

Micronúcleos

Mariano Street

4 de junio de 2019

¿Qué es un núcleo?

¿Qué es el núcleo de un sistema operativo?

¿Qué es un núcleo?

- Pieza fundamental de un SO.
- Administra el hardware. Provee acceso al mismo de forma compartida y segura.
- Multiplexa el hardware, de forma que múltiples procesos puedan acceder a un mismo dispositivo como si en realidad cada uno tuviera el suyo.
- Lo primero que se carga al arrancar un SO (después del gestor de arranque).

Núcleo y programas de usuario

¿Qué distingue a un núcleo de los demás programas? ¿No podría cualquier otro programa administrar también el hardware?

- El núcleo funciona como intermediario entre los otros programas y el hardware.
- Muchas arquitecturas modernas ofrecen protección.
- 2 modos o espacios de ejecución:

Modo/Espacio de núcleo o de supervisor

Privilegiado. Con acceso libre a todo el hardware. Aquí corre el núcleo.

Modo/Espacio de usuario

Restringido. Sin acceso al hardware. Los programas de usuario corren acá.

Por lo general, el núcleo es un solo programa. Todos sus componentes comparten la misma memoria.

Al principio se trataba de programas chicos.

Para los '80, ya estaban creciendo demasiado: controladores de dispositivos, protocolos de Internet, sistemas de archivos...

Problemas: mantenibilidad, estabilidad, seguridad.

Disyuntiva: ¿qué cosas exactamente forman parte del núcleo? En otras palabras, ¿qué va en espacio de núcleo y qué en espacio de usuario?

La idea de los micronúcleos (en inglés, *microkernels*) es, en principio, que todo lo posible vaya en espacio de usuario, que se usen abstracciones de hardware simples y que el núcleo sea mínimo. Que el núcleo implemente mecanismos, no políticas.

Mínimamente, se necesita tener:

1. Alguna gestión de procesos.
2. Manejo básico de espacios de direcciones.
3. Comunicación entre procesos (IPC); generalmente pasaje de mensajes.

Dos estructuras clásicas alternativas a los micronúcleos:

Núcleo monolítico

Separa espacio de núcleo y espacio de usuario, protege la memoria entre procesos y la del núcleo en su conjunto, pero incorpora dentro de sí todas las funciones comunes de un sistema. Así son los núcleos de la familia Unix, incluido Linux.

Núcleo sin protección

No hace distinción entre espacio de núcleo y espacio de usuario, no hay protección de memoria y los procesos pueden acceder directamente al hardware. Así es la familia DOS.

¿A qué estructura corresponde Nachos?

Casos históricos emblemáticos de micronúcleos:

- 1969: RC 4000 (Regnecentralen, Dinamarca), precursor
 - Per Brinch Hansen
- 1975: Aleph (Universidad de Rochester, EEUU)
- 1981: Accent (Universidad de Carnegie Mellon, EEUU)
- 1985: Mach (Universidad de Carnegie Mellon, EEUU)

En los '80 eran un tema muy popular en la academia.

Algunos micronúcleos que se usan hoy:

- Mach, micronúcleo insignia de la primera generación
- L4, micronúcleo insignia de la segunda generación
- MINIX, sistema para educación, dispositivos embebidos y tiempo real
- QNX, sistema de tiempo real

- La primera generación de micronúcleos.
- Desarrollado en los '80 por CMU (Carnegie Mellon University de EEUU).
- Está basado en BSD (4.2BSD y 4.3BSD). El núcleo de BSD es monolítico; la idea era transicionar progresivamente hacia un micronúcleo.
- Paginador externo (en espacio de usuario).
- Es la base de GNU Hurd.
- Es la base de XNU.

Cuando dejó de ser tan monolítico y adoptar el diseño de micronúcleo, Mach pasó a tener un mal rendimiento. Algunos factores:

- Conjunto de trabajo grande, poca localidad espacial: el código para IPC es grande y disperso y genera muchos fallos de caché.
- Muchas copias de datos de proceso a núcleo y viceversa.
- Muchas validaciones y comprobaciones de permisos para cada mensaje.
- Portabilidad. No está optimizado para una arquitectura en particular.

- Proyecto GNU: todo un SO de tipo UNIX que fuera libre.
- Arrancó en 1983 y para 1989 tenían todo lo esencial salvo el núcleo.
- En 1991 anunciaron un micronúcleo: Hurd.
- Está basado en Mach 3.
- La idea estaba desde mucho antes pero se retrasaron por incertidumbre sobre licencias.
- Enseguida apareció Linux y ocupó su lugar, conformando el SO libre completo GNU/Linux.
- El desarrollo decayó mucho aunque siguió, y en los últimos años repuntó un poco.
- Algunas distribuciones lo ofrecen: Debian GNU/Hurd, Arch Hurd, GNU GuixSD.

Mach tenía un rendimiento pobre. Hurd no se materializaba. IBM tuvo un proyecto Workplace OS que fue demasiado ruido y nulas nueces.

En este contexto, en los '90 la popularidad de los micronúcleos se derrumbó. Se pasó a pensar que no funcionaban en la práctica.

- Hecho por Jochen Liedtke (Gesellschaft für Mathematik und Datenverarbeitung, Alemania). Se publicó en 1988.
- Más eficiente que Mach.
- Pensado desde cero para ser un micronúcleo.
- Específico para una arquitectura. Particularmente IA-32.
- Escrito en ensamblador.
- Sin comprobaciones de permisos a nivel del núcleo.

- Hecho por Liedtke en 1995. Sucesor de L3.
- La segunda generación de micronúcleos.
- Transferencia de mensaje corto (8 bytes):
 - CPU: Intel 80486DX-50 a 50 MHz con caché interna de 8 KiB y externa de 256 KiB.
 - En Mach, 115 μ s; en L3, 5.2 μ s. *¡22 veces más rápido!*
- Específico para IA-32.
- Diseño minimalista. 7 llamadas a sistema; (*¡menos que Nachos!*).
- Muchos derivados. Ahora es una familia de micronúcleos.

L4: algunas técnicas

- Mínima secuencia de llamadas a sistema para IPC: solo *call* y *receive & reply*. En caso de varias transferencias sucesivas, juntarlas en una.
- Copia de proceso a proceso, sin pasar por el núcleo. Ventana de comunicación: región de memoria compartida accesible solo por el núcleo.
- Hilos en modo supervisor se gestionan igual que en modo usuario. Una pila por cada uno, guardada en el mismo TCB.
- TCB guardados en un arreglo virtual en la memoria compartida.

L4: algunas técnicas (cont.)

- Planificación perezosa. Se evita encolar en lo posible.
- Cambios de contexto directos para IPC.
- Pasaje de datos por registros.
- Reducción de misses de caché, tanto de datos como de TLB.
- Uso óptimo de registros.
- Mínima cantidad de saltos y chequeos.
- Mínimo trabajo para cambiar de contexto.
- Y más...

L4: derivados

Principales derivados de L4:

L4Ka::Hazelnut En C++ con rendimiento aceptable.

L4Ka::Pistachio Portabilidad eficiente.

L4/Fiasco En C++, multitarea totalmente apropiativa, luego otras mejoras.

L4/MIPS Para la arquitectura MIPS.

L4/Alpha Para la arquitectura Alpha.

NICTA::L4-embedded Sistemas embebidos.

OKL4 Versión comercial.

seL4 Seguridad y fiabilidad, verificación formal con Haskell. Hay quienes lo consideran parte de una tercera generación.

L4Linux es una adaptación de Linux para ejecutarlo sobre L4.

- Es el núcleo de Darwin, desarrollado desde 1996.
- Darwin es la base de los SO de Apple (macOS, iOS y demás).
- Está basado en Mach 2.5, en el núcleo de 4.3BSD, en el de FreeBSD y en el de NextSTEP.
- Fue muy modificado, no es un micronúcleo sino un híbrido.

- Clon de UNIX.
- Incluye un micronúcleo.
- Desarrollado por Andrew Tanenbaum para enseñar SO (precursor de Nachos).
- Primera versión en 1987.
- A partir de la versión 3 (desde 2005), el fin no es solo educativo sino servir para dispositivos embebidos y de tiempo real.
- Fue la inspiración de Linus Torvalds para desarrollar Linux,
- Pero Torvalds no siguió la idea de los micronúcleos.
- Famoso debate entre Tanenbaum y Torvalds en 1992 y 2006.

Ventajas de micronúcleos

- Es más robusto.

Si hay un error en un componente del sistema, solo ese componente es afectado. Se lo puede reiniciar sin afectar a los demás.

Se facilita también la tarea de actualizar partes del sistema mientras está funcionando.

- Es más fácil de depurar y mantener. También puede ser más simple de entender.
- Resulta fácil de extender, muchas veces basta con agregar un proceso de usuario.
- Se pueden apilar más sistemas operativos sobre el micronúcleo, al fin y al cabo es todo espacio de usuario. Resulta ser como una máquina virtual.
- Se presta al armado de sistemas distribuidos.

Desventajas de micronúcleos

- Darle un rendimiento aceptable conlleva más esfuerzo que para un monolítico.
- Requiere pensar la arquitectura del software de antemano, es decir, qué clases de componentes va a haber y cómo van a interactuar entre sí.

- FUSE (Filesystem in Userspace) para Linux y otros sistemas tipo Unix.
- Módulos cargables y descargables en Linux y otros sistemas.
- NT (núcleo de Windows NT y sus sucesores).
- XNU (núcleo de Darwin, base de los sistemas de Apple).

Exokernel Abstracciones en bibliotecas, el núcleo solo resguarda la seguridad.

SPIN Módulos en lenguaje de programación gestionado por el núcleo (Modula-3).

- Liedtke. *Improving IPC by kernel design*. 1993.
- Liedtke. *Toward real microkernels*. 1996.
- Bugaev. *The three generations of microkernels*. 2019.
[https://plume.dyomedea.com/~3542@fediverse.blog/
the-three-generations-of-microkernels](https://plume.dyomedea.com/~3542@fediverse.blog/the-three-generations-of-microkernels)