LEZIONE 1

mercoledì 11 ottobre 2023 10:47

COS'È UN INTERRUPTS?

Segnale asincrono di sistema che esprime la "necessità di attenzione" verso la CPU espressa da un componente hardware.

Un interrupt hardware obbliga il processore a commutare il suo stato per eseguire l'operazione indicata dall'interrupt stesso.

Svolto il compito, il processore tornerà a lavorare sull'attività precedentemente in corso.

Se questi interrupt sono molto frequenti possono mandare in stallo la CPU.

/PROC

Il file system /proc è un file system creato e mantenuto a run-time dal Kernel di Linux per tenere traccia dei vari processi che stanno funzionando sulla macchina e sul loro stato.

/PROC/SYS

Contiene molti parametri usati dal Kernel Linux; questa directory viene utilizzata per controllare e modificare alcuni di questi parametri.

/PROC/CPUINFO

Riporta informazioni sul processore (tipo, marca, modello e varie performance)

/PROC/DEVICES

Elenca i driver configurati nel Kernel corrente

/PROC/DMA

Mostra quali canali DMA sono in uso al momento

/PROC/FILESYSTEMS

Mostra i filesystem configurati, supportati nel Kernel

/PROC/INTERRUPTS

Visualizza gli interrupts in uso e quanti e quali sono usati

/PROC/IOPORTS

Mostra quali porte di I/O sono in uso al momento

COMANDI:

cat /proc/interrupts:

Questo file registra il numero di interrupts.

La prima colonna si riferisce al numero di IRQ(Interrupt Request).

Ogni CPU ha a propria colonna e il proprio numero di interruzioni per IRQ.

La colonna successiva riporta la tipologia di interruzione e l'ultima colonna contiene il nome del dispositivo che si trova su quell'IRQ.

sudo cat /proc/iports:

L'output di /proc/ioports fornisce un elenco delle porte attualmente registrate, utilizzate per la comunicazione di input o output con un dispositivo.

La prima colonna fornisce l'intervallo degli indirizzi di memoria (che vengono assegnati durante la fase di boot) delle porte I/O riservato per il dispositivo elencato nella seconda colonna.

Questo comando necessita dei permessi di amministratore, altrimenti mi restituisce in output tutti gli indirizzi di memoria pari a 0.

COS'È LA DMA?

Direct Memory Access è una funzionalità fornita da alcune architetture bus di computer che consente l'invio diretto dei dati da un dispositivo collegato, come un'unità disco, alla memoria principale della scheda madre del computer, liberando così la CPU dal coinvolgimento nel trasferimento dei dati, accelerando il funzionameno complessivo del computer.

cat /proc/dma:

Questo file contiene un elenco dei canali ISA DMA(Industry Standard Architecture Direct Memory Access) registrati in uso.

sudo /proc/iomem:

Questo file mostra la mappa attuale della memoria di sistema per ciascun dispositivo fisico.

La prima colonna visualizza i registri di memoria utilizzati, la seconda elenca il tipo di memoria situata all'interno di tali registri e visualizza quali registri di memoria vengono utilizzati dal kernel all'interno della RAM di sistema.

cat /proc/meminfo:

Questo file riporta una grande quantità di informazioni sull'utilizzo della RAM del sistema.

La maggior parte delle informazini /proc/meminfo vengono utilizzate dai comandi free, top e ps.

Il contenuto di questo file è:

- MemTotal: quantità totale di RAM utilizzabile, rappresentata in KiB(1024 B)

- MemFree: quantità di memoria RAM lasciata inutilizzata dal sistema

- Buffers: quantità in KiB di spazio di archiviazione temporaneo per i blocchi del disco non elaborato

- Cached: quantità di RAM fisica utilizzata come memoria cache

Sono presenti anche altri elementi come:

- Quantità totale di memoria swap disponibile

- Quantità di memoria utilizzata di recente

- Quantità di memoria utilizzata meno recentemente

- Quantità di memoria utilizzata dalle allocazioni dello stack del kernel effettuate per ciascuna attività nel sistema

DIFFERENZA TRA DISPOSITIVO HOT PLUG E COLD PLUG

L'hot plugging è l'aggiunta di un componente a un sistema informatico in esecuzione senza interruzioni significative del funzionamento del sistema.

Il collegamento a caldo di un dispositivo non richiede il riavvio del sistema.

Ciò è particolarmente utile per i sistemi che devono rimanere sempre in funzione, come un server.

Esempi di hot plug possono essere l'unità disco rigido e unità a stato solido, che possono essere aggiunte a un sistema di archiviazione; oppure dispositivi USB, come mouse, tastiere e stampanti.

Il cold plugging si riferisce alla situazione in cui un computer deve essere spento per aggiungere o rimuovere un componente dal computer.

Il collegamento a freddo viene spesso utilizzato come precauzione aggiuntiva per garantire che un componente non venga danneggiato durante la rimozione o la sostituzione.

Un dispositivo cold plug sostituito a caldo può causare malfunzionamenti e danni al dispositivo o al sistema.

Esempi di cold plug possono essere la RAM, la CPU e la scheda video.

COMANDI LS:

lscpu: restituisce informazioni sull'architettura della CPU da sysfs e /proc/cpuinfo.

Le informazioni restituite includono, ad esempio l'ID_venditore, la cpu family, il modello della CPU (model), il nome del modello (model_name), la velocità della CPU in megahertz (cpu mhx), la dimensione della cache (cache size), il numero di core (core number), i flag supportati dalla CPU (flags) e le informazioni sui socket e i nodi NUMA (Non-Uniform Memory Access).

Negli ambienti virtualizzati, le informazioni sull'architettura della CPU visualizzato riflette la configurazione operativa del guest sistema che è tipicamente diverso da quello fisico (host) sistema.

COS'È LA CPU?

La CPU (Control Processing Unit) è il componente principale di un computer che funge da "centro di controllo".

La CPU, chiamata anche processore "centrale", è un insieme complesso di circuiti elettronici che gestiscono il sistema operativo e le app della macchina.

La CPU interpreta, elabora ed esegue istruzioni, molto spesso dai programmi hardware e software in esecuzione sul dispositivo.

La CPU esegue operazioni aritmetiche, logiche e di altro tipo per trasformare i dati immessi in informazioni più utilizzabili.

COS'È IL PCI?

Peripheral Component Interconnect, è un interfaccia per aggiungere ulteriori componenti hardware a un sistema informatico.

Ad esempio, supponiamo che tu voglia aggiungere una scheda Ethernet al tuo computer in modo

che possa accedere a Internet e scambiare dati.

Bene, la scheda necessita di un protocollo per comunicare con il resto del sistema interno, PCI può essere l'interfaccia standard utilizzata per aggiungere questa scheda al tuo sistema.

Hai ancora bisogno di un driver per questa scheda affinché il kernel possa usarla, tuttavia PCI è lo slot, il bus e l'interfaccia che verranno utilizzati per aggiungere l'hardware nel sistema con un'interfaccia standard.

lspci:

Il comando Linux lspci (elenco PCI) visualizza informazioni su ciascun bus PCI nel sistema, inclusi dettagli sui dispositivi connessi al sottosistema PCI. Queste informazioni includono i seguenti campi visualizzati in ogni riga dell'output:

- Slot
- Classe
- ID Produttore
- ID Dispositivo

COS'È USB?

L'USB (Universal Serial Bus) è un'interfaccia comune che consente la comunicazione tra dispositivi e un controller host come un PC o uno smartphone.

Collega dispositivi periferici come fotocamere digitali, mouse, tastiere, stampanti, scanner, dischi rigidi esterni e unità flash.

Una USB ha lo scopo di migliorare il plug-and-play e consentire lo scambio a caldo.

Il plug-and-play consente al sistema operativo di configurare e rilevare un nuovo dispositivo periferico senza dover riavviare il computer.

lsusb:

È un comando di Linux che consente agli utenti di elencare i dispositivi USB collegati al sistema.

Questa unità fa parte del paccheto "usbutils", che fornisce utilità per visualizzare informazion sui bus USB nel sistema e sui dispositivi ad essi collegati.

lsusb -v / lsusb -t: restituisce in output alcune informazioni aggiuntive sulle periferiche.

COS'È IL KERNEL?

Il kernel è il componente principale di un sistema operativo.

Gestisce le risorse del sistema ed è un ponte tra l'hardware e il software del computer.

Un modulo del kernel, spesso chiamato driver, è un pezzo di codice che estende le funzionalità del kernel.

I moduli vengono compilati come moduli caricabili incorportati nel kernel.

I moduli caricabili possono essere caricati e scaricati nel kernel in esecuzione su richiesta, senza la necessità di riavviare il sistema.

Questi moduli vengono caricati su richiesta da "udev"(gestione dispositivi).

e sono memorizzati in /lib/modules/<kernel_version> (per trovare la versione del kernel in esecuzione utilizzare il comando uname -r).

lsmod:

Il comando lsmod viene utilizzato per visualizzare lo stato dei moduli nel kernel Linux.

Ciò che fa il comando è leggere /proc/modules e visualizzare il contenuto del file in un elenco ben formattato.

I dettagli che vengono restituiti in output sono:

- modulo

- dimensione modulo

- istanze attive

- quali sono i moduli richiamati (numero di volte che il modulo è stato caricato)

Questi moduli vengono richiamati, essendo i driver generici e che necessitano di ulteriori implementazioni (comandi specifici) per svolgere mansioni particolari.

modinfo:

Il comando modinfo estrae le informazioni dai moduli del kernel Linux forniti da linea di comando.

Modinfo elenca ogni attributo del modulo (nomecampo:valore), per una facile lettura.

esempio: modinfo pcspkr.

Per disattivare un modulo utilizziamo il comando sudo rmmod nomeModulo, mentre per riattivarlo utilizziamo sudo modprobe nomeModulo.

Quando inseriamo una nuova periferica nel nostro PC, per essere raggiungibile dalla CPU e quindi utilizzabile, devono essere caricati dei driver.

Quando colleghiamo una periferica gli viene assegnato un range di indirizzi di memoria virtuale.

La procedura che viene eseguita per rendere raggiungibile la periferica sono:

1. Inserimento della periferica

2. Interruption

3. Caricamento del driver (un daemon crea in /dev/ un nuovo file per rappresentare la nuova periferica appena inserita).

Una periferica riesce a comunicare con la CPU solamente tramite interrupt.

SCOLLEGAMENTO DRIVER:

In presenza di un malfunzionamento dela periferica possiamo utilizzare il terminale per scollegare un driver e, di conseguenza, disabilitare la periferica che necessitava di quel driver per comunicare con la CPU.

Questo metodo può essere usato per esempio quando stiamo lavorando su un server, per evitare il down-time.

Siccome ogni volta che riavvio il PC/Server i driver delle periferiche vengono reinstallati, se voglio che un driver non venga più reinstallato devo inserirlo in una blacklist.

COMANDO WATCH:

Il comando watch mi permette di lanciare il successivo comando ogni n secondi.

La sintassi è:

watch -n tempo

watch -n 0.1 cat /proc/interrupts

DIFFERENZA TRA BIOS E UEFI:

Il BIOS (Basic Input Output System) e il UEFI (Unified Extensible Firmware Interface) sono i programmi utilizzati dal microprocessore di un computer per avviare il sistema informatico dopo l'accensione.

Gestiscono inoltre il flusso di dati tra il Sistema Operativo del pc e i dispositivi collegati.

Ciò che li distingue è che il BIOS inizializza le periferiche e cerca un eseguibile nell'MBR (primi 512 B) che si trova nel primo settore del disco principale.

L'UEFI, d'altra parte, è considerato come un sistema operativo a parte, che restituisce a video non solo i dispositivi fisici da cui poter eseguire la fase di boot, ma possiamo vedere anche ogni eseguibile .efi installato.

L'UEFI risulta più flessibile, quindi non si limita a cercare solamente nei primi 512 B il boot loader, bensì esegue ogni tipo di eseguibile (che abbia al suo interno un flag di default) che riesce a trovare (come per esempio il Sistema Operativo nella fase di boot).

FASE DI AVVIO DI UN COMPUTER:

-> Fase di boot

-> Viene caricato il bootloader GRUB: programma minimale che carica tutto quello che serve per avviare il Sistema Operativo; a questo punto viene caricato l'Initial RunDisk (disco che contiene driver indispensabili per montare la partizione del Sistema Operativo e altri driver minimali come per esempio quelli per la tastiera, il mouse e alcuni driver grafici per ricevere un output).

-> Viene avviato il Kernel, ovvero il cuore del sistema operativo che mette in counicazione il Software con l'Hardware in modo che gli applicativi non debbano dipendere dalle specifiche hardware.

-> Viene avviato l'Init System, un meta programma che serve a lanciare tutti i servizi che rendono operativo il nostro pc, come il servizio di networking, il servizio SSH e altri servizi che girano in background durante la fase di avvio.

-> Viene caricato il Display Manager o schermata di login

-> Una volta che l'autenticazione viene verificata abbiamo accesso al desktop

NVRAM --> FASE di GRUB --> KERNEL --> INIT

NVRAM --> FASE di GRUB --> KERNEL --> SHELL (vengono eseguiti degli script/programmi con il massimo dei privilegi) -> SHELL DI ROOT -> init=/bin/sh

In /etc/default/grub --> vedo le impostazioni del grub (GRUB_TIMEOUT = 5, GRUB_TMEOUT_STYLE = menu)

COMANDI:

/sbin/init --version-> per conoscere il sistema di init che stiamo utilizzando

ls -l /etc/systemd/system/display-manager.service -> per conoscere quale display manager stiamo

utilizzando

cat /etc/passwd o cat /etc/shadow -> per visualizzare l'elenco delle password

echo $DESKTOP_SESSION -> per conoscere il desktop environment attualmente in uso

screenfetch -> per ricevere in output un sommario con informazioni generali sul sistema

neofetch -> per ricevere in output un sommario con informazioni generali sul sistema

SYSTEMCTL:

Systemctl è il tool con cui controlliamo e gestiamo i servizi systemd.

Systemd è costituito da un set di daemon, librerie e strumenti che cnsentono l'amministrazione e la configurazione del sistema e interagiscono con il kernel del sistema GNU/Linux.

COMANDI:

systemctl get-default restituisce il run-level attualmente in uso.

systemctl set-deafult permette di modificare il run-level di avvio.

COS'È LA NVRAM:

La NVRAM (Not Volatile Random Access Memory) è un chip fisso con pochi KB di memoria che contiene le preferenze di boot.

SYSTEMD vs SYSTEMV:

SystemV e systemd sono due diverse modalità di gestire l'avvio e la gestione dei servizi e dei processi in un sistema operativo basato su Linux.

Entrambi offrono un approccio per controllare il comportamento del sistema durante l'avvio e in diverse situazioni, ma differiscono nella loro architettura e implementazione.

SystemD è il successore di SystemV, fornisce un avvio molto più rapido e una migliore gestione delle dipendenze.

SystemD gestisce i processi di avvio tramite file .service, mentre SystemV gestisce i processi tramite script di shell in /etc/init*.

COM'è FATTA UN'UNITà SYSTEMD?

Possiamo controllare la struttura di un servizio fatto partire da SystemD con questo comando: cat /lib/systemd/system/nome_servizio.

All'interno di questo documento possiamo trovare diverse informazioni riguardo la modalità di avvio del servizio, comprese le sue dipendenze e dopo quali servizi deve essere avviato.

In SystemV, il sistema operativo offre sette runlevel, ognuno dei quali definisce uno stato diverso per il sistema.

Questi runlevel determinano quali servizi e processi vengono avviati all'avvio del sistema:

Runlevel 0: Arrestare il sistema

Runlevel 1: Modalità singolo-utente:

Runlevel 2: Modalità multi-utente:

Runlevel 3: Modalità di default, tutti i sistemi sono attivati

Runlevel 4: Non viene usato, riservato per scopi futuri

Runlevel 5: Sessione grafica, modalità multiutente, funzionalità di networking attive

Runlevel 6: Riavvio del sistema

Con l'introduzione di systemd, il concetto di runlevel è stato sostituito dai "target".

I target rappresentano diversi stati del sistema e specificano quali servizi devono essere attivati in ciascuno di essi.

Target 1: PowerOff.target

Target 2,3,4: Rescue.target

Target 5: Graphical.target

Target 6: Reboot.target

In /etc/ troviamo delle cartelle nominate "rc*.d" (rc1.d, rc2.d, …) contententi gli script da terminare quando si avvia un determinato run level.

Nella NVRAM è contenuto il GRUB EFI, che funge da bootloader.

Il bootloader è il primo programma che viene eseguito all'avvio del computer e ha il compito di caricare il kernel del sistema operativo nella memoria del computer.

Il bootloader può essere configurato per avviare diverse opzioni di sistema operativo, come versioni diverse del kernel o sistemi operativi multipli, se presenti.

Una volta caricato il kernel, vengono anche caricati i moduli dei driver necessari per inizializzare le periferiche hardware del sistema.

Questi driver consentono al sistema operativo di comunicare con le periferiche hardware, come schede di rete, schede video e dispositivi di archiviazione.

LEZIONE 2

lunedì 9 ottobre 2023 07:44

LOG: Log is the file extension for an automatically produced file that contains a record of events from certain software and OS.
Log files are used to show all events associated with the system or application that created them.

The point of a log file is to keep track of what's happening behind the scenes and if something should happen within a complex system, you have access to a detailed list of events that took place before the malfunction. Basically, whatever the application, server, or OS thinks needs to be recorded.

Those files are achieved into the /var/log/ folder.

DMESG:
In linux i can control log files with the command dmesg.
The dmesg command allows you to review the messages that are stored in the ring buffer.

What is a Ring Buffer?

A ring buffer is a memory space reserved for messages. It is simple in design, and of a fixed size. When it is full, newer messages overwrite the oldest messages. Conceptually it can be thought of as a "circular buffer".

The kernel ring buffer stores information such as the initialization messages of device drivers, messages from hardware, and messages from kernel modules. Because it contains these low-level startup messages, the ring buffer is a good place to start an investigation into hardware errors or other startup issues.

I log non vengono cancellati di default, siamo noi ad impostare uno spazio massimo di memoria dedicata ai log.

COMANDO:

Journalctl –b -> comando persistente, mi permette di vedere i log della fase di boot B.
Journalctl –list –boots -> mi mostra il numero di boot
Journalctl –since "1 hour ago" until "1 hour ago"
Journalctl –since "anno-mese-giorno"

Dmesg -> comando non persistente, posso vedere solamente i log dalla fase di boot corrente in poi. Quando il buffer si riempie i nuovi log prenderanno il posto di quelli meno recenti (Archiviazione Circolare).

Systemctl vs Journalctl:

Con systemctl vedo lo stato dei servizi, mentre con journatlctl vedo i log dei servizi.

Journalctl _UID=1000 -> filtro la ricerca per l'applicativo (con l'ID=1000) e non per l'utente.
Journalctl –unit=bluetooth.service -f(follow) -> vedo i log in tempo reale di quel servizio
Journalctl –k -> restituisce i log del kernel

PARTIZIONAMENTO DISCO

lunedì 9 ottobre 2023 07:44

Partizionamento di un disco:

Sudo parted –l -> mostra come è stato partizionato il disco
Sudo fdisk -> idem

FS + SWAP

Sudo parted /dev/sdb
Specifico l'inizio e la fine della partizione (in M oppure in %)
Creo oltre ad un partizionamento col FS, uno che contenga la memoria Swap
Mkpart
Mkfs –t ext4 dev/sdb1 -> all'interno di questa partizione verrà inserito il FS (deve essere montato all'interno di una cartella vuota, altrimenti finché non verrà smontato, il contenuto di quella cartella verrà mascherato) -> per questo utilizziamo una cartella di montaggio di default (/mnt/)
Sudo mount /dev/sdb1 /mnt/
Mkswap /dev/sdb2 -> la seconda partizione la utilizzo per la memoria swap
Sudo swapon /dev/sdb2 -> montaggio della partizione di swap
Swapon: comando per vedere le aree di swap disponibili

Cos'è la memoria Swap?
La memoria Swap serve nel caso la memoria RAM sia in fase di saturazione, andando per l'appunto a swappare memoria, evitando così che si vada ad utilizzare tutta la memoria RAM.

Posso regolare la soglia massima di occupazione della RAM, oltre la quale i dati cominciano ad essere scritti sull'area di swap, regolando il valore in /proc/sys/vm/swappiness (se inserisco 30, al 70% di occupazione della RAM swappo), ma le modifiche rimangono solamente per la sessione corrente, perché è solamente un'interfaccia del kernel per mostrarci i valori di configurazione.
Swappando memoria notiamo che la velocità di scrittura è molto ridotta, essendo che stiamo scrivendo su un hdd.

Può essere utile anche quando riscontriamo dei memory leak; ovvero la RAM, una volta svolto il compito per il quale della memoria era stata allocata , non viene deallocata e, quindi, lo spazio non viene liberato.
Sudo vim /etc/sysctl.conf posso modificarlo, rendendo definitive le modifiche.

COMANDI PER PARTIZIONARE UN DISCO:

LOGICAL VOLUME MANAGER

lunedì 9 ottobre 2023 11:13

Sistema per andare oltre il solito sistema di ripartizionamento, per trattare i dischi come fossero un disco solo.

Nel caso non si utilizzi un LVM, una volta esaurita la memoria va sostituita, però è un metodo antiquato.
Utilizzando l'LVM possiamo avere più volumi fisici inseriti, per estendere i Logical Volume (/dev/sda, /dev/sdb).
Volume Group
Logical Volume (corrispettivo delle partizioni), che vengono montati in /dev/nomeVG/nomePartizione

RESIZE2FS:
Utilizzando questo metodo la grandezza dei logical volume possono essere modificati, aggiungendo o togliendo spazio.
Se voglio aggiungere spazio, devo prima aumentare lo spazio e poi il File System; mentre se voglio togliere spazio prima diminuisco il File System e poi la dimensione della partizione.

COMANDI:
Pvscan: identificare i physical volume
Vgscan: identificare i volume group
Lvscan: identificare i logical volume

LVM is a mechanism that provides an alternative method of managing storage systems than the traditional partition-based one.

Partitions = Logical Volumes
Disks = Volume Groups

+: Easy resize a logical volume/volume group (without the partition part)

PHYSICAL VOLUME: the raw materials/building blocksthat are used to achieve the abstraction that is logical volumes.
A physical volume can be a partition.

LIBRERIE CONDIVISE

lunedì 9 ottobre 2023 12:10

Le librerie sono dei pezzi di codice che dei software utilizzano per funzionare.
In Windows vengono chiamati file.dll, mentre con linux si chiamano file.so.
Per vedere le librerie che un software sta utilizzando si verificano con il comando ldd /path/to/the/software.
Per funzionare correttamente, tutte le librerie non devono essere compromesse.
Posso controllare le librerie che un software utilizza anche tramite ldd /usr/bin/software | less.

LIBRERIE LINKATE STATICAMENTE
Un programma contiene internamente le librerie di cui ha bisogno, risultando più solido, non dovendo richiamare nessuna libreria esterna, ma anche più pesante.

LIBRERIE LINKATE DINAMICAMENTE
Ogni volta che un software viene avviato richiama le librerie di cui ha bisogno, risultando più leggero.

In programming, a library is an assortment of pre-compiled pieces of code that can be reused in a program.

Linux supports two classes of libraries:

- Static libraries: are bound to a program statically at compile time
- Dynamic/Shared libraries: loaded when a program is launched and loaded into memory and binding occurs at run time.

Dynamic or shared libraries can further be categorized into:

- Dynamically linked libraries: here a program is linked with the shared library and the kernel loads upon execution.
- Dybamically loaded libraries: the program takes full control by calling functions with the library.

APT PACKAGE MANAGER

mercoledì 11 ottobre 2023 08:03

APT: Advanced Package Manage, is respondible for downloading, installing, updating and removing packages and their dependencies from yor Debian based system.

Packages: archive files that contain multiple deb files that are used by the dpkg to install programs.

Linux System --> /etc/apt/sources.list or /etc/apt/sources.list.d
Packages are downloaded by trusted sources and stored in a local cache.

APT CACHE: internal database that contain all the packages that were downloaded during the sudo apt update phase.
The APT cache is used to provide offline information about current packages installed on your system.
It guarantees that you are able to access packages information without having to be connected to internet.

SEARCH PACKAGE IN THE CACHE: apt-cache search python

SHOW PACKAGE INFO: apt-cache show gcc

SEARCH AN INSTALLED PACKAGE: whereis gcc
By default apt packages are installed into /etc/apt/.

COMANDI:
Apt update: aggiorna la lista dei pacchetti disponibili nella cache
Apt upgrade: installa i pacchetti disponibili ed esegue l'aggiornamento
Apt full-upgrade: esegue l'aggiornamento completo, eliminando anche i pacchetti non necessari
Apt install: installazione di pacchetti
Apt remove: rimozione di pacchetti
L'installatore di pacchetti di APT è DPKG.

DNF PACKAGE MANAGER

mercoledì 11 ottobre 2023 08:21

Dnf è il PM per i sistemi Red-Hat.
Il suo installatore di pacchetti è rpm.

Dnf update: esegue l'aggiornamento dei pacchetti con la rimozione dei pacchetti non necessari (l'equivalente di apt full-upgrade).
Dnf install: aggiorna la lista dei pacchetti disponibili e installa il pacchetto richiesto (equivalente ad apt update && update install)

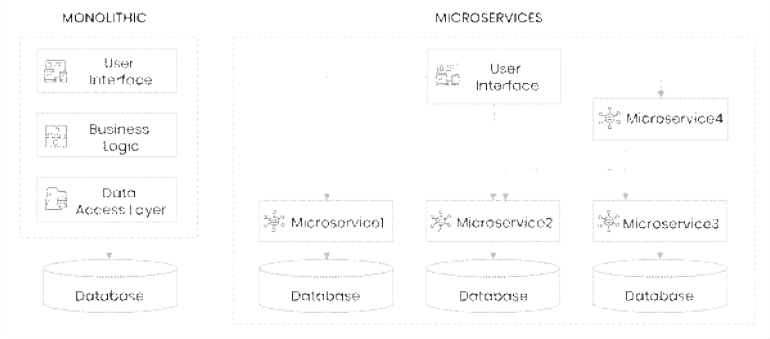
RPM packages will not install unless the requisites are installed forst so YUM performed the dependency analysis and installed pre-requisites before installing the package.

CONTAINER vs VIRTUALIZATION

Saturday, October 14, 2023 10:36 PM

Containerization is a form of virtualization. Virtualization aims to run multiple OS instances on a single server, whereas containerization runs a single OS instance, with multiple user spaces to isolate processes from one another. This means containerization makes sense for one AWS cloud user that plans to run multiple processes simultaneously.

Containerization is achieved by packaging software code, libraries, frameworks, and other dependencies together in an isolated user space called a container. This container is portable and can be used on any infrastructure in any environment that supports the container technology, such as Docker and Kubernetes.



1. Isolation
Virtualization results in a fully isolated OS and VM instance, while containerization isolates the host operating system machine and containers from one another. However, all containers are at risk if an attacker controls the host.

2. Different Operating Systems
Virtualization can host more than one complete operating system, each with its own kernel, whereas containerization runs all containers via user mode on one OS.

3. Guest Support
Virtualization allows for a range of operating systems to be used on the same server or machine. On the other hand, containerization is reliant on the host OS, meaning Linux containers cannot be run on Windows and vice-versa.

4. Deployment
Virtualization means each virtual machine has its own hypervisor. With containerization, either Docker is used to deploy an individual container, or Kubernetes is used to orchestrate multiple containers across multiple systems.

5. Persistent Virtual Storage
Virtualization assigns a virtual hard disk (VHD) to each individual virtual machine, or a server message block (SMB) if shared storage is used across multiple servers. With containerization, the local hard disk is used for storage per node, with SMB for shared storage across multiple nodes.

6. Virtual Load Balancing
Virtualization means failover clusters are used to run VMs with load balancing support. Since containerization uses orchestration via Docker or Kubernetes to start and stop containers, it maximizes resource utilization. However, decommissioning for load balancing with containerization occurs when limits on available resources are reached.

7. Virtualized Networking
Virtualization uses virtual network adaptors (VNA) to facilitate networking, running through a master network interface card (NIC). With containerization, the VNA is split into multiple isolated views for lightweight network virtualization.

What Are the Benefits of Virtualization?

Virtualization can increase application scalability while simultaneously reducing expenses. Here are five more ways virtualization can help your business:

- More efficient resource utilization via multi-tenant support on hardware.
- High availability by spooling a virtualized resource immediately and decommission once processes complete.
- Greater business continuity with easy virtual instance recovery via duplication and backups.
- Virtual machines can be quickly deployed, as the underlying OS and dependencies are already loaded on the hypervisor.
- Cloud portability is enhanced thanks to virtualization, leading to easier multi-cloud migrations.

What Are the Disadvantages of Virtualization?

While virtualization does offer the ability to run multiple applications on a single physical server, it can also hinder performance. Here are six more considerations when deciding if virtualization is right for your business:

- The return on investment (ROI) with virtualization can take years, meaning higher upfront costs but lower overall day-to-day costs.
- Public cloud virtual instances can have a risk of data loss or breach, due to multi-tenant infrastructure and the possibility of data or kernel leaks to other users.
- Scaling can take a long time for multiple virtualized instances, where velocity is key.
- Hypervisor technologies always come with a performance overhead, meaning less performance with an equal number of resources.
- Virtual servers containing virtualized instances can sprawl endlessly, creating additional management burdens for the IT department if not monitored.

What Are the Benefits of Containerization?

The platform-agnostic nature of containerization makes it an appealing solution for scaling cloud-based applications. Here are three more benefits to help you decide if containerization is right for you:

- Containers are lightweight and fast to deploy. Compared to virtualization, where each instance may be gigabytes (GB) in size, containers can be mere megabytes (MB) in size.
- Thanks to dependencies, libraries, binaries, and configuration files being bundled together, containers can be redeployed as needed to any platform or environment.
- The lightweight nature of containers can lead to meaningful operational and developmental cost reductions.

What Are the Disadvantages of Containerization?

While containerization offers scalability and agility when modernizing applications in the cloud, it also has serval drawbacks. Here are five disadvantages of containerization:

- Containerization is well-supported on Linux-based distributions, but Windows support is not truly adequate for enterprise use. This limits users to Linux in most use cases.
- Kernel vulnerabilities mean every container in a K8S cluster can be compromised, not just an isolated few.
- Networking is difficult as each container is running on a single server. This would require a network bridge or a macvlan driver (combination of MAC addresses and virtual local area network) to map container network interfaces to host interfaces.
- Monitoring hundreds of containers containing individual processes is more difficult than monitoring multiple processes on a single virtual machine instance.
- Containerization does not always benefit workloads and can sometimes result in worse performance.

COM'è COMPOSTO UN AMBIENTE DOCKERIZZATO -> + leggera, memoria dinamica

App
Binari/librerie
Container/daemon
Host OS kernel
HW

COM'è COMPOSTO UN AMBIENTE VIRTUALIZZATO -> l'hw deve sostenere la virtualizzazione, l'isolamento è migliore

App
Binari
Guest OS kernel
Hypervisor
Host OS kernel
HW

TTY vs PTS

Saturday, October 14, 2023 10:45 PM

TTY: teletypewriter is a text I/O environment