

LPIC-2 / Examen 203 - Sistema de Archivos y Dispositivos

203.2 Mantener un sistema de archivos de Linux

Teoría

Los sistemas de archivos, a pesar de ser robustos, pueden sufrir corrupción debido a fallos de hardware (disco defectuoso), cortes de energía inesperados, errores en el kernel o problemas de software. Mantener la integridad de los sistemas de archivos es crucial para prevenir la pérdida de datos y asegurar el funcionamiento estable del sistema.

Importancia del Journaling:

- Los sistemas de archivos con journal (como ext3, ext4, XFS, Btrfs) mejoran significativamente la capacidad de recuperación después de un fallo (crash) inesperado en comparación con sistemas sin journal (como ext2 o FAT).
- Un journal (o diario) es un área especial en el disco donde el sistema de archivos registra los cambios planificados en los metadatos (información sobre archivos, directorios, permisos, etc.) antes de aplicarlos al sistema de archivos principal.
- Después de un fallo, el sistema puede "reproducir" el journal para completar las operaciones pendientes o revertir las incompletas, llevando el sistema de archivos a un estado consistente mucho más rápido y con menos riesgo de corrupción que una comprobación completa.

Herramientas para Comprobar y Reparar Sistemas de Archivos:

La utilidad principal es `fsck`, que es un front-end para herramientas específicas de cada tipo de sistema de archivos.

1. **fsck (filesystem check):**

- Es un envoltorio (`wrapper`) que llama a la utilidad de comprobación apropiada (ej: `e2fsck`, `fsck.xfs`) basándose en el tipo de sistema de archivos detectado en el dispositivo.
- **Uso Básico:** `sudo fsck <dispositivo>` (ej: `sudo fsck /dev/sda1`).
- **¡REGLA DE ORO! `fsck` DEBE ejecutarse idealmente en un sistema de archivos DESMONTADO.** Ejecutar `fsck` en un sistema de archivos montado, especialmente si se realizan escrituras, puede causar más corrupción. La única excepción es el sistema de archivos raíz (`/`), que `fsck` comprueba durante el arranque *antes* de montarlo de lectura/escritura.
- Opciones comunes que `fsck` pasa a la utilidad subyacente:
 - `-f`: Fuerza la comprobación incluso si el sistema de archivos parece limpio.
 - `-y`: Responde "sí" automáticamente a todas las preguntas de reparación.
 - `-n`: Responde "no" automáticamente a todas las preguntas de reparación (modo seguro, solo muestra lo que haría).

2. e2fsck:

- La herramienta específica para sistemas de archivos ext2, ext3 y ext4. Es el `fsck` subyacente para estos tipos.
- `sudo e2fsck [-f] [-y|-n] <dispositivo>` (ej: `sudo e2fsck /dev/sdb1`).
- A menudo se ejecuta automáticamente durante el arranque si una comprobación periódica (`tune2fs -c o -i`) indica que es necesaria, o si el sistema no se desmontó limpiamente.
- **Nota:** El paquete que lo provee es `e2fsprogs` en ambas ramas (Debian/Red Hat).

3. xfs_repair:

- La herramienta específica para sistemas de archivos XFS. El enfoque de XFS en la recuperación de fallos es diferente; confía más en su journal y está diseñado para recuperarse rápidamente al montar después de un fallo. `xfs_repair` se usa para corrupción más grave que el journal no puede manejar.
- `sudo xfs_repair <dispositivo>` (ej: `sudo xfs_repair /dev/sdc1`).
- **¡También DEBE ejecutarse en un sistema de archivos DESMONTADO!**
- **No tiene modo interactivo.** Si detecta corrupción que no puede reparar de forma segura, puede detenerse o requerir opciones avanzadas (como `-L` para limpiar el log, ¡lo que puede causar pérdida de datos!).
- **Nota:** El paquete que lo provee es `xfsprogs` en ambas ramas. También existen `xfs_info` para ver detalles y `xfs_check` (menos usado que `xfs_repair`).

4. btrfs check:

- La herramienta para sistemas de archivos Btrfs. Es una herramienta en evolución; la capacidad de reparación (`btrfs check --repair`) aún puede ser experimental y riesgosa en algunos casos.
- **Nota:** El paquete es `btrfs-progs` en ambas ramas.

Comprobaciones de Sistema de Archivos Durante el Arranque:

- El campo `<pass>` en `/etc/fstab` (ver 203.1) controla el orden de las comprobaciones `fsck` durante el arranque:
 - `0`: No comprobar este sistema de archivos.
 - `1`: Comprobar primero (reservado para el sistema de archivos raíz `/`).
 - `2`: Comprobar después (para otros sistemas de archivos).
- El script de arranque o el sistema de inicio (`systemd`) ejecuta `fsck` en los dispositivos con `pass` mayor que 0 antes de montarlos. El sistema de archivos raíz (con `pass=1`) se comprueba primero por el `initramfs` o un script temprano.

- Si la comprobación de `fscck` en el arranque encuentra errores que no puede reparar automáticamente, el sistema puede iniciar en modo monousuario (`single user mode`) para permitir al administrador intervenir.

Forzar Comprobación de `fscck` en el Próximo Arranque:

Hay formas de indicarle al sistema que ejecute `fscck` en ciertos sistemas de archivos en el próximo arranque, incluso si parecen limpios:

- **Crear un archivo marcador:** Crear un archivo llamado `forcefsck` en la raíz del sistema. `sudo touch /forcefsck`. El script de arranque de SysVinit o un servicio de systemd lo detectará y forzará la comprobación. Se elimina automáticamente después.
- **Parámetro del Kernel:** Añadir el parámetro `fscck.mode=force` a la línea de comandos del kernel en el gestor de arranque (GRUB). Esto se puede hacer temporalmente (editando la entrada en el menú de GRUB) o permanentemente (modificando `/etc/default/grub` y regenerando `grub.cfg`).

Monitorización de Inodos:

- Además del espacio en disco (`df -h`), es importante monitorizar el uso de **inodos**. Un inodo es una estructura de datos que describe un archivo o directorio (metadatos). Cada archivo o directorio consume un inodo.
- `df -i`: Muestra el número de inodos utilizados y disponibles en cada sistema de archivos.
- Quedarse sin inodos (IUsed cerca del 100%) significa que no puedes crear nuevos archivos o directorios en ese sistema de archivos, aunque haya mucho espacio en disco libre (ej: si tienes millones de archivos pequeños).

Herramientas para Encontrar Archivos Grandes:

Si un sistema de archivos se llena, necesitas encontrar qué archivos o directorios ocupan espacio.

- `du` (disk usage): Estima el espacio usado por archivos y directorios.
 - `du -sh <directorio>`: Muestra el tamaño total de un directorio en formato legible por humanos.
 - `du -sch <directorio>/* | sort -h`: Lista el tamaño de los elementos dentro de un directorio y los ordena por tamaño.
- `find`: Puede buscar archivos por tamaño.
 - `find <directorio> -size +<tamaño>`: Busca archivos mayores que el tamaño especificado (ej: `+1G` para `>1GB`, `+1M` para `>1MB`).