Hacman Viorica Gabriela
42 Roma Luiss
10-02-2025

Born2beroot

Le partizioni del disco

Le Partizioni di un Disco Rigido: Teoria e Funzionamento

Quando si installa un sistema operativo o si organizza un disco rigido, è fondamentale comprendere il concetto di **partizioni**. Una partizione è una suddivisione logica di un disco fisico, che consente di gestire meglio i dati e di supportare più sistemi operativi sullo stesso hardware. Esistono tre tipi principali di partizioni: **primaria, estesa e logica.**

1. Partizione Primaria

La **partizione primaria** è quella in cui può essere installato un sistema operativo. Il BIOS o l'UEFI riconoscono solo le partizioni primarie come avviabili (bootable).

- Un disco può contenere al massimo **quattro** partizioni primarie.
- In alternativa, si possono avere tre partizioni primarie e una partizione estesa.
- È possibile formattarla con vari file system, come **ext4**, NTFS, FAT32, ecc.
- Una di queste partizioni può essere impostata come **attiva**, ovvero quella da cui il sistema operativo verrà avviato.

2. Partizione Estesa

La partizione estesa è un tipo speciale di partizione primaria che serve a superare il limite delle quattro partizioni primarie su un disco.

- Un disco può contenere una sola partizione estesa.
- La partizione estesa non può contenere direttamente dati o un sistema operativo.
- Al suo interno possono essere create partizioni logiche, che funzionano come unità separate.
- È utile quando si devono creare più di quattro partizioni su un singolo disco.

3. Partizione Logica

Le partizioni logiche sono create all'interno di una partizione estesa. Sono trattate dal sistema operativo come unità indipendenti e possono contenere dati o sistemi operativi secondari.

- Sono contenute esclusivamente in una partizione estesa.
- Possono essere formattate con diversi file system, come ext4, NTFS o FAT32.

Il numero massimo di partizioni logiche varia:

- In teoria: fino a 23 partizioni logiche per una partizione estesa.
- In Linux: il limite è ridotto a 15 partizioni logiche, un numero comunque sufficiente per la maggior parte delle configurazioni.

1. Logical Volume Management (LVM)

Cos'è LVM?

LVM è un sistema di gestione dello storage che introduce un livello di astrazione sopra i dispositivi fisici. A differenza delle partizioni tradizionali, LVM permette una gestione più flessibile dello spazio su disco, consentendo operazioni avanzate come ridimensionamento dinamico, snapshot e gestione di più dischi come un'unica unità logica.

Struttura Gerarchica di LVM

LVM utilizza una struttura gerarchica che comprende tre livelli:

1. Physical Volumes (PV)
   - Sono i dispositivi fisici di storage, come dischi rigidi o SSD.
   - Possono essere intere partizioni o dischi interi.
   - Vengono inizializzati con il comando `pvcreate`.
2. Volume Groups (VG)
   - Sono gruppi di volumi fisici che aggregano più PV in un unico spazio di archiviazione gestito in modo dinamico.
   - Permettono di combinare più dischi in un unico pool di storage.
   - Si creano con il comando `vgcreate`.
3. Logical Volumes (LV)
   - Sono le "partizioni virtuali" all'interno di un Volume Group.
   - Possono essere ridimensionate dinamicamente, senza dover riavviare il sistema.
   - Si creano con il comando `lvcreate`.

Vantaggi di LVM

✅ **Flessibilità**: Possibilità di ridimensionare volumi logici senza preoccuparsi della struttura delle partizioni.
✅ **Snapshot**: Creazione di snapshot per backup senza interrompere l'operatività del sistema.
✅ **Gestione dinamica**: Possibilità di aggiungere nuovi dischi senza dover ridisegnare la tabella delle partizioni.

---

2. LUKS (Linux Unified Key Setup)

Cos'è LUKS?

LUKS è lo standard di crittografia per i dischi in Linux, utilizzato per proteggere i dati da accessi non autorizzati. A differenza di altri metodi di cifratura, LUKS offre una gestione centralizzata delle chiavi di crittografia ed è compatibile con vari strumenti di sicurezza.

VirtualBox offre la cifratura dei dischi virtuali attraverso tre semplici passaggi:

1. Attivazione: Nelle impostazioni del disco virtuale → Attivare cifratura → Impostare password
2. Utilizzo: All'avvio della VM → Inserire password → Accesso al disco
3. Funzionamento: Dopo lo sblocco → Utilizzo trasparente → Cifratura/decifratura automatica

Questo meccanismo protegge i dati della VM anche se qualcuno accede fisicamente ai file del disco virtuale.

3. Gerarchia delle Directory in Linux

Struttura del File System di Linux

Linux segue la **Filesystem Hierarchy Standard (FHS)**, che definisce la disposizione delle directory principali.

📂 **/ (root)**
- È la directory principale del sistema, da cui dipendono tutte le altre directory.
- Contiene i file essenziali per l'avvio e il funzionamento del sistema.

📂 **/home**
- Contiene le directory personali degli utenti.
- Ogni utente ha una propria sottodirectory (`/home/nomeutente`).

📂 **/var**
- Contiene dati variabili come log di sistema, cache e spool di stampa.
- Ad esempio, `/var/log` memorizza i log degli eventi del sistema.

📂 **/tmp**
- Area temporanea per i file creati dalle applicazioni.
- Viene svuotata automaticamente ad ogni riavvio.

📂 **/srv**
- Contiene dati utilizzati dai servizi di rete, come siti web serviti da un web server.

📂 **swap**
- Spazio su disco utilizzato come memoria virtuale per supportare la RAM quando questa è piena.

SCRIPT SYSTEM INFO

Uno **script** è un file di testo contenente una serie di comandi che vengono eseguiti in sequenza. Gli script vengono utilizzati per automatizzare operazioni ripetitive, gestire il sistema e facilitare l'interazione con il computer.

Cos'è uno script?

Uno script è un insieme di istruzioni scritte in un linguaggio di scripting (come Bash, Python o Perl) e salvate in un file con un'estensione specifica (.sh per Bash, .py per Python, ecc.). Quando eseguito, il sistema interpreta ed esegue ogni comando nell'ordine in cui è scritto.
Gli script possono essere utilizzati per raccogliere e visualizzare informazioni sul sistema, come:

- Il nome del sistema operativo
- La versione del kernel
- L'uso della CPU e della RAM
- Lo spazio disponibile sul disco
- Gli utenti attualmente connessi

Come funziona uno script?
Creazione del file, come ad esempio script.sh
Scrittura dei comandi, si inseriscono i comandi desiderati all'interno del file.
Impostazione dei permessi (chmod ...)
Esecuzione → ./script.sh.
5. Importanza degli script
Gli script sono strumenti potenti per:

- Automatizzare attività amministrative
- Monitorare lo stato del sistema
- Eseguire operazioni ripetitive in modo rapido ed efficiente
- Personalizzare l'ambiente di lavoro

architettura di sistema

L'architettura si riferisce alla progettazione e all'organizzazione dei componenti di un computer, come il processore, la memoria e il sistema operativo.

1. Cos'è l'architettura di un computer?

L'architettura di un computer definisce il modo in cui le sue parti hardware e software interagiscono.
Per ottenere informazioni sull'architettura di sistema si usa il comando

uname -a

Il comando uname -a fornisce informazioni sul sistema operativo. L'opzione -a (equivalente a --all) mostra tutti i dettagli disponibili. (nome del sistema operativo, la versione del kernel, il tipo di macchina su cui e' in esecuzione il sistema, la data di compilazione del kernel).
NON include dettagli sull'hardware o sul processore. Se servono informazioni sulla CPU, si possono usare comandi aggiuntivi come

```
lscpu      #mostra dettagli sulla CPU
arch       #Mostra l'architettura del processore (ex. x86_64)
```

Esempio output di **uname -a**

```
Linux my-computer 5.15.0-84-generic #93-Ubuntu SMP Mon Jan 8 11:35:23 UTC 2024 x86_64 GNU/Linux
```

CORE FISICI DELLA CPU

I **core fisici** sono le unità di elaborazione presenti fisicamente all'interno di un processore (CPU). Ogni core può eseguire istruzioni in modo indipendente, permettendo al sistema di gestire più processi contemporaneamente.

1. Differenza tra core fisici e core logici

- **Core fisici** → Sono i componenti reali della CPU che eseguono le istruzioni.
- **Core logici** → Sono core virtuali generati tramite tecnologie come l'**Hyper-Threading** (Intel) o il **Simultaneous Multithreading (SMT)** (AMD), che permettono a un singolo core fisico di gestire più thread contemporaneamente.

Ad esempio, una CPU con **4 core fisici** e **Hyper-Threading** può gestire **8 core logici**.

`grep "physical id" /proc/cpuinfo | sort -u | wc -l` #per contare il numero di physical core

- **`grep "physical id" /proc/cpuinfo`**: Cerca le righe che contengono "physical id" nel file `/proc/cpuinfo`.
- **`sort -u`**: Ordina l'output e rimuove le righe duplicate.
- **`wc -l`**: Conta il numero di righe nell'output, che corrisponde al numero di CPU fisiche presenti nel sistema.

CORE VIRTUALI DELLA CPU

I **core virtuali** (o **core logici**) sono suddivisioni logiche di un core fisico, create tramite tecnologie come **Hyper-Threading (Intel)** o **Simultaneous Multithreading - SMT (AMD)**. Queste tecnologie permettono a un singolo core fisico di eseguire più thread contemporaneamente, migliorando l'efficienza del processore.

1. Differenza tra core fisici e core virtuali

- **Core fisici** → Sono le unità di elaborazione reali all'interno della CPU.
- **Core virtuali** → Sono divisioni logiche di un core fisico, che consentono di eseguire più thread simultaneamente.

`grep -c "processor" /proc/cpuinfo` #per visualizzare il numero di core virtuali

Cosa fa il comando?

- **`grep -c "processor" /proc/cpuinfo`**
  - Cerca tutte le righe che contengono la parola `"processor"` nel file `/proc/cpuinfo`.
  - Conta quante volte appare questa parola.
  - Il numero di occorrenze corrisponde al numero totale di **core virtuali** (ovvero, i processori logici).

RAM (RANDOM ACCESS MEMORY) - MEMORIA AD ACCESSO CASUALE

La RAM è la memoria volatile del computer, utilizzata per archiviare temporaneamente dati e programmi in esecuzione. È chiamata memoria ad accesso casuale perché permette di leggere e scrivere dati in qualsiasi posizione con la stessa velocità. A differenza dell'hard disk o dell'SSD, la RAM perde tutte le informazioni quando il computer viene spento.

Più RAM ha un sistema, più programmi può eseguire contemporaneamente senza rallentamenti. Quando la RAM è insufficiente, il sistema utilizza la memoria di swap, un'area del disco rigido che simula la RAM, ma con prestazioni molto inferiori.

IL COMANDO FREE

In ambiente Linux, il comando free viene utilizzato per mostrare informazioni sull'utilizzo della memoria, compresa la RAM totale, quella in uso e quella disponibile.

L'opzione `--mega` serve per visualizzare i valori in megabyte (MB) invece che in kilobyte (KB), rendendo i dati più leggibili.

Esempio di output del comando `free --mega`

```sh
               total        used        free      shared  buff/cache   available
Mem:            7980        2123        3746         349        2110        5234
Swap:          16383           0       16383
```

Le colonne hanno il seguente significato:

- **total** → Memoria RAM totale installata nel sistema.
- **used** → RAM attualmente utilizzata dai processi in esecuzione.
- **free** → Memoria completamente libera e non utilizzata.
- **shared** → Memoria usata da più processi contemporaneamente.
- **buff/cache** → Memoria occupata da buffer e cache per ottimizzare le operazioni di lettura/scrittura.
- **available** → Memoria stimata come disponibile per nuovi processi (può essere maggiore di "free" grazie al meccanismo di caching).

Utilizzo di `awk` per estrarre informazioni specifiche

`awk` è un potente linguaggio di scripting per il filtraggio e l'elaborazione di dati strutturati in forma testuale. Lo utilizziamo per estrarre solo le informazioni necessarie dal comando `free --mega`.

Estrazione della memoria utilizzata

```sh
free --mega | awk '$1 == "Mem:" {print $3}'
```

Spiegazione passo per passo:

1. `free --mega` → Visualizza i dettagli della RAM in megabyte.
2. `awk '$1 == "Mem:" {print $3}'`:
   - `$1 == "Mem:"` → Seleziona solo la riga che inizia con `"Mem:"` (escludendo altre righe come "Swap:").
   - `{print $3}` → Stampa il terzo valore della riga, corrispondente alla **memoria utilizzata (used)**.

Estrazione della memoria totale

```sh
free --mega | awk '$1 == "Mem:" {print $2}'
```

- `{print $2}` stampa il secondo valore della riga `"Mem:"`, ovvero la **RAM totale** installata nel sistema.

Esempio di output (su un sistema con 8 GB di RAM totale):

```sh
7980
```

Calcolo della percentuale di RAM utilizzata

```sh
free --mega | awk '$1 == "Mem:" {printf("(%.2f%%)\n", $3/$2*100)}'
```

Analisi del codice:

1. `$3/$2*100` → Divide la memoria usata per quella totale e moltiplica per 100 per ottenere la percentuale.
2. `printf("(%.2f%%)\n", valore)` → Stampa il valore con **due cifre decimali** e il simbolo `%`.
   - `%.2f` → Formatta il numero con **due decimali**.
   - `%%` → Stampa il carattere `%` (dato che `%` è un carattere speciale in `printf`).
3. `\n` → Va a capo dopo la stampa.

Esempio di output:

```sh
(26.61%)
```

Se la memoria totale fosse **7980 MB** e quella usata fosse **2123 MB**, il calcolo sarebbe:

$$\frac{2123}{7980} \times 100 = 26.61\%$$

Memoria del Disco

La memoria del disco si riferisce alla capacità di archiviazione di un'unità di memoria (HDD, SSD o altro dispositivo di archiviazione). A differenza della RAM, la memoria del disco è non volatile, il che significa che i dati vengono conservati anche dopo lo spegnimento del computer.

In Linux, per analizzare l'utilizzo dello spazio su disco, si utilizza il comando `df` (disk filesystem), che fornisce un riepilogo dello spazio occupato e disponibile sulle unità di memoria.

Il comando df

Il comando `df` permette di visualizzare lo spazio utilizzato e disponibile su tutte le partizioni del sistema. Per mostrare i valori in megabyte (MB), si usa l'opzione `-m`.

Mostrare la memoria usata sul disco

```
df -m | grep "/dev/" | grep -v "/boot" | awk '{memory_use += $3} END {print memory_use}'
```

Analisi del comando:

1. `df -m` → Mostra l'uso del disco in MB.
2. `grep "/dev/"` → Seleziona solo le righe contenenti `/dev/`, che indicano i dispositivi di archiviazione.
3. `grep -v "/boot"` → Esclude le righe contenenti `/boot`, poiché la partizione di avvio non deve essere considerata.
4. `awk '{memory_use += $3} END {print memory_use}'`:
   - `$3` rappresenta la **colonna dello spazio utilizzato**.
   - `memory_use += $3` somma i valori della colonna `$3`.
   - `END {print memory_use}` stampa il totale alla fine.

Esempio di output (se il totale dello spazio usato è 40500 MB):

sh Copy Edit

```
40500
```

Mostrare la memoria totale del disco (convertita in GB)

```
df -m | grep "/dev/" | grep -v "/boot" | awk '{memory_result += $2} END {printf("%.0fGb\n", memory_result/1024)}'
```

Calcolare la percentuale di memoria del disco utilizzata

```
df -m | grep "/dev/" | grep -v "/boot" | awk '{use += $3} {total += $2} END {printf("(%d%%)\n", use/total*100)}'
```

Percentuale di utilizzo della CPU

L'uso della CPU rappresenta il carico di lavoro attuale del processore. Un valore elevato può indicare che il sistema è sotto forte stress, mentre un valore basso suggerisce che il sistema è inattivo o ha risorse disponibili.
Per ottenere la percentuale di utilizzo della CPU in tempo reale, possiamo usare il comando vmstat, che fornisce statistiche dettagliate sul sistema, inclusi processi, memoria, I/O e CPU.

Ottenere l'utilizzo della CPU

```
vmstat 1 3 | tail -1 | awk '{print $15}'
```

1. `vmstat 1 3` → Prende 3 campioni a intervalli di 1 secondo.
2. `tail -1` → Seleziona solo l'ultima riga dell'output.
3. `awk '{print $15}'` → Estrae la 15ª colonna, che rappresenta la percentuale di CPU idle (inattiva).

Calcolare la percentuale di utilizzo della CPU

```
vmstat 1 3 | tail -1 | awk '{print 100 - $15"%"}'
```

- `100 - $15` → Sottrae la percentuale di CPU idle da 100 per ottenere l'uso effettivo.
- `"%""` → Aggiunge il simbolo **%** per indicare che il valore è una percentuale.

Perché prendere 3 campioni invece di 1?

Il primo campione di vmstat potrebbe contenere valori non affidabili perché rappresenta dati cumulativi dall'avvio del sistema. Prendendo almeno 3 campioni, possiamo escludere il primo e ottenere un valore più preciso.