

102.1 Diseño del particionado del disco duro

Teoría Profunda:

Un diseño de particionamiento adecuado es fundamental para la flexibilidad, seguridad y estabilidad de un sistema Linux. Implica dividir el disco físico en secciones lógicas.

1. Tipos de Tablas de Partición:

- **MBR (Master Boot Record):** El método tradicional. Reside en el primer sector del disco.
 - Soporta hasta 4 particiones primarias.
 - Para tener más de 4, una partición primaria se designa como "extendida", y dentro de ella se crean "particiones lógicas".
 - Limitación de tamaño: Las particiones MBR no pueden ser mayores de 2 TB.
 - El MBR también contiene una pequeña parte del código del gestor de arranque.
- **GPT (GUID Partition Table):** El estándar moderno asociado con UEFI, pero también puede usarse con BIOS (requiere un gestor de arranque que lo soporte, como GRUB2).
 - Utiliza identificadores únicos globales (GUID) para cada partición.
 - Soporta muchísimas particiones (típicamente 128 por defecto).
 - Soporta tamaños de disco y partición mucho mayores (hasta 8 ZiB).
 - Almacena copias de la tabla de partición al principio y al final del disco para mayor redundancia.
 - No almacena código de arranque en la tabla de partición principal como MBR. En sistemas UEFI, el gestor de arranque se instala en la ESP (EFI System Partition).

2. Particiones Comunes en Linux:

- **/ (Raíz - Root):** Contiene el sistema de archivos principal, incluyendo el kernel, el directorio `/bin`, `/etc`, `/lib`, `/usr`, etc. Es *esencial* y debe estar presente.
- **/boot:** Contiene los archivos necesarios para el proceso de arranque, incluyendo el kernel, el `initramfs` y los archivos de configuración del gestor de arranque (ej: `/boot/grub`). Es buena práctica tenerla como una partición separada, especialmente si la partición raíz utiliza LVM o cifrado, ya que el gestor de arranque a menudo no puede leer sistemas de archivos complejos al inicio. También es necesaria para UEFI (la ESP suele montarse aquí o en `/boot/efi`). Si se utiliza MBR, debe ser una partición *primaria*.
- **swap:** Espacio de intercambio utilizado cuando la memoria RAM se agota. El kernel mueve páginas de memoria inactivas al espacio de swap. Se recomienda tener un tamaño de swap al menos igual a la RAM, o más si se necesita hibernación. Es una partición sin sistema de archivos, marcada con un tipo específico (Linux swap).

- **/home:** Contiene los directorios personales de los usuarios. Separarla facilita las copias de seguridad, reinstalaciones del sistema operativo sin afectar los datos de usuario, y permite usar cuotas de disco por usuario más fácilmente.
- **/var:** Contiene datos variables del sistema, como archivos de log (`/var/log`), cachés de paquetes (`/var/cache/apt`), colas de impresión (`/var/spool`), archivos temporales para reinicios (`/var/tmp`). Puede crecer considerablemente. Separarla previene que el crecimiento excesivo llene la partición raíz.
- **/tmp:** Archivos temporales. A menudo montada como un `tmpfs` (sistema de archivos en RAM) para mejorar el rendimiento y la seguridad, o como una partición separada que se limpia en cada arranque.
- **/usr:** Contiene la mayoría de las aplicaciones de usuario y utilidades. Históricamente significaba "Unix Software Resource". Aunque ahora es parte fundamental del sistema raíz en la mayoría de las instalaciones por defecto, en entornos empresariales a veces se considera separada para facilitar montarla de solo lectura y compartirla entre sistemas (thin clients). Contiene `/usr/bin`, `/usr/sbin`, `/usr/lib`, `/usr/share`.

3. Sistemas de Archivos:

- Una vez creadas las particiones, deben ser formateadas con un sistema de archivos.
- Los sistemas de archivos comunes en Linux incluyen:
 - **Ext4:** El sistema de archivos por defecto en la mayoría de las distribuciones modernas. Evolución de ext3, con mejor rendimiento, tamaños de archivos y particiones mayores, y otras características. Es un sistema de archivos "con journal" (lleva un registro de transacciones) para recuperaciones más rápidas tras fallos de energía.
 - **XFS:** Común en RHEL/CentOS/Fedora. Excelente rendimiento para sistemas de archivos muy grandes y cargas de trabajo intensivas. También con journal.
 - **Btrfs (B-tree File System):** Un sistema de archivos moderno con características avanzadas como snapshots, compresión transparente, subvolúmenes, y gestión de múltiples dispositivos/RAID a nivel del sistema de archivos.
 - **ZFS:** Otro sistema de archivos avanzado (originalmente de Sun/Oracle, con una implementación open source como OpenZFS). Similar a Btrfs en características (snapshots, pools, datasets, etc.).
 - **FAT32/exFAT:** Usados para la ESP en UEFI y para compatibilidad con Windows y dispositivos USB.
 - **NTFS:** Usado para acceder a particiones de Windows (generalmente requiere drivers FUSE).

4. LVM (Logical Volume Management):

- Una capa de abstracción sobre las particiones físicas. Permite crear volúmenes lógicos (LVs) que pueden abarcar múltiples discos físicos o particiones.
- **Conceptos LVM:**

- **Physical Volumes (PVs):** Discos o particiones inicializados para ser utilizados por LVM.
- **Volume Groups (VGs):** Agrupaciones de uno o más PVs. El espacio total de los PVs en un VG está disponible para crear LVs.
- **Logical Volumes (LVs):** Son las "particiones" lógicas que se crean a partir del espacio en un VG. Son las que se formatean con un sistema de archivos y se montan.
- **Ventajas de LVM:**
 - Flexibilidad: Los LVs pueden redimensionarse fácilmente (ampliarse o reducirse).
 - Snapshots: Se pueden crear copias puntuales (snapshots) de LVs para copias de seguridad consistentes.
 - Striping: Distribuir datos a través de múltiples discos para mejorar el rendimiento (similar a RAID 0).
 - Mirroring: Replicar datos en múltiples discos para redundancia (similar a RAID 1).
 - Facilita la gestión de espacio en entornos dinámicos.
- **Desventajas:** Un poco más complejo de configurar inicialmente que las particiones tradicionales. Requiere que el initramfs contenga soporte para LVM para montar la partición raíz si reside en un LV.

5. Consideraciones para el Diseño:

- **Propósito del sistema:** Un servidor web puede necesitar un `/var` grande, mientras que una estación de trabajo puede priorizar `/home`.
- **Número de usuarios:** Si hay muchos usuarios, separar `/home` es casi obligatorio.
- **Seguridad:** Montar particiones como `/tmp`, `/var/tmp`, `/home` con opciones como `noexec` (no permitir ejecución de binarios), `nodev` (no interpretar nodos de dispositivo), `nosuid` (ignorar bits SUID/SGID) aumenta la seguridad. Montar `/usr` de solo lectura (`ro`) también es una opción en algunos entornos.
- **Espacio de crecimiento:** Asignar suficiente espacio a las particiones que tienden a crecer (`/var`, `/home`). LVM ayuda mucho aquí.
- **Rendimiento:** Separar particiones utilizadas intensivamente (`/var`, `/tmp`) en discos físicos diferentes (o SSDs) puede mejorar el rendimiento.
- **Respaldo y recuperación:** Separar datos importantes (`/home`, `/var/www`, bases de datos) en particiones separadas facilita las copias de seguridad.
- **Sistemas dual-boot:** Planificar el espacio para otros sistemas operativos y sus particiones (ej: NTFS para Windows).
- **UEFI vs BIOS:** La elección de MBR o GPT y la necesidad de una ESP depende del firmware.

Conceptos Clave: MBR, GPT, Partición Primaria, Partición Extendida, Partición Lógica, ESP, `/`, `/boot`, `swap`, `/home`, `/var`, `/t`

