

Los contenedores como **LXC** (Linux Containers) aprovechan funcionalidades del kernel Linux como **namespaces** y **cgroups** para aislar procesos y recursos sin virtualizar hardware completo como una máquina virtual tradicional. Pero esto supone un problema: muchos programas dentro de un contenedor leen información del sistema desde /proc y esperan valores que reflejen solo *lo que el contenedor puede usar*, no el total del host. Por ejemplo:

- /proc/meminfo reporta memoria total.
- /proc/uptime indica cuánto tiempo lleva activo el sistema.
- /proc/cpuinfo muestra CPUs disponibles.

Sin aislamiento de estos valores, herramientas como top, uptime o incluso systemd no funcionan correctamente dentro de contenedores porque ven el sistema completo, no solo el contenedor.

LXCFS (Linux Containers Filesystem) es un **filesystem en el espacio de usuario basado en FUSE** diseñado para “corregir” esta visión de /proc y los cgroups para contenedores.

Objetivo principal: que dentro de un contenedor los valores de ciertos archivos reflejen lo que esa instancia realmente puede usar, no lo que hay en el host.

Técnicamente, LXCFS:

- Proporciona una vista “virtualizada” de archivos clave bajo /proc tales como /proc/cpuinfo, /proc/meminfo, /proc/stat, /proc/uptime, etc.
- Ofrece un árbol estilo cgroupfs que es consciente de contenedor (container-aware).
- Usa FUSE para interceptar y presentar estos datos de forma dinámica según los límites del contenedor.

Todo esto está implementado en C, ofrece una biblioteca compartida liblxcfs y un binario lxcfs que maneja la lógica de FUSE para generar los contenidos virtualizados.

En lugar de que los procesos del contenedor lean directamente del /proc del host, se montan versiones “procesadas” de esos archivos creadas por LXCFS. Así, al leer /proc/meminfo dentro del contenedor, se ve solo la memoria asignada por cgroups y no toda la del host.

Resumen de componentes:

- **FUSE filesystem**: intercepta lecturas de /proc y genera contenido ajustado.
- **Bind-mounts**: los valores generados por LXCFS se montan sobre los originales de /proc dentro del contenedor.
- **CGroup awareness**: LXCFS sabe qué recursos están limitados a ese contenedor y genera los datos correctos con base en esa información.

FUSE (Filesystem in Userspace): LXCFS usa FUSE para crear un sistema de archivos en espacio de usuario que puede responder a lecturas de forma programática.

CGroup awareness: LXCFS ajusta valores usando información de cgroups (cpu, memoria, etc.). Esto permite reportar solo los recursos disponibles para ese contenedor.

Soporte para múltiples archivos /proc: LXCFS virtualiza varios archivos sensibles a recursos, entre ellos:

- /proc/cpuinfo
- /proc/meminfo
- /proc/stat
- /proc/uptime
- /proc/swaps

- /proc/pressure/*
- /sys/devices/system/cpu/online

Todos los valores se ajustan a lo que el contenedor puede ver y usar, no al total del host.

LXCFS nació como complemento de **LXC**, pero puede usarse con cualquier runtime que necesite mejorar vistas de /proc para contenedores.

En distribuciones como Ubuntu, LXCFS se instala y activa automáticamente para que los contenedores tengan vistas correctas de recursos desde el inicio.

PROBLEMAS REALES QUE SOLUCIONA

Sin LXCFS, dentro de un contenedor típico:

- uptime podría mostrar que el sistema lleva días activo, aunque el contenedor se creó hace minutos.
- top o htop mostrarían más CPUs o más memoria de la que el contenedor realmente tiene asignada.
- systemd podría no funcionar porque espera una vista consistente de cgroups y recursos.
- cat /proc/meminfo mostraría sólo la memoria asignada por cgroups (como systemd) dentro de contenedores sin confundirlos con el host.

Esto funciona porque LXCFS reemplaza esos puntos de montaje con su versión filtrada usando FUSE y bind-mounts.

CÓDIGO ABIERTO

LXCFS es de código abierto, bajo licencia Apache-2, mantenido en [GitHub](#).

Incluye:

- El servidor FUSE que genera los sistemas de archivos virtuales.
- La librería liblxcfs para integraciones más directas.
- Soporte continuo para nuevas métricas de cgroups y archivos /proc.

