



Examen 104 - Dispositivos, Sistemas de Archivos Linux y Jerarquía Estándar

104.7 Encontrar archivos del sistema y ubicarlos en el lugar correcto

Teoría

Un sistema Linux típico contiene miles o millones de archivos dispersos en una estructura de directorios compleja. Saber cómo encontrar archivos de manera eficiente y entender dónde deben ubicarse los diferentes tipos de archivos es esencial para la administración.

Concepto Clave: El Estándar de Jerarquía del Sistema de Archivos (FHS - Filesystem Hierarchy Standard)

El FHS es un estándar que define la estructura principal de directorios en los sistemas operativos tipo Unix, incluyendo Linux. Su objetivo es proporcionar una disposición consistente para archivos y directorios, independientemente de la distribución, lo que facilita la ubicación de archivos por parte de usuarios y programas, y simplifica el desarrollo de software y scripts portables.

Directorios Clave según FHS:

- `/`: El directorio raíz. La cima de la jerarquía. Contiene solo subdirectorios necesarios para el arranque y para montar otros sistemas de archivos.
- `/bin`: Binarios esenciales de usuario (comandos básicos que necesitan estar disponibles en modo monousuario, ej: `ls`, `cat`, `mv`, `rm`, `bash`).
- `/sbin`: Binarios esenciales del sistema (comandos de administración que necesitan estar disponibles en modo monousuario, ej: `fdisk`, `fsck`, `init`, `ifconfig` - aunque algunos están migrando a `/usr/sbin`).
- `/etc`: Archivos de configuración específicos de la máquina. Contiene archivos de configuración para la mayoría de los servicios y programas del sistema (ej: `/etc/passwd`, `/etc/fstab`, `/etc/ssh/sshd_config`). **No** contiene binarios ni archivos de estado variable.
- `/home`: Directorios personales de los usuarios. Cada usuario suele tener un subdirectorio aquí (ej: `/home/tu_usuario`).
- `/usr`: Jerarquía secundaria. Contiene la mayoría de las aplicaciones de usuario y utilidades. Históricamente "Unix Software Resource". A menudo montado de solo lectura.
 - `/usr/bin`: Binarios de usuario no esenciales.
 - `/usr/sbin`: Binarios del sistema no esenciales.
 - `/usr/lib`: Bibliotecas compartidas no esenciales.
 - `/usr/share`: Datos compartidos independientes de la arquitectura (documentación, iconos, archivos de localización).
 - `/usr/local`: Jerarquía terciaria para software instalado localmente (compilado desde fuentes, etc.) para evitar conflictos con archivos gestionados por el sistema de paquetes.

- `/var`: Archivos de datos variables. Contenido que cambia continuamente mientras el sistema está en ejecución.
 - `/var/log`: Archivos de registro (logs).
 - `/var/cache`: Caché de aplicaciones (ej: paquetes descargados por gestores de paquetes).
 - `/var/spool`: Colas de tareas (impresión, correo, cron).
 - `/var/tmp`: Archivos temporales que deben preservarse entre reinicios.
- `/opt`: Software de terceros opcional. Suele contener software instalado por proveedores externos en subdirectorios separados (ej: `/opt/google/chrome`).
- `/tmp`: Archivos temporales. Suelen borrarse en cada reinicio o con frecuencia. A menudo montado como `tmpfs` (en RAM).
- `/proc`: Sistema de archivos virtual para información del proceso y del kernel.
- `/sys`: Sistema de archivos virtual para información del hardware y del kernel.
- `/dev`: Archivos especiales de dispositivos.
- `/mnt`: Punto de montaje temporal para sistemas de archivos montados manualmente.
- `/media`: Punto de montaje para medios extraíbles (CD-ROMs, USB drives).
- `/srv`: Datos para servicios proporcionados por el sistema (ej: datos de servidores web, FTP).
- `/lib`, `/lib64`: Bibliotecas compartidas esenciales (necesarias incluso en modo monousuario).

Comandos para Encontrar Archivos:

1. `find`:

- Busca archivos en una jerarquía de directorios en *tiempo real*. Es potente y flexible, permitiendo buscar por nombre, tipo, tamaño, tiempo de modificación/acceso/cambio, permisos, propietario, etc.
- `find <ruta_inicio> [criterios] [acción]`
- `find . -name "mi_archivo.txt"`: Busca en el directorio actual (.) y subdirectorios un archivo llamado `mi_archivo.txt`. Los patrones suelen ir entre comillas para evitar que la shell los interprete.
- `find /home -name "*.jpg"`: Busca archivos `.jpg` en `/home` y subdirectorios.
- `find /var/log -type f -size +1M`: Busca archivos regulares (`-type f`) en `/var/log` con tamaño mayor a 1MB (`-size +1M`).
- `find /etc -perm 644`: Busca archivos con permisos 644 en `/etc`.
- `find /tmp -mtime +7`: Busca archivos en `/tmp` modificados hace más de 7 días.
- `find / -user tu_usuario`: Busca archivos propiedad de `tu_usuario` en todo el sistema (requiere `sudo` para evitar errores de permisos en algunos directorios).

- `find . -name "conf" -ls`: Busca archivos/directorios llamados "conf" y muestra información detallada (-ls).
- `find . -name "core" -delete`: Busca archivos llamados "core" y los elimina (¡usar con precaución!).

2. **locate**:

- Busca archivos usando una base de datos pre-indexada (creada por el comando `updatedb`, que suele ejecutarse diariamente por un cron job).
- Es mucho *más rápido* que `find` porque no recorre el sistema de archivos en tiempo real.
- `locate <nombre_archivo_o_patron>`
- `locate passwd`: Busca entradas en la base de datos que contengan "passwd".
- `locate "*.conf"`: Busca archivos que terminan en `.conf` (el patrón se interpreta por la propia base de datos, no la shell).
- **Limitación**: La base de datos no está siempre actualizada; los archivos creados o eliminados desde la última ejecución de `updatedb` no aparecerán o seguirán apareciendo incorrectamente. La base de datos por defecto a menudo excluye directorios volátiles como `/tmp` y directorios de usuarios por privacidad/seguridad.
- `sudo updatedb`: Actualiza la base de datos de `locate`.

3. **which**:

- Busca el *camino completo (path)* a un comando ejecutable. Solo busca en los directorios listados en la variable de entorno `PATH`.
- `which ls`: Muestra la ruta completa al ejecutable `ls` (ej: `/usr/bin/ls`).

4. **whereis**:

- Busca el binario, el código fuente y la página de manual de un comando. Busca en ubicaciones estándar predefinidas, no solo en `PATH`.
- `whereis ls`: Muestra algo como `ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz`.

Ubicando Archivos Correctamente:

La colocación correcta de archivos es esencial para la administración del sistema, el cumplimiento de estándares y la interoperabilidad:

- Los binarios compilados manualmente deben ir en `/usr/local/bin`.
- Las bibliotecas asociadas a esos binarios deben ir en `/usr/local/lib` o `/usr/local/lib64`.
- Los archivos de configuración específicos de una aplicación instalada localmente podrían ir en `/usr/local/etc` (menos común) o más a menudo en subdirectorios de `/etc`.
- Los archivos de configuración de servicios instalados por el gestor de paquetes siempre van en `/etc`.
- Los scripts personalizados de administración para el sistema local deben ir en `/usr/local/sbin` o `/opt/bin` si son parte de una suite.

- Los archivos de log van en `/var/log`.
- Los cachés de programas van en `/var/cache`.
- Los archivos temporales para una sola sesión o que no necesitan persistir van en `/tmp`.
- Los archivos temporales que sí deben persistir entre reinicios van en `/var/tmp`.

Seguir el FHS es una buena práctica de administración.