

Micronúcleos: Una Alternativa para Implantar Sistemas Distribuidos

Tomás Balderas Contreras

balderas@ccc.inaoep.mx

Instituto Nacional de Astrofísica, Óptica y Electrónica
Coordinación de Ciencias de la Computación
Sistemas Distribuidos

26 de abril, 2003

Contenido

1	Introducción	1
2	Sistemas comerciales basados en micronúcleo	2
2.1	NeXTSTEP	3
2.2	MacOS X	3
2.3	UNICOS/mk	3
2.4	Tru64	3
3	Mach	3
3.1	Principios de diseño	3
3.2	Abstracciones	4
3.3	Comunicación entre procesos	5
3.4	Comunicación en ambientes distribuidos	5
3.5	Escalabilidad	6
4	Conclusiones	6

Resumen

Los micronúcleos surgieron de un esfuerzo de la comunidad de investigadores y desarrolladores de sistemas operativos para proponer una técnica de diseño e implantación diferente a las empleadas tradicionalmente en la construcción de sistemas operativos. Al mismo tiempo, los grandes avances en el área de sistemas distribuidos, resultado de la aparición de nuevas tecnologías y de extensas jornadas de trabajo de investigación, permitieron concebir ambientes de trabajo en los que es posible repartir la carga de procesamiento entre varios componentes, compartir de forma transparente distintos recursos, hacer uso de los mismos de forma más eficiente, establecer comunicaciones entre computadoras a lo largo y ancho del planeta y participar en grupos de trabajo colaborativo donde los participantes pueden encontrarse en distintas regiones geográficas.

Los micronúcleos pueden ser de gran utilidad para tratar el problema de garantizar ciertos requerimientos

en el diseño de sistemas distribuidos, tales como heterogeneidad, escalabilidad, transparencia y concurrencia. El objetivo de este documento es describir las características de un sistema operativo basado en micronúcleo y las ventajas que ofrece el empleo de esta tecnología al desarrollo de sistemas distribuidos.

PALABRAS CLAVE: Micronúcleo, Sistemas Distribuidos, Mach, Multiprocesadores, Sistemas Operativos.

1 Introducción

Alrededor del año 1969 Ken Thompson, desarrollador en los Laboratorios Bell de AT&T, inició el proyecto de implantación del sistema operativo UNIX, el cual ganó popularidad rápidamente entre desarrolladores de software y usuarios de computadoras, debido principalmente a sus características como sistema multiusuario y multitarea. Desde su creación, se ha invertido una gran cantidad de esfuerzo en la prueba, mejoramiento y extensión de las facilidades proporcionadas por UNIX. Inicialmente fue diseñado para minicomputadoras con poca memoria, sin subsistema de memoria virtual y paginación y sin soporte para redes; actualmente es ejecutado en sistemas multiprocesadores y supercomputadoras, con jerarquías de memoria de varios niveles y en ambientes de procesamiento distribuido. Esta evolución natural requirió la adición de nuevas características al núcleo y esto ha provocado el incremento en su tamaño y complejidad, haciendo que desaparezcan rasgos importantes como la simplicidad en el diseño y la estructura del núcleo.

La tecnología de *micronúcleo* (microkernel) es una propuesta de estructuración de sistemas operativos que consiste en remover los componentes que no sean esenciales para el núcleo e implantarlos como servidores a nivel de usuario [Golub 1990, Silberschatz et al. 2002]. El resultado es un núcleo más compacto que permite y fomenta un diseño modular. No hay mucho consenso en el tipo de servicios primitivos que debe proporcionar un micronúcleo, sin embargo, podemos delimitar y enu-

merar de forma precisa las primitivas proporcionadas por micronúcleos actuales:

- gestión de procesos
- administración de memoria
- mecanismos de comunicación entre procesos.

Algunos autores afirman que un micronúcleo debe proveer también entrada y salida de bajo nivel. La función principal del micronúcleo es proporcionar un medio de comunicación entre la aplicación cliente y otros programas servidores también en ejecución en el espacio del usuario. Esta comunicación se realiza mediante paso de mensajes. Por ejemplo, en un sistema operativo basado en micronúcleo si un proceso necesita hacer uso de un archivo, debe interactuar con el servidor del sistema de archivos. La aplicación cliente y el servidor nunca realizan interacción directa, se comunican de forma indirecta a través del intercambio de mensajes con el micronúcleo.

Los beneficios de este enfoque son los siguientes:

Modularidad: Existe una interfaz muy bien definida con cada servicio (el conjunto de mensajes que comprende el servidor), los cuales son igualmente accesibles a los clientes, independientemente de su posición.

Facilidad de extensión: Algún nuevo servicio necesario es añadido en el espacio del usuario y el núcleo, en consecuencia, no requiere modificaciones. Es fácil implantar, instalar y depurar nuevos componentes pues la modificación o adición de un servicio no requiere detener todo el sistema y sus aplicaciones. Es necesario mencionar que la existencia del micronúcleo es transparente al usuario.

Portabilidad: Mantener un núcleo pequeño implica un menor esfuerzo de compilación del núcleo en una plataforma nueva. Además permite identificar y localizar errores de manera más rápida. Las aplicaciones y servidores que hacen uso de los servicios primitivos no necesitan ser modificadas.

Seguridad: Los procesos servidores o de aplicación que se ejecutan fuera del núcleo no representan un riesgo a la estabilidad y funcionamiento del sistema.

Confiabilidad: Si algún servidor o programa de aplicación falla, el resto del sistema operativo continúa trabajando de forma normal.

El punto de controversia sobre la efectividad de los sistemas basados en micronúcleo se encuentra en el rendimiento. Se afirma que el empleo de paso de mensajes como mecanismo único de comunicación es un factor en la disminución del rendimiento. Sin embargo, se

discutirá más adelante algunas formas eficientes de implantación del esquema de comunicación mediante envío de mensajes.

Los micronúcleos son de particular interés debido a que proporcionan un marco de trabajo por demás adecuado para crear *ambientes de emulación* [Golub 1990]. En tales ambientes existe un único micronúcleo asociado al hardware de un sistema de cómputo donde, en un nivel superior, se pueden construir componentes que proporcionen los servicios propios a algún sistema operativo específico. A este escenario es posible agregar los componentes que emulan los servicios de algún otro sistema operativo, y entonces los componentes de un tercer sistema y así. Como resultado tenemos un ambiente donde conviven más de un sistema operativo en una misma computadora, donde cada uno tiene asociados programas de aplicación muy particulares pero todos emplean los servicios primitivos de un mismo y único micronúcleo. La figura 1 muestra una computadora basada en la arquitectura x86 (que puede tener desde uno hasta miles de procesadores) que ejecuta un micronúcleo, que contiene sistemas de emulación para varios sistemas operativos y que ejecuta varios procesos de usuario de forma concurrente.

El resto de este documento se encuentra estructurado de la siguiente forma. En la sección 2 se ilustran algunas arquitecturas de computadoras y los sistemas operativos, basados en micronúcleo, que ejecutan. La sección 3 discute los detalles de un micronúcleo ampliamente desarrollado llamado Mach, en esta sección se exponen algunas de las características que permiten implantar sistemas distribuidos eficientemente. Finalmente se presentan las conclusiones en la sección 4.

2 Sistemas comerciales basados en micronúcleo

El dominio de aplicación de los principios de la tecnología de micronúcleos en el desarrollo de sistemas operativos comerciales es muy amplio. Los líderes de proyectos de desarrollo e implantación de sistemas operativos en compañías tan importantes como Cray, Apple, HP/Compaq, Microsoft, etcetera, se han convencido de las ventajas de implantar los sistemas operativos comerciales mediante el empleo de un micronúcleo como base.

La figura 2 muestra cuatro tipos de computadoras comerciales cuyos sistemas operativos, que también se mencionan, están diseñados mediante la tecnología en discusión. Los sistemas operativos considerados son los siguientes:

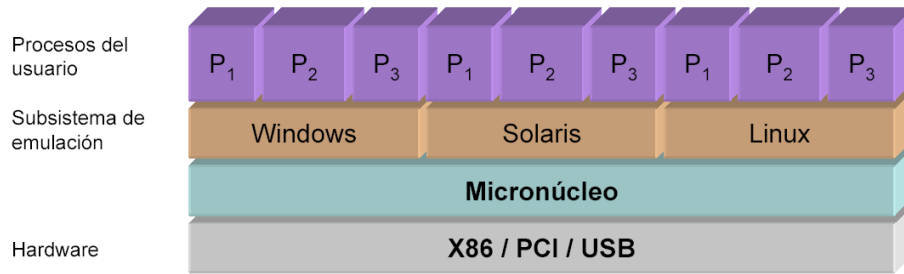


Figura 1: Ambiente de emulación de varios sistemas operativos mediante un micronúcleo.

2.1 NeXTSTEP

Un ambiente operativo desarrollado en la hoy desaparecida compañía NeXT Computers Corp [NeXT 1992]. NeXTSTEP es una combinación y balance perfecto entre poder y desempeño. NeXTSTEP es un sistema operativo compatible con UNIX BSD 4.3 y fue diseñado e implantado utilizando el micronúcleo Mach. NeXTSTEP también es un ambiente de operación que fue concebido con un enfoque completamente orientado a objetos, lo cual se nota tanto en la interfaz gráfica como en el proceso sencillo de desarrollo de programas de aplicación. La figura 2–(c) ilustra una computadora NeXTStation, la cual estaba desarrollada en base a procesadores Motorola de la familia 68000 y otros dispositivos poco comunes para su época como procesadores de señales digitales (DSPs) y lectores de discos ópticos [Ibarra y Vergara 1995].

2.2 MacOS X

La última versión de este popular sistema operativo significa una revolución de la compañía en el campo de los sistemas operativos. MacOS X hereda algunas características de NeXTSTEP, como el hecho de estar construido a partir de Mach. MacOS X se compone de un sistema operativo compatible con UNIX, y diseñado sobre Mach, llamado Darwin. En niveles más altos encontramos la novedosa interfaz gráfica Aqua junto con los programas de aplicación y las herramientas de desarrollo apropiadas. La figura 2–(b) ilustra un modelo reciente de las novedosas estaciones de trabajo iMac que ejecutan el ambiente MacOS X. Estas computadoras contienen microprocesadores RISC PowerPC y se caracterizan en nuestros días por su gran flexibilidad y diseños atractivos.

2.3 UNICOS/mk

Un sistema operativo con una propuesta de diseño distinta a la alternativa de implantación de los sistemas operativos monolíticos previos de la compañía Cray Inc [Broner 1996]. UNICOS/mk también se compone de

un micronúcleo y varios servidores, y se deriva de Chorus y UNICOS. La parte de Chorus proporciona el micronúcleo, el esquema de implantación de servidores y las bases para la gestión de procesos. Los servidores fueron desarrollados a partir del código de UNICOS. La supercomputadora Cray T3E es una arquitectura basada en microprocesadores RISC Alpha y contrasta un poco con otros diseños de la empresa construidos a partir de procesadores vectoriales propietarios.

2.4 Tru64

Tru64 es otro sistema operativo compatible con UNIX que se encuentra basado en Mach. La computadora AlphaServer SC45 es un cluster de computadoras con microprocesadores Alpha que puede escalar hasta varios cientos de nodos.

3 Mach

El proyecto Mach se llevó a cabo en la Universidad Carnegie Mellon (CMU) durante la década de los 80s como una propuesta de solución al fenómeno de la creciente complejidad de los sistemas operativos monolíticos, como UNIX. El resultado es un sistema operativo compatible con la distribución UNIX BSD 4.3 diseñado para ser ejecutado en máquinas con múltiples procesadores de memoria compartida o distribuida. Su diseño modular permite portarlo de forma más simple y así funcionar en plataformas heterogéneas. En la actualidad este sistema brinda un ambiente excelente para desarrollar sistemas distribuidos.

3.1 Principios de diseño

La siguiente es una lista de las características con las que se pensó que Mach debería contar y los servicios que debería proveer:

- Mach fue diseñado para funcionar en diferentes arquitecturas, incluyendo sistemas multiprocesadores con varios grados de acceso a memoria compartida. Por ejemplo, computadoras de tipo NUMA (acceso



(a) Tru64-Alpha Server SC45



(b) MacOS X-iMac G4



(c) NeXTSTEP-NeXTStation



(d) UNICOS/mk-T3E

Figura 2: Computadoras que utilizan sistemas operativos basados en tecnología micrónúcleo.

no uniforme a memoria), UMA (acceso uniforme a memoria), etc.

- Mach fomenta y explota el empleo de paralelismo tanto a nivel de software de sistema como de programas de aplicación. La idea es explorar los multiprocesadores y los sistemas distribuidos, y a la vez emular los sistemas operativos ya existentes.
- Mach está diseñado como un sistema heterogéneo que permite tenerlo ampliamente disponible y en operación entre sistemas computacionales de diversos fabricantes.
- Mach posee un núcleo con una estructura simplificada que proporciona un pequeño número de abstracciones, las cuales deben ser lo bastante generales para permitir la construcción de sistemas operativos completos sobre Mach.
- Mach cuenta con mecanismos de gestión de memoria y de intercomunicación entre procesos integrados, de tal forma que es posible implantar esquemas eficientes de comunicación en base a paso de mensajes mediante la manipulación del espacio de direcciones asignado a los procesos. Por ejemplo la implantación de paso de mensajes mediante la técnica *copy on write* para copiar regiones de memoria entre procesos.

Desde el punto de vista de los sistemas distribuidos, que es el área que nos interesa, Mach fue diseñado tomando en cuenta los siguientes aspectos:

- Habilidad de funcionar con velocidades variables en las redes de interconexión de computadoras. Desde redes de área amplia, redes de área local de alta velocidad y multiprocesadores fuertemente acoplados.
- Operación distribuida que proporciona a los clientes un acceso transparente a los recursos en la red y una organización “orientada a objetos” tanto externa como interna debido al mecanismo de comunicación de paso de mensajes.

3.2 Abstracciones

Hasta este momento se ha hecho un tremendo énfasis en el número reducido de servicios proporcionados por los componentes (o *abstracciones*) contenidos en el micrónúcleo. Ahora es tiempo de describir las abstracciones que forman parte del micrónúcleo (Mach en particular) y la forma en que se combinan para proporcionar componentes y servicios más complejos. Mach ofrece las siguientes abstracciones:

Tarea: Es un ambiente de ejecución compuesto de un espacio de direcciones virtuales. Es la unidad básica de asignación de los recursos del sistema, cuyo acceso se encuentra protegido mediante puertos. La

tarea es el entorno en el que se ejecutan uno o más hilos.

Hilo: Es la unidad básica de ejecución y se debe ejecutar en el contexto de una tarea que le proporcione un espacio de direcciones. Todos los hilos en una tarea comparten los recursos de la misma, i.e. puertos, memoria, etc. El hilo está compuesto por un identificador, un contador de programa, una configuración para los registros del CPU y una pila.

Puerto: Es el mecanismo básico de referencia a objetos. Está implantado como un canal de comunicación protegido por el núcleo. Las operaciones a realizar sobre un objeto se invocan enviando un mensaje al puerto asociado y el objeto, representado por tal puerto, los recibe. Los puertos se encuentran protegidos mediante *derechos a puerto* que adquieren las tareas,

Mensaje: Es el método básico de comunicación entre hilos. Se compone de un encabezado de longitud fija, un cuerpo de longitud variable que, a su vez, está compuesto de cero o más objetos con tipo. Un mensaje puede transmitir datos o apuntadores a los mismos. Las tareas pueden compartir derechos a puerto enviándolos a través de mensajes.

Objeto de memoria: Es una fuente de memoria. Son administrados por gestores de memoria externos al núcleo. En general es cualquier objeto que tenga sentido mapear a memoria.

Podemos concebir, y de hecho implantar, la noción de proceso en sistemas operativos tradicionales, e.g. UNIX, a través de una tarea y un hilo de ejecución. De la misma forma es posible implantar medios de intercomunicación entre procesos mediante mensajes, puertos, tareas e hilos. La figura 3 muestra un esquema de la localización de las abstracciones del micronúcleo asociadas a un proceso en ejecución en un sistema operativo basado en micronúcleo.

3.3 Comunicación entre procesos

En general, en sistemas tradicionales es difícil proporcionar la *transparencia de la posición de los objetos* en el momento de desarrollar aplicaciones distribuidas. Mach tiene ventajas en este sentido puesto que permite el envío de mensajes a puertos, esta acción de paso de mensajes se realiza sin tener conocimiento de la localización del objeto asociado al puerto. Además, hay que recalcar la importancia de vista que ofrece Mach al desarrollador; la de objetos que conviven en el sistema a los que se les pueden solicitar servicios o invocar alguna operación sobre ellos mediante envío de mensajes. La finalidad es proporcionar un ambiente de desarrollo “orientado a objetos”. Los mensajes son interpretados de forma simple pues todos tienen un formato bien establecido.

La seguridad en la comunicación es un detalle importante a considerar, afortunadamente Mach proporciona esquemas de derechos a puerto que impiden a tareas invasoras invocar operaciones nocivas. Los derechos a puerto se componen de un identificador al puerto al que están asociados y de una lista de *posibilidades* (capabilities) que el derecho otorga sobre el puerto, estas posibilidades pueden ser de envío (SEND) o de recepción (RECEIVE). De esta forma es necesario contar con derechos para enviar un mensaje a un puerto o recibir un mensaje desde el mismo. Únicamente una tarea a la vez puede contar con derechos de recepción sobre un puerto, pero varias otras tareas pueden tener derechos de envío hacia ese mismo puerto.

3.4 Comunicación en ambientes distribuidos

Ahora es tiempo de considerar la forma en que el mecanismo de intercomunicación entre procesos, implantado en base a las abstracciones de Mach, nos permite establecer comunicaciones distribuidas independientes de la localización de los objetos. Esta característica es de gran importancia y tiene múltiples aplicaciones en el campo del desarrollo de sistemas distribuidos [Coulouris et al. 2001, Bellardo et al. 1999].

El componente clave que permite realizar el envío de mensajes entre computadoras es un servidor llamado NetMsgServer, que está replicado en cada computadora y se emplea cuando es necesario transmitir un mensaje a un puerto que no se encuentra en el núcleo de una computadora sino en el de otra. Este componente proporciona un servicio de nombres que permite que permite a las tareas registrar puertos que son empleados por otras tareas en cualquier nodo de la red, este servicio de nombres permite solucionar el problema de enviar y registrar el primer puerto que permite el intercambio de mensajes entre todas las tareas. Los servidores NetMsgServer mantienen una base de datos distribuida que contiene derechos a puertos que han sido intercambiados entre computadoras y los puertos a los que corresponden tales derechos.

Los siguientes son los pasos a seguir para enviar mensajes entre diferentes nodos de la red:

1. Emplear comunicación local para transmitir un mensaje al servidor NetMsgServer local
2. El servidor emplea un protocolo de red apropiado y transmite el mensaje a su contraparte en el nodo destino
3. El servidor local en el nodo destino envía el mensaje a la tarea destino correcta

Los servidores NetMsgServer emplean *puertos proxy* que representan los puertos de otros objetos distribuidos a lo largo de la red. Estos puertos son creados cuando se ha

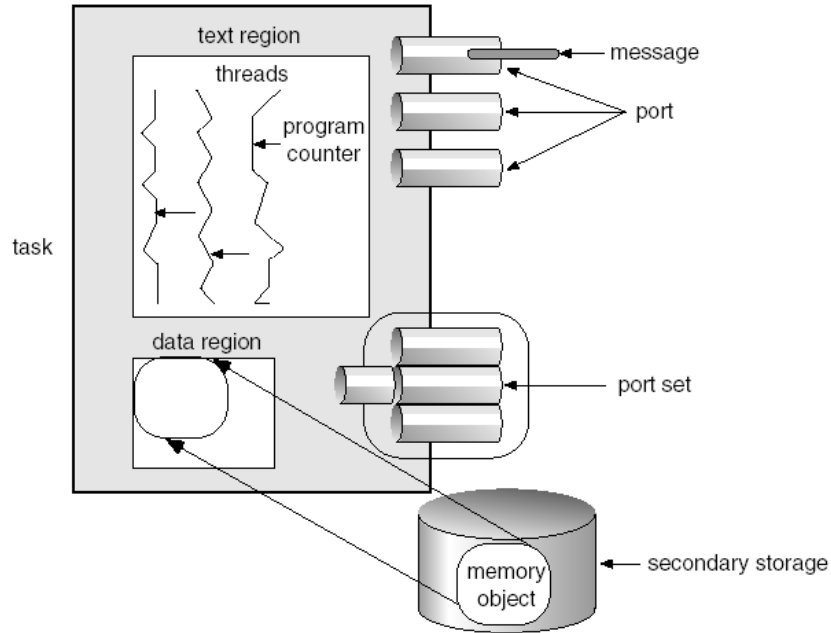


Figura 3: Las abstracciones proporcionadas por el micrókernel Mach.

transmitido un derecho de envío. Los servidores NetMsgServer deben coincidir en el protocolo empleado para la comunicación y emplean la información contenida en el mismo mensaje para determinar la extensión de un tipo de datos y el ordenamiento de sus bytes *endianess*. Como Mach está diseñado para funcionar en ambientes heterogéneos el hecho de contar con plataformas diversas a lo largo de la red no afecta el desempeño mientras haya un acuerdo en el protocolo de comunicación.

3.5 Escalabilidad

Como varios servicios pertenecientes al sistema operativo, y fuera de él, están implantados como programas a nivel de usuario es posible configurar cada nodo de un sistema distribuido con diferentes servicios. Esta situación es muy ventajosa pues nos permite eliminar servicios que nunca sean empleados en cada nodo, lo que incrementa el rendimiento y permite ahorrar recursos de cómputo. Esta configuración personalizada en cada nodo se puede llevar a cabo sin necesidad de recompilar el núcleo del sistema operativo. Además, si un servidor falla el sistema completo no se detiene en consecuencia.

4 Conclusiones

Un micrókernel proporciona las abstracciones básicas que permiten implantar las operaciones más elaboradas en un sistema operativo y han demostrado ser de gran utilidad tanto a nivel académico como a nivel comercial.

Mach es un micrókernel que está respaldado por

grandes esfuerzos de investigación y desarrollo y ha sido ampliamente utilizado por varias compañías como base de sus sistemas operativos.

Los desarrolladores de sistemas distribuidos pueden aprovechar las funcionalidades proporcionadas por los micrókernels para implantar algoritmos distribuidos en forma eficiente. Es posible extender las funcionalidades del micrókernel desarrollando servidores que permitan la gestión e transacciones y control de concurrencia, la sincronización, el ordenamiento de eventos, middleware, etc. En términos generales, es una tecnología que permite evaluar nuevas propuestas algorítmicas y de diseño.

Referencias

- [Bellardo et al. 1999] Bellardo, J., M. Copenhafer y G. Hamerly. 1999. *Microkernels as Foundations for Distributed Systems*. Documento de clase.
- [Broner 1996] Broner, G. 1996. *UNICOS/mk: A Scalable Distributed Operating System*. En *Proceedings of Thirty-Seventh Semi-Annual Cray User Group Meeting*.
- [Coulouris et al. 2001] Coulouris, G., J. Dollimore y T. Kindberg. 2001. *Distributed Systems Concepts and Design*. Tercera edición. Massachusetts: Addison-Wesley.
- [Golub 1990] Golub, D., R. Dean, A. Forin y R. Rashid. 1990. *UNIX as an Application Program*. En *Proceeding of the USENIX Summer Conference*.

- [Ibarra y Vergara 1995] Ibarra, E. y G. Vergara. 1995. Evaluación del sistema operativo NeXTSTEP en el Grupo Financiero InverMéxico. *Soluciones Avanzadas* 3(18):5–11.
- [NeXT 1992] NeXT Computer, Inc. 1992. *NeXTSTEP Operating System Software*. Massachusetts: Addison-Wesley.
- [Silberschatz et al. 2002] Silberschatz, A., P. B. Galvin y G. Gagne. 2002. *Operating System Concepts*. Sexta edición. New York: John Wiley & Sons, Inc.