

Born2Beroot

MACCHINA VIRTUALE

E' un computer simulato all'interno di un computer fisico.

E' un ambiente isolato che emula un sistema informatico completo, questo ambiente virtuale ha il proprio sistema operativo, memoria, spazio di archiviazione e risorse di elaborazione, tutto gestito e isolato dal sistema ospitante.

COME FUNZIONA UNA MACCHINA VIRTUALE

Una macchina virtuale funziona attraverso un software chiamato **Hypervisor - Virtual Machine Monitor (VMM)**, che gestisce e coordina i diversi comportamenti chiave:

- **CPU** → Central Processing Unit → E' il "cervello" del computer. Esegue i calcoli e le istruzioni del programma; coordina il funzionamento di tutte le altre parti del computer come la memoria, disco rigido, le periferiche.
- **RAM** → Random Access Memory → e' la memoria temporanea del computer che svolge due funzioni principali:
 - E' come un tavolo di lavoro dove il computer tiene i programmi e i dati che sta usando in quel momento. E' veloce da accedere, ma temporanea, quando spegni il PC si cancella. A differenza del disco rigido (che è come un magazzino permanente ma più lento), la RAM permette al computer di accedere rapidamente alle informazioni necessarie mentre stai usando i programmi.
- **Disco rigido** → Il disco rigido è il "magazzino" permanente del computer dove vengono conservati:
 - Tutti i tuoi file (documenti, foto, video, ecc.)
 - I programmi installati
 - Il sistema operativo (Windows, MacOS, Linux)Può essere di due tipi: HDD[(Hard Disk Drive)]i classici dischi meccanici con parti in movimento, più economici ma più lenti] o SSD [(Solid State Drive)]dischi di nuova generazione senza parti meccaniche, più veloci ma più costosi].
- **Scheda di rete**→ E' il componente che permette al computer di connettersi a Internet o a una rete locale. Funziona via cavo(Ethernet) o via wireless.
- **Dispositivi Input/Output**

Il compito di un Hypervisor e' quello di astrarre e distribuire le risorse hardware tra le diverse VM in modo che possano operare in maniera indipendente.

Gli Hypervisor possono essere di due tipi:

- **Bare-Metal** → eseguono direttamente sull'hardware senza bisogno di un sistema operativo sottostante; sono piu' efficaci e sicuri perche' hanno accesso diretto alle risorse fisiche. (ex: Microsoft Hyper-V, KVM)
- **Hosted** → sono installati come software applicativo sopra un sistema operativo esistente; sono piu' facili da configurare e utilizzare. Ha prestazioni più ridotte perche' hanno un livello di astrazione in piu'. (ex: VMware Workstation, Oracle virtualbox).

L'hypervisor si occupa di:

- Allocare le risorse fisiche del computer host tra le varie macchine virtuali
- Schedulare l'esecuzione delle istruzioni della CPU
- Gestire la memoria condivisa
- Coordinare l'accesso alle periferiche

LO SCOPO DELLE MACCHINE VIRTUALI

Le macchine virtuali sono computer simulati che girano all'interno di un computer fisico.

Questa tecnologia permette di eseguire più sistemi operativi contemporaneamente sulla stessa macchina, mantenendo ogni ambiente completamente separato e sicuro. È uno strumento essenziale nell'IT moderno per ottimizzare l'uso dell'hardware, testare software in sicurezza e gestire backup di interi sistemi. La virtualizzazione consente inoltre di spostare facilmente questi ambienti virtuali tra diversi computer fisici, garantendo flessibilità e continuità operativa.

DEBIAN

Debian è una delle distribuzioni Linux più antiche, creata nel 1993 da Ian Murdock. È un sistema operativo libero, basato sul Kernel Linux e mantenuto da una comunità di volontari. È nota per essere la base di molte distribuzioni popolari come UBUNTU.

Il sistema di gestione dei pacchetti Debian è uno dei punti forza di questa distribuzione Linux. È basato su APT (Advanced Package Tool). Questo sistema gestisce l'installazione, l'aggiornamento e la rimozione del software attraverso un approccio centralizzato e coerente.

Al cuore del sistema troviamo il formato **.deb**, che rappresenta il formato nativo dei pacchetti Debian. Un pacchetto **.deb** è un archivio che contiene non solo il software compilato pronto per l'installazione, ma anche tutte le informazioni necessarie per una corretta integrazione nel sistema. (i metadati che descrivono le dipendenze, gli script di pre e post installazione necessari per la configurazione corretta, e la documentazione associata).

A basso livello c'è **dpkg**, che gestisce direttamente i pacchetti **.deb**. Ogni pacchetto **.deb** contiene un file di controllo che specifica le sue relazioni con altri pacchetti attraverso diversi campi:

- **Depends**: dipendenze strettamente necessarie
- **Pre-Depends**: dipendenze che devono essere già installate e configurate prima dell'installazione
- **Recommends**: pacchetti fortemente raccomandati (installati di default)
- **Suggests**: pacchetti opzionali che possono migliorare le funzionalità
- **Conflicts**: pacchetti che non possono essere installati contemporaneamente
- **Provides**: capacità virtuali fornite dal pacchetto
- **Replaces**: pacchetti che questo pacchetto sostituisce

A livello superiore c'è **APT (Advanced Package Tool)** che:

1. Legge queste relazioni
2. Calcola l'ordine corretto di installazione
3. Risolve automaticamente le dipendenze
4. Gestisce le dipendenze virtuali scegliendo tra alternative possibili

L'architettura del sistema prevede diversi livelli di strumenti. A basso livello, **dpkg** gestisce direttamente i pacchetti **.deb**, mentre APT fornisce un'interfaccia di più alto livello che automatizza la gestione delle dipendenze e l'accesso ai repository. Strumenti come **aptitude** offrono ulteriori funzionalità avanzate e un'interfaccia interattiva.

APTITUDE

Aptitude, è un gestore di pacchetti avanzato per le distribuzioni Linux basate su DEBIAN, che combina le funzionalità di apt e dpkg con alcune caratteristiche aggiuntive.

Il cuore di Aptitude è il suo **motore di risoluzione delle dipendenze**. Quando un utente richiede l'installazione o la rimozione di un pacchetto, Aptitude costruisce un grafo completo delle dipendenze e analizza multiple strategie di risoluzione. *Per esempio, se un utente tenta di installare un pacchetto che entra in conflitto con software esistenti, Aptitude non si limita a rifiutare l'operazione, ma cerca attivamente soluzioni alternative, considerando anche la possibilità di rimuovere o aggiornare altri pacchetti per risolvere il conflitto.*

Gestione dello stato di sistema

Aptitude mantiene un database dettagliato delle azioni dell'utente e delle loro motivazioni. Questo permette al sistema di prendere decisioni più informate durante la risoluzione dei conflitti. Aptitude tiene traccia di come e perché hai installato i programmi sul tuo sistema. Se hai installato un programma di proposito, Aptitude farà il possibile per mantenerlo durante gli aggiornamenti, mentre è più probabile che rimuova i programmi che sono stati installati automaticamente solo come supporto ad altri programmi.

APPARMOR

(Application Armor)

È un sistema di sicurezza per Linux che implementa il controllo degli accessi obbligatorio - MAC (Mandatory Access Control). Il suo scopo principale è limitare i programmi a specifiche risorse di sistema. AppArmor si occupa di gestire le limitazioni sull'esecuzione di altri programmi, garantendo al contempo una solida protezione della memoria del sistema. Un aspetto cruciale del suo funzionamento è il costante monitoraggio e la registrazione di eventuali violazioni dei profili di sicurezza stabiliti.

I profili di sicurezza in AppArmor possono essere configurati secondo due modalità principali: la Enforcement Mode, che applica attivamente tutte le restrizioni definite, e la Complain Mode, che si limita a monitorare le attività senza bloccarle, risultando particolarmente utile durante le fasi di testing e configurazione. Tra i principali benefici, AppArmor offre un miglioramento della sicurezza del sistema, fornendo protezione anche contro vulnerabilità non ancora scoperte. In caso di compromissione di un'applicazione, il sistema limita efficacemente i potenziali danni. Un punto di forza particolare è la sua relativa semplicità di configurazione, specialmente se confrontata con alternative come SELinux. Un aspetto da non sottovalutare è la necessità di una manutenzione costante per mantenere i profili di sicurezza sempre aggiornati e funzionali.

Caratteristica	APT	APTITUDE	AppArmor
Tipo di strumento	Gestore Pacchetti Base	Gestore pacchetti avanzato	Sistema di sicurezza MAC
Interfaccia	CLI ¹ moderna e semplificata	CLI + interfaccia TUI ² interattiva	Profili di configurazione
Gestione Dipendenze	Base	Base	N/A
Uso principale	Installazione/rimozione pacchetti base	Risoluzione intelligente delle dipendenze, interfaccia interattiva, più opzioni di configurazione	Controllo di accessi applicazione
Caratteristiche distintive	Comandi semplici e intuitivi, sintassi moderna	aptitude search "-i linux-image"	profili per app, basato su path, integrato nel kernel
Esempio d'uso	apt install firefox	Media	Media-Alta
Complessità	Bassa	Utenti avanzati/admin	Amministratori di sistema
Target Utente	Utente base/intermedio		

Apt rappresenta la soluzione più accessibile per la gestione dei pacchetti, pensata per l'utente quotidiano che necessita di operazioni base come installare, aggiornare o rimuovere software. La sua forza sta nella semplicità e immediatezza d'uso.

Aptitude si posiziona come strumento più sofisticato, ideale per amministratori di sistema e utenti avanzati che necessitano di un controllo più granulare sulla gestione dei pacchetti. La sua interfaccia TUI e le capacità avanzate di risoluzione delle dipendenze lo rendono particolarmente utile in scenari complessi.

AppArmor, invece, opera su un piano completamente diverso: è un sistema di sicurezza integrato nel kernel che fornisce un livello aggiuntivo di protezione attraverso il controllo degli accessi delle applicazioni. È uno strumento essenziale per la sicurezza del sistema, ma richiede competenze specifiche per la sua gestione ottimale.

SISTEMA DI RILASCIO DEBIAN

IL sistema di rilascio di Debian si basa su tre rami di sviluppo principali che garantiscono diversi livelli di stabilità e aggiornamento del software.

Il ramo **Stable** rappresenta la versione ufficiale e stabile di DEBIAN, rilasciata ogni 2-3 anni. Questa versione è considerata la più affidabile e sicura, ideale per server e sistemi che richiedono massima stabilità. Per esempio, l'attuale versione Stable è Debian 12 "Bookworm", rilasciata nel 2023.

Il ramo **Testing** contiene i pacchetti che si stanno preparando per entrare nella prossima

¹ La CLI (Command Line Interface) è l'interfaccia a riga di comando che permette all'utente di interagire con il sistema operativo attraverso comandi testuali

² Text User Interface, crea una sorta di pseudo grafica utilizzando carattere ASCII

versione Stable. Il software in Testing ha già superato alcuni controlli di qualità iniziali, ma potrebbe ancora contenere bug o problemi. È un buon compromesso tra stabilità e modernità del software, spesso utilizzato da utenti desktop che desiderano programmi più recenti mantenendo una discreta affidabilità.

Unstable, soprannominato "Sid" (come il bambino che rompeva i giocattoli in Toy Story), è il ramo più dinamico e volatile. Qui confluiscono le versioni più recenti dei pacchetti software, spesso appena rilasciate dagli sviluppatori upstream. Sid è in continua evoluzione e può subire frequenti modifiche e rotture, rendendolo adatto principalmente agli sviluppatori e agli utenti più esperti che vogliono testare le ultime novità o contribuire allo sviluppo

Il flusso di sviluppo segue un percorso preciso: i nuovi pacchetti entrano prima in Unstable, dove vengono testati inizialmente. Se non presentano problemi critici per un certo periodo, migrano automaticamente in Testing. Infine, quando Testing raggiunge un livello di maturità sufficiente, viene "congelato" e dopo un'intensa fase di debug e correzioni diventa la nuova versione Stable.

VANTAGGI	SVANTAGGI
<p>Stabilità:</p> <ul style="list-style-type: none">affidabile e stabileTest approfonditiPerfetto per serverSupporto a lungo termine <p>Sicurezza:</p> <ul style="list-style-type: none">Aggiornamenti di sicurezza tempestiviPolitiche rigide sul software liberoProcessi di revisione del codice <p>Comunità:</p> <ul style="list-style-type: none">Grande comunità di sviluppatoriDocumentazione estesa <p>Compatibilità:</p> <ul style="list-style-type: none">Supporta molte architetture hardwareUtilizzabile sia su desktop che su server	<p>Aggiornamenti:</p> <ul style="list-style-type: none">Software non sempre aggiornato nella versione più recente, anche se si privilegia la stabilità rispetto alle ultime versioni. I pacchetti possono essere anche di diversi mesi più vecchi rispetto ad altre distribuzioni. <p>Complessità:</p> <ul style="list-style-type: none">Meno user-friendly rispetto ad altre distribuzioni come Ubuntu o Linux Mint, con una curva di apprendimento più alta per chi proviene da Windows o macOS. <p>Requisiti minimi:</p> <ul style="list-style-type: none">CPU: 1 GHzRAM: 512 MB (minimo), 2 GB (raccomandati)Spazio su disco: 10 GB (minimo), 20 GB (raccomandati)Scheda grafica base con supporto <u>SVGA</u>³

ROCKY

ROCKY Linux è una distribuzione Linux enterprise-grade, creata come sostituto di CentOS dopo che Red Hat ha modificato la direzione di CentOS. È una distribuzione binaria compatibile con Red Hat Enterprise Linux (RHEL). Una **distribuzione binaria** significa che il software viene distribuito in forma già compilata e pronta per l'esecuzione, invece che come codice sorgente da compilare. Nel caso specifico di ROCKY Linux, essere una "distribuzione binaria compatibile con RHEL" significa che: I pacchetti software sono precompilati esattamente come quelli di RHEL; I binari (programmi eseguibili) sono identici o molto simili a quelli di RHEL; Gli utenti non devono compilare il software da sorgente; È possibile utilizzare gli stessi pacchetti RPM compilati per RHEL.

Le principali differenze con Debian sono:

- ROCKY usa DNF come gestore dei pacchetti, mentre Debian usa APT
- ROCKY è basato su RHEL e usa pacchetti .rpm, Debian usa pacchetti .deb
- ROCKY punta alla stabilità enterprise e compatibilità RHEL, Debian ha un focus più generale
- ROCKY ha cicli di supporto più lunghi (tipicamente 10 anni), simili a RHEL

SELINUX SU ROCKY:

SELinux è un sistema di sicurezza avanzato che funziona secondo il principio del "controllo di accesso obbligatorio" (MAC - Mandatory Access Control). Su ROCKY è attivo di default e fornisce un livello aggiuntivo di sicurezza attraverso policies che definiscono quali processi possono accedere a quali risorse. È come avere, oltre alla serratura normale: Un badge elettronico (l'etichetta); Un sistema che controlla se quel badge specifico può accedere a quella specifica area (le policies).

³ SVGA (Super Video Graphics Array) è uno standard di visualizzazione grafica. Quando si parla di "scheda grafica base con supporto SVGA" significa che la scheda grafica deve essere in grado di supportare:

- Risoluzioni minime di 800x600 pixel (anche se oggi SVGA supporta risoluzioni molto più alte)
- Profondità di colore di almeno 16 bit (65.536 colori)
- Frequenza di aggiornamento di almeno 60 Hz

DNF (DANDIFIED YUM) SU ROCKY:

DNF è il gestore dei pacchetti predefinito, evoluzione di YUM. Permette di:

- Installare/rimuovere/aggiornare pacchetti software
- Gestire repository e dipendenze
- Fare rollback delle transazioni
- Gestire gruppi di pacchetti

Il sistema DNF (Dandified Yum) rappresenta un'evoluzione fondamentale nella gestione dei pacchetti per i sistemi Linux basati su Red Hat. La sua architettura concettuale si basa su diversi livelli che cooperano tra loro in modo integrato.

A livello di interfaccia, DNF gestisce l'interazione con l'utente interpretando i comandi e fornendo feedback appropriati sulle operazioni in corso. Il sistema di gestione delle dipendenze utilizza un risolutore basato su SAT⁴ (Boolean Satisfiability) che costruisce e analizza un grafo completo delle dipendenze per calcolare la soluzione ottimale per ogni installazione.

LA GESTIONE DEI REPOSITORY COSTITUISCE UN ALTRO ASPETTO CRUCIALE:

DNF si occupa dei metadati, della sincronizzazione con i server remoti e del mantenimento delle cache locali. Questo processo segue principi fondamentali come l'atomicità delle transazioni, la consistenza del sistema, l'isolamento delle operazioni e la durabilità delle modifiche apportate.

In modo semplice:

- **Atomicità:** o funziona tutto o niente. Come quando installi un gioco sul computer - o si installa completamente o non si installa proprio.
- **Consistenza:** il sistema deve sempre funzionare correttamente. È come un puzzle - tutti i pezzi devono incastrarsi bene insieme.
- **Isolamento:** quando fai più cose insieme, non devono disturbarsi a vicenda. Come quando più persone lavorano su progetti diversi senza interferire tra loro.
- **Durabilità:** quando finisci un'installazione, le modifiche rimangono anche se spegni il computer. Come quando salvi un documento - lo ritrovi anche dopo aver riavviato.

Per quanto riguarda i metadati, DNF implementa un sistema basato su database SQLite⁵ per la cache. Mantiene inoltre uno storico dettagliato delle transazioni per garantire tracciabilità e possibilità di rollback⁶. Rispetto al suo predecessore YUM, DNF presenta significative innovazioni architetturali: un'implementazione più modulare, una gestione ottimizzata della memoria, un'API meglio strutturata e documentata, oltre a un supporto nativo per le estensioni. Il sistema di risoluzione delle dipendenze rappresenta uno dei punti di forza di DNF, grazie all'utilizzo di algoritmi ottimizzati che gestiscono sia dipendenze deboli che forti, risolvendo automaticamente i conflitti secondo politiche di priorità ben definite.

⁴ Il problema SAT consiste nel determinare se esiste un'assegnazione di valori vero/falso alle variabili di una formula booleana che renda l'intera formula vera. Nel caso di DNF, le variabili rappresentano i pacchetti e le loro versioni, mentre le clausole della formula rappresentano le dipendenze e i conflitti tra pacchetti. Per esempio, se abbiamo un pacchetto A che dipende da B o C, e B è in conflitto con D, questo può essere espresso come una formula SAT: $(A \rightarrow (B \vee C)) \wedge (\neg B \vee \neg D)$. Il risolutore SAT in DNF cerca una soluzione che soddisfi tutte queste condizioni simultaneamente. Se trova una soluzione, questa rappresenta una combinazione valida di pacchetti che possono essere installati insieme. Se non esiste soluzione, significa che le dipendenze sono incompatibili.

⁵ SQLite viene usato come un contenitore (database) dove DNF salva informazioni importanti come nome dei programmi disponibili, quale versione è stata installata, da dove provengono i programmi(repository), quali altri programmi servono per farli funzionare. quando sono stati aggiornati l'ultima volta.

⁶ come se fosse un tasto annulla per le installazioni di programmi.