](http://es.wikipedia.org/w/index.php?title=Nemesis_(inform%C3%A1tica)&action=edit&redlink=1), creado por la [Universidad de Cambridge](http://es.wikipedia.org/wiki/Universidad_de_Cambridge), la [Universidad de Glasgow](http://es.wikipedia.org/wiki/Universidad_de_Glasgow), [Citrix Systems](http://es.wikipedia.org/wiki/Citrix_Systems) y el [Instituto Sueco de Informática](http://es.wikipedia.org/w/index.php?title=Instituto_Sueco_de_Inform%C3%A1tica&action=edit&redlink=1). El [MIT](http://es.wikipedia.org/wiki/Instituto_Tecnol%C3%B3gico_de_Massachusetts) también ha diseñado algunos sistemas basados en exonúcleos. Los exonúcleos se manejan en diferente estructura dado que también cumplen funciones distintas