Engineer

Núcleo

Parte central de un sistema operativo

Qué es

El núcleo es el programa situado en el centro de un sistema operativo, que se ejecuta con todos los privilegios del procesador y controla el acceso al hardware. Decide qué programa se ejecuta después, reparte memoria y separa la de cada programa de la de los demás, lee y escribe discos mediante sistemas de archivos, mueve paquetes por la pila de red y hace cumplir las reglas que determinan quién puede acceder a cada archivo, puerto o dispositivo. Todo lo demás se ejecuta en el espacio de usuario con menos privilegios y pide al núcleo que actúe en su nombre.

Esas peticiones son llamadas al sistema, que el manual de Linux denomina «la interfaz fundamental entre una aplicación y el núcleo de Linux». Los programas rara vez las realizan a mano: llaman a funciones envoltorio de una biblioteca C, como glibc, que transmiten los argumentos y convierten la respuesta del núcleo en un valor de retorno y un código de error. Cuando un programa se detiene con «Permission denied», ese texto suele ser la respuesta del núcleo a una de esas llamadas, traducida por la biblioteca.

Los núcleos difieren en cuánto conservan dentro. Linux mantiene los controladores de dispositivos, los sistemas de archivos y la pila de red en el propio núcleo, y puede cargar más como módulos mientras se ejecuta, un diseño que suele llamarse monolítico. Cada controlador se ejecuta entonces con todos los privilegios del núcleo. Un micronúcleo conserva solo lo mínimo. seL4 ofrece mecanismos para controlar el acceso al espacio de direcciones físicas, las interrupciones y el tiempo de procesador, y ejecuta controladores, sistemas de archivos y pilas de red como programas corrientes en modo usuario; su versión verificada para RISC-V de 64 bits tiene unas 10.000 líneas de código, con una prueba formal de que la implementación C hace lo que dice su especificación. El núcleo de macOS, XNU, es un híbrido que combina el micronúcleo Mach con partes de FreeBSD.

Una actualización del núcleo normalmente solo surte efecto cuando la máquina arranca con ella, y el número de versión del núcleo, como 6.12, es independiente del número de versión de la distribución que lo rodea.

Utilice el concepto en una decisión

Aplique «¿Qué es un núcleo?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. El cuidado diario de servidores, correo, cuentas y actualizaciones para que los servicios de los que depende todo el mundo sigan funcionando sin sorpresas.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 10 servidores PostgreSQL y MS SQL en Ubuntu — 10 servidores de bases de datos desplegados y mantenidos

10 servidores PostgreSQL y MS SQL en Ubuntu Póngase en contacto →

Por qué importa

Para una empresa, el núcleo resuelve tres preguntas: cuánto se aíslan realmente sus cargas de trabajo, cuánto tiempo de inactividad cuesta una corrección de seguridad y dónde reside la verdad sobre una máquina.

Primero, el aislamiento. Un contenedor es un conjunto de procesos corrientes que el núcleo ha situado en espacios de nombres, que dan a cada grupo su propia vista de identificadores de proceso, interfaces de red, montajes y usuarios, y en grupos de control, que reparten tiempo de procesador, memoria y entrada y salida de disco. Todos los contenedores de un host se ejecutan sobre su único núcleo, por lo que un fallo en él afecta a todos a la vez, y una imagen que lleva sus propias bibliotecas sigue dependiendo de las funciones del núcleo del host. Una máquina virtual arranca con un núcleo propio. Para los servicios de un equipo, los contenedores son una frontera razonable; para clientes que no confían entre sí, dé a cada organización cliente su propio núcleo.

Después, las correcciones. Un paquete de núcleo parcheado en disco no cambia nada hasta que la máquina arranca con él, y en un servidor ese reinicio es el coste. Los parches en caliente existen porque, en palabras de la documentación del núcleo, los usuarios son «reacios a reiniciar»: redirigen funciones concretas a versiones corregidas mientras el sistema se ejecuta. Solo se pueden parchear así las funciones que el núcleo puede trazar, por lo que los parches en caliente dan tiempo ante una corrección urgente en lugar de sustituir los reinicios. Un plan con reinicios periódicos y ensayados cuesta menos que uno que supone que nunca harán falta.

Por último, el núcleo contiene el estado real en ejecución. Un archivo de cortafuegos describe reglas; las que están en el filtro de paquetes del núcleo deciden el destino de un paquete. Un archivo de `/etc/sysctl.d` describe un ajuste; el valor de `/proc/sys` es el que está vigente. Un servicio puede declararse sano haya llegado o no su acción al núcleo. Las comprobaciones que leen el estado del propio núcleo detectan fallos que una revisión de los archivos de configuración daría por buenos.

Cómo se aplica en la práctica

Una revisión de seguridad de un host de producción en agosto de 2026 preguntó si funcionaban sus protecciones frente a la fuerza bruta en SSH. Todas estaban configuradas y aparecían en todos los informes, por lo que la revisión las contrastó con lo que el núcleo hace realmente con un paquete, en lugar de con los archivos que describían lo que debía ocurrir. Desde la perspectiva del núcleo, ninguna de las tres había actuado ni una vez desde la creación del host. Los bloqueos del jail indicaban el puerto 22, donde ya no escuchaba ningún socket después de trasladar SSH al puerto 1986. El límite de frecuencia de paquetes sí llegaba al filtro de paquetes, que evalúa las reglas de arriba abajo y se detiene en la primera coincidencia; una regla más amplia, situada por encima, capturaba todos los paquetes antes de consultar el límite. La acción de bloqueo ejecutaba un programa de filtrado de paquetes que su paquete declara como dependencia recomendada en lugar de obligatoria; en un host sin él, el jail arrancaba, contaba y decidía, y después fallaba justo cuando debía entregar una regla al núcleo. La revisión produjo dos reglas: un control de seguridad se demuestra mediante una aserción, algo que un comentario no puede hacer, y un cortafuegos se comprueba por el orden en que el núcleo mantiene sus reglas, no por la mera existencia de una regla.

Después se añadió al proceso de bloqueo una prueba que se ejecuta cada vez que se aplica la configuración. En cada ejecución bloquea una dirección de un bloque reservado para documentación, vuelve a leer la regla resultante en el filtro de paquetes del núcleo y retira el bloqueo en un paso de limpieza que se ejecuta tanto si hay éxito como si hay fallo. Esa lectura es lo que la convierte en una prueba: el estado del propio servicio de bloqueo indicaría éxito ante un bloqueo que no produjera ninguna regla. Una dirección reservada no pertenece a ninguna red real, por lo que un bloqueo que quedara tras una ejecución fallida no dejaría fuera a nadie. El coste es de unos segundos por ejecución y una regla de cortafuegos que se escribe en producción y se retira cada vez.

Un host de 464 MB sin espacio para una plataforma de métricas recibió siete roles de solo lectura que representan la máquina en ejecución como Markdown; su instantánea de datos calcula la utilización del procesador a partir de dos muestras del propio contador del núcleo. El informe de seguridad imprime las reglas del cortafuegos en su orden de coincidencia, junto a las reglas de bloqueo activas, y cada uno de los siete roles imprime solo cifras leídas en su propia ejecución, incluidos los contadores del núcleo, nunca una estimación para rellenar un hueco. Esos informes encontraron dos controles de seguridad inactivos y un puerto a la escucha que nadie había declarado.

Dónde surgen los problemas

Un ajuste del núcleo modificado a mano vuelve a su valor anterior tras el siguiente reinicio. `sysctl -w` solo escribe el valor en el núcleo en ejecución, por lo que dura hasta reiniciar. El descuido contrario: un archivo de `/etc/sysctl.d` se lee al arrancar, así que el cambio espera, sin avisar, a un reinicio posterior. Escriba el ajuste en un archivo, cárguelo ahora y vuelva a leer el valor activo:

```
echo 'vm.swappiness = 10' | sudo tee /etc/sysctl.d/90-local.conf
sudo sysctl --system
sysctl vm.swappiness
```

Un gráfico de CPU permanece plano durante un pico de carga. La línea `cpu` de `/proc/stat` cuenta el tiempo desde el arranque en unidades USER_HZ —centésimas de segundo en la mayoría de las arquitecturas—, por lo que una única lectura da la media desde el arranque, que apenas se mueve en una máquina que lleva meses encendida. Léala dos veces y divida las diferencias:

```
head -1 /proc/stat; sleep 5; head -1 /proc/stat
```

El tiempo ocupado es la variación de user, nice, system, irq y softirq; el tiempo inactivo es la variación de idle e iowait; y la utilización es el tiempo ocupado dividido entre la suma del ocupado y el inactivo. El manual considera iowait poco fiable, así que trátelo como indicio, no como alerta.

Un informe de fallo recibe como respuesta una petición de reproducirlo en un núcleo no contaminado. El núcleo se marca como contaminado cuando ocurre algo que puede importar en una investigación posterior, como cargar un módulo

propietario, externo al árbol del núcleo o sin firma, y su documentación advierte de que los desarrolladores suelen ignorar los informes de errores de núcleos contaminados. La marca permanece hasta reiniciar, incluso después de retirar el módulo de la memoria. Léala antes de enviar el informe:

```
cat /proc/sys/kernel/tainted
```

Cero significa no contaminado; cualquier otro número es una suma de indicadores, entre ellos `1` para un módulo propietario, `4096` para uno externo al árbol y `8192` para uno sin firma. Reproduzca el fallo después de arrancar sin el módulo que activó el indicador.

Una tarjeta de red nueva aparece en la lista de hardware y el sistema no ofrece ninguna interfaz para ella. Los controladores forman parte del núcleo, y uno anterior al hardware puede carecer del necesario. `lspci -k` muestra el controlador que usa cada dispositivo y los módulos capaces de gestionarlo; si un dispositivo no tiene una línea de controlador en uso, no tiene ninguno asociado. Instale una serie más reciente del núcleo de la distribución en lugar de compilar a mano el módulo del fabricante: un módulo externo al árbol debe recompilarse para cada núcleo nuevo, y cargarlo contamina el núcleo.

Lecturas adicionales

- Guía del usuario y del administrador del núcleo Linux — el manual del propio proyecto del núcleo para ejecutar Linux.
- syscalls(2) — todas las llamadas al sistema de Linux, con la versión del núcleo que introdujo cada una.
- namespaces(7) — los tipos de espacios de nombres y cómo se construyen los contenedores a partir de ellos.
- proc_stat(5) — todos los campos de `/proc/stat`, incluida la advertencia sobre iowait.
- Núcleos contaminados — cada indicador de contaminación y cómo decodificar el número.
- Preguntas frecuentes de seL4 — qué deja fuera un micronúcleo y los supuestos de la prueba de seL4.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es un núcleo?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/kernel/