UNIT 5

Virtual Machine abstracts the hardware of our personal computer such as CPU, disk drives, memory, NIC (Network Interface Card) etc, into many different execution environments as per our requirements, hence giving us a feel that each execution environment is a single computer. For example, VirtualBox.

When we run different processes on an operating system, it creates an illusion that each process is running on a different processor having its own virtual memory, with the help of CPU scheduling and virtual-memory techniques. There are additional features of a process that cannot be provided by the hardware alone like system calls and a file system. The virtual machine approach does not provide these additional functionalities but it only provides an interface that is same as basic hardware. Each process is provided with a virtual copy of the underlying computer system.

We can create a virtual machine for several reasons, all of which are fundamentally related to the ability to share the same basic hardware yet can also support different execution environments, i.e., different operating systems simultaneously.

The main drawback with the virtual-machine approach involves disk systems. Let us suppose that the physical machine has only three disk drives but wants to support seven virtual machines. Obviously, it cannot allocate a disk drive to each virtual machine, because virtual-machine software itself will need substantial disk space to provide virtual memory and spooling. The solution is to provide virtual disks.

Users are thus given their own virtual machines. After which they can run any of the operating systems or software packages that are available on the underlying machine. The virtual-machine software is concerned with multi-programming multiple virtual machines onto a physical machine, but it does not need to consider any user-support software. This arrangement can provide a useful way to divide the problem of designing a multi-user interactive system, into two smaller pieces.

Advantages:

1. There are no protection problems because each virtual machine is completely isolated from all other virtual machines.
2. Virtual machine can provide an instruction set architecture that differs from real computers.
3. Easy maintenance, availability and convenient recovery.

Disadvantages:

1. When multiple virtual machines are simultaneously running on a host computer, one virtual machine can be affected by other running virtual machines, depending on the workload.
2. Virtual machines are not as efficient as a real one when accessing the hardware.

HISTORY

Both system virtual machines and process virtual machines date to the 1960s and continue to be areas of active development.

System virtual machines grew out of time-sharing, as notably implemented in the Compatible Time-Sharing System (CTSS). Time-sharing allowed multiple users to use a computer concurrently: each program appeared to have full access to the machine, but only one program was executed at the time, with the system switching between programs in time slices, saving and restoring state each time. This evolved into virtual machines, notably via IBM's research systems: the M44/44X, which used partial virtualization, and the CP-40 and SIMMON, which used full virtualization, and were early examples of hypervisors. The first widely available virtual machine architecture was the CP-67/CMS (see History of CP/CMS for details). An important distinction was between using multiple virtual machines on one host system for time-sharing, as in M44/44X and CP-40, and using one virtual machine on a host system for prototyping, as in SIMMON. Emulators, with hardware emulation of earlier systems for compatibility, date back to the IBM System/360 in 1963,[6][7] while the software emulation (then-called "simulation") predates it.

Process virtual machines arose originally as abstract platforms for an intermediate language used as the intermediate representation of a program by a compiler; early examples date to around 1966. An early 1966 example was the O-code machine, a virtual machine that executes O-code (object code) emitted by the front end of the BCPL compiler. This abstraction allowed the compiler to be easily ported to a new architecture by implementing a new back end that took the existing O-code and compiled it to machine code for the underlying physical machine. The Euler language used a similar design, with the intermediate language named *P* (portable).[8] This was popularized around 1970 by Pascal, notably in the Pascal-P system (1973) and Pascal-S compiler (1975), in which it was termed p-code and the resulting machine as a p-code machine.

This has been influential, and virtual machines in this sense have been often generally called p-code machines. In addition to being an intermediate language, Pascal p-code was also executed directly by an interpreter implementing the virtual machine, notably in UCSD Pascal (1978); this influenced later interpreters, notably the Java virtual machine (JVM). Another early example was SNOBOL4 (1967), which was written in the SNOBOL Implementation Language (SIL), an assembly language for a virtual machine, which was then targeted to physical machines by transpiling to their native assembler via a macro assembler.[9] Macros have since fallen out of favor, however, so this approach has been less influential. Process virtual machines were a popular approach to implementing early microcomputer software, including Tiny BASIC and adventure games, from one-off implementations such as Pyramid 2000 to a general-purpose engine like Infocom's z-machine, which Graham Nelson argues is "possibly the most portable virtual machine ever created".

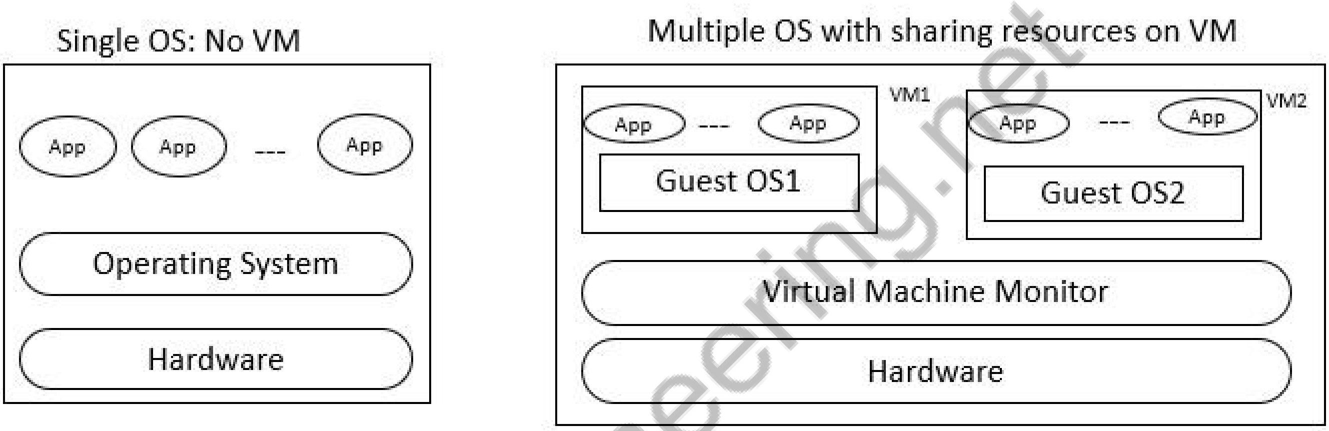
Significant advances occurred in the implementation of Smalltalk-80, particularly the Deutsch/Schiffmann implementation which pushed just-in-time (JIT) compilation forward as an implementation approach that uses process virtual machine. Later notable Smalltalk VMs were VisualWorks, the Squeak Virtual Machine, and Strongtalk. A related language that produced a lot of virtual machine innovation was the Self programming language, which pioneered adaptive optimization[17] and generational garbage collection. These techniques proved commercially successful in 1999 in the HotSpot Java virtual machine.[18] Other innovations include having a register-based virtual machine, to better match the underlying hardware, rather than a stack-based virtual machine, which is a closer match for the programming language; in 1995, this was pioneered by the Dis virtual machine for the Limbo language. OpenJ9 is an alternative for HotSpot JVM in OpenJDK and is an open source eclipse project claiming better startup and less resource consumption compared to HotSpot.

FEATURES OF VIRTUAL MACHINES

The features of the virtual machines are as follows −

- Multiple OS systems use the same hardware and partition resources between virtual computers.
- Separate Security and configuration identity.
- Ability to move the virtual computers between the physical host computers as holistically integrated files.

The below diagram shows you the difference between the single OS with no VM and Multiple OS with VM −



BENEFITS

Let us see the major benefits of virtual machines for operating-system designers and users which are as follows −

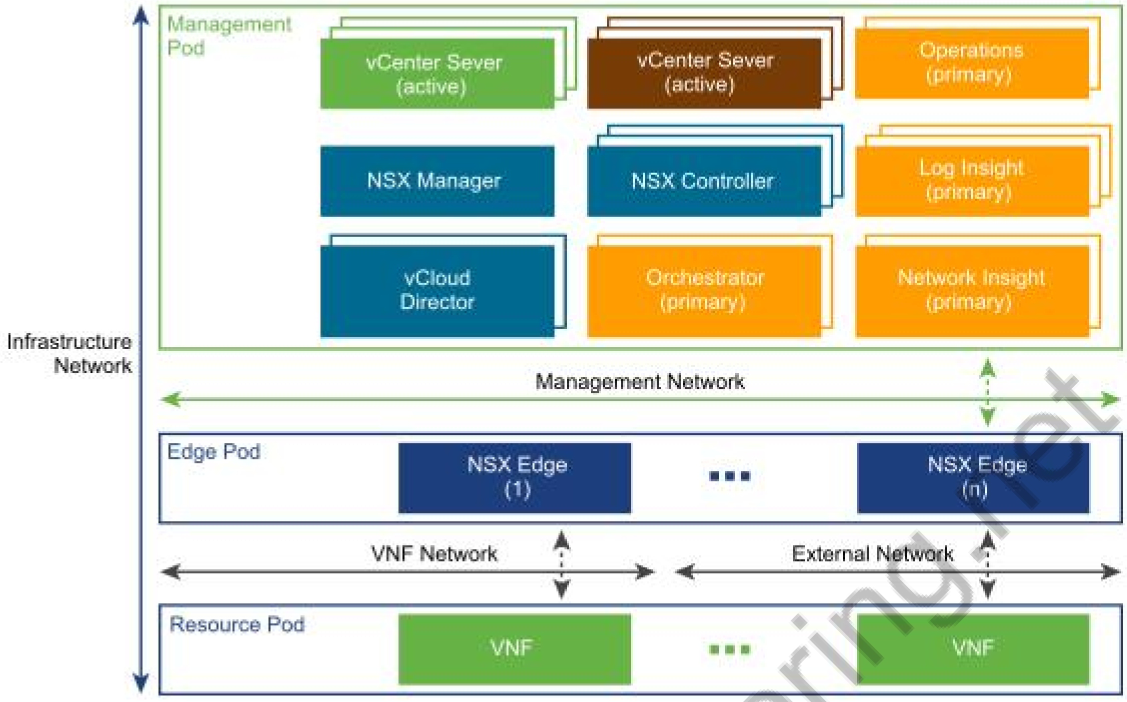
- The multiple Operating system environments exist simultaneously on the same machine, which is isolated from each other.
- Virtual machine offers an instruction set architecture which differs from real computer.
- Using virtual machines, there is easy maintenance, application provisioning, availability and convenient recovery.

Virtual Machine encourages the users to go beyond the limitations of hardware to achieve their goals.

The operating system achieves virtualization with the help of a specialized software called a hypervisor, which emulates the PC client or server CPU, memory, hard disk, network and other hardware resources completely, enabling virtual machines to share resources.

The hypervisor can emulate multiple virtual hardware platforms that are isolated from each other allowing virtual machines to run Linux and window server operating machines on the same underlying physical host.

VIRTUAL BUILDING BLOCKS



Storage Design

This reference architecture uses a shared storage design that is based on vSAN. vCloud NFV also supports certified third-party shared storage solutions, as listed in the VMware Compatibility Guide.

vSAN is a software feature built in the ESXi hypervisor that allows locally attached storage to be pooled and presented as a shared storage pool for all hosts in a vSphere cluster. This simplifies the storage configuration with a single datastore per cluster for management and VNF workloads. With vSAN, VM data is stored as objects and components. One object consists of multiple components, which are distributed across the vSAN cluster based on the policy that is assigned to the object. The policy for the object ensures a highly available storage backend for the cluster workload, with no single point of failure.

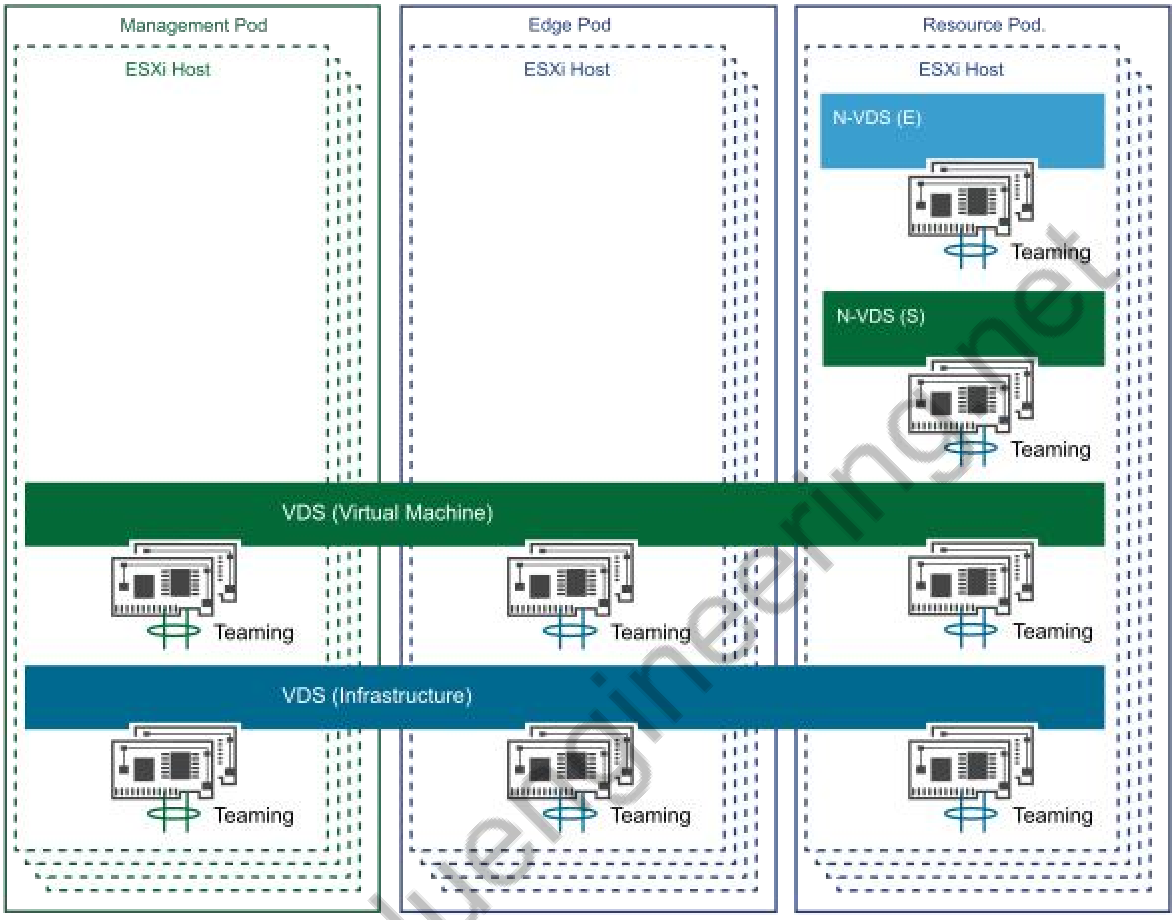
vSAN is a fully integrated hyperconverged storage software. Creating a cluster of server hard disk drives (HDDs) and solid-state drives (SSDs), vSAN presents a flash-optimized, highly resilient, shared storage datastore to ESXi hosts and virtual machines. This allows for the control of capacity, performance, and availability through storage policies, on a per VM basis.

Network Design

Thev Cloud NFV platform consists of infrastructure networks and VM networks. Infrastructure networks are host level networks that connect hypervisors to physical networks. Each ESXi host has multiple port groups configured for each infrastructure network.

The hosts in each Pod are configured with VMware vSphere® Distributed Switch™ (VDS) devices that provide consistent network configuration across multiple hosts. One vSphere Distributed Switch is used for VM networks and the other one maintains the infrastructure networks. Also, the N-VDS switch is used as the transport for telco workload traffic.

Virtual Network Design



Infrastructure networks are used by the ESXi hypervisor for vMotion, VMware vSphere Replication, vSAN traffic, and management and backup. The Virtual Machine networks are used by VMs to communicate with each other. For each Pod, the separation between infrastructure and VM networks ensures security and provides network resources where needed. This separation is implemented by two vSphere Distributed Switches, one for infrastructure networks and another one for VM networks. Each distributed switch has separate uplink connectivity to the physical data center network, completely separating its traffic from other network traffic. The uplinks are mapped to a pair of physical NICs on each ESXi host, for optimal performance and resiliency.

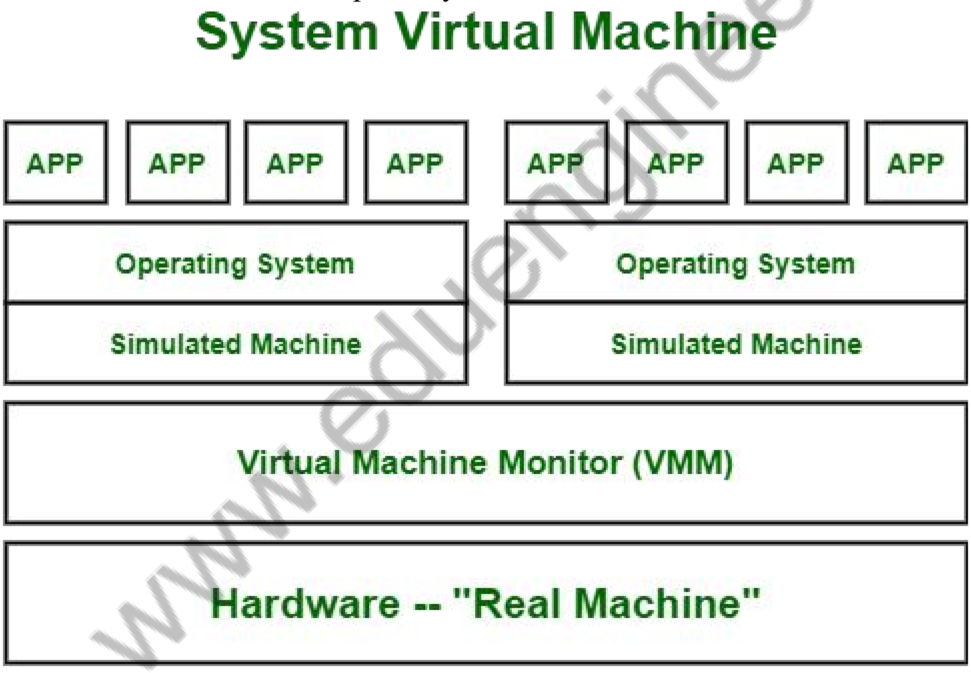
VMs can be connected to each other over a VLAN or over Geneve-based overlay tunnels. Both networks are designed according to the requirements of the workloads that are hosted by a specific Pod. The infrastructure vSphere Distributed Switch and networks remain the same regardless of the Pod function. However, the VM networks depend on the networks that the specific Pod requires. The VM networks are created by NSX-T Data Center to provide enhanced networking services and performance to the Pod workloads. The ESXi host's physical NICs are used as uplinks to connect the distributed switches to the physical network switches. All ESXi physical NICs connect to layer 2 or layer 3 managed switches on the physical network. It is common to use two switches for connecting to the host physical NICs for redundancy purposes.

TYPES OF VIRTUAL MACHINES

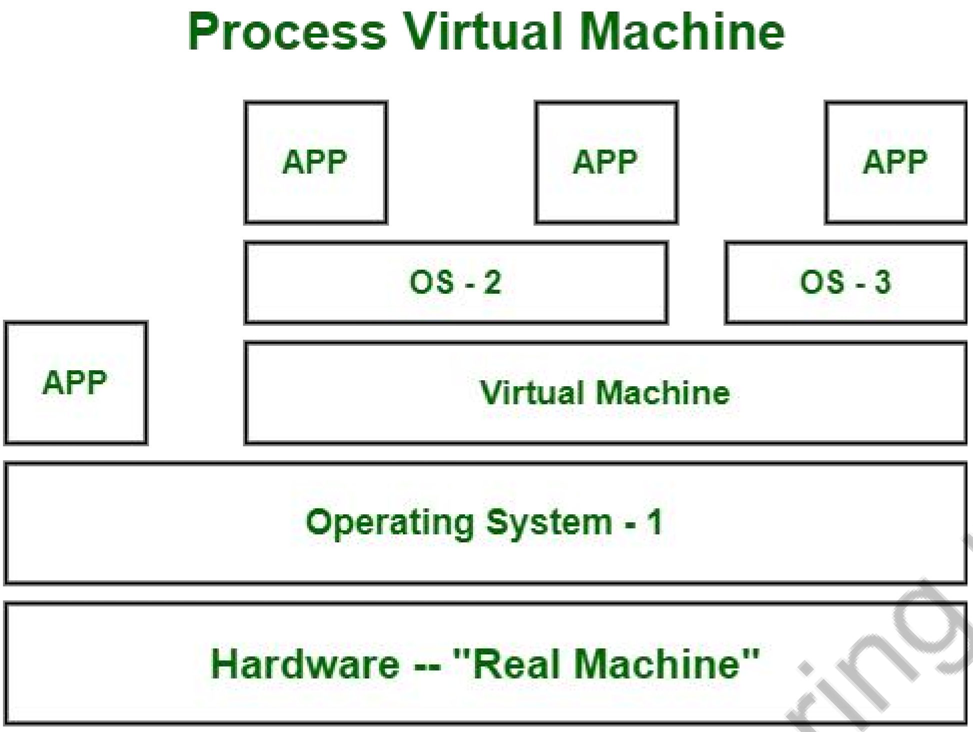
Virtual Machine is like fake computer system operating on your hardware. It partially uses the hardware of your system (like CPU, RAM, disk space, etc.) but its space is completely separated from your main system. Two virtual machines don't interrupt in each other's working and functioning nor they can access each other's space which gives an illusion that we are using totally different hardware system. More detail at Virtual Machine.

Types of Virtual Machines : Virtual machines are classified into two types:

1. **System Virtual Machine:** These types of virtual machines gives us complete system platform and gives the execution of the complete virtual operating system. Just like virtual box, system virtual machine is providing an environment for an OS to be installed completely. We can see in below image that our hardware of Real Machine is being distributed between two simulated operating systems by Virtual machine monitor. And then some programs, processes are going on in that distributed hardware of simulated machines separately.



2. Process Virtual Machine : While process virtual machines, unlike system virtual machine, does not provide us with the facility to install the virtual operating system completely. Rather it creates virtual environment of that OS while using some app or program and this environment will be destroyed as soon as we exit from that app. Like in below image, there are some apps running on main OS as well some virtual machines are created to run other apps. This shows that as those programs required different OS, process virtual machine provided them with that for the time being those programs are running. **Example** – Wine software in Linux helps to run Windows applications.



Virtual Machine Language : It's type of language which can be understood by different operating systems. It is platform-independent. Just like to run any programming language (C, python, or java) we need

Types of VMs – Type 0 Hypervisor

Old idea, under many names by HW manufacturers

- "partitions","domains"
- A HW feature implemented by fimware
- OS need to nothing special, VMM is in fimware
- Smaller feature set than other types
- Each guest has dedicated HW
- I/O a challenge as difficul to have enough devices, controlers to dedicate to each guest
- Sometimes VMM implements a **control partition** running daemons that other guests communicate with for shared I/O
- Can provide virtualization-within-virtualization(guest itself can be a VMM with guests
- Other types have difficulty in doing this.

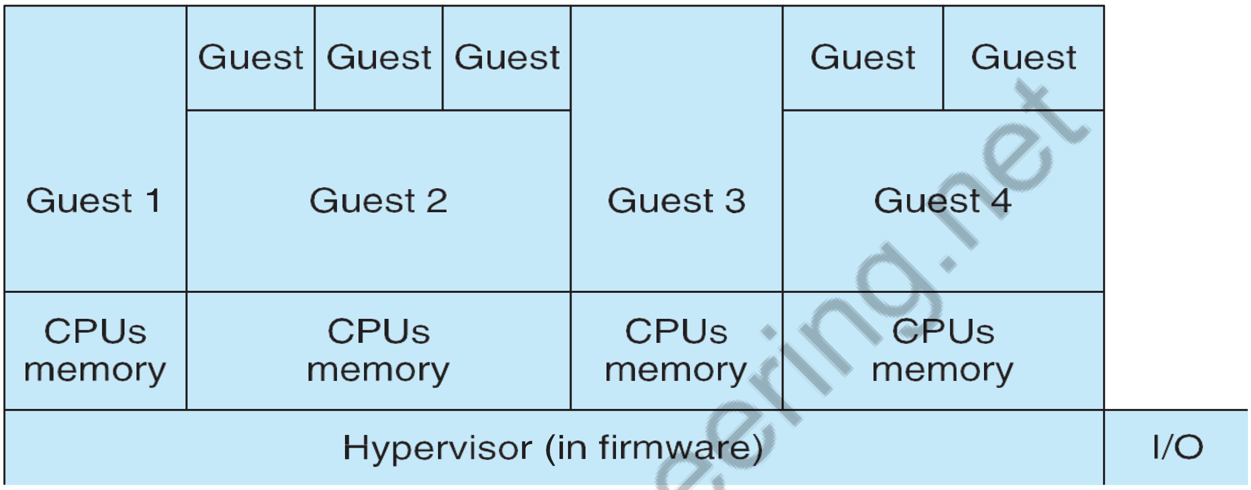
A **virtual machine (VM)** is a virtual environment which functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system.

Types of VMs – Type 1 Hypervisor

Commonly found in company data centers

- ▪ Special purpose operating systems that run natively on HW

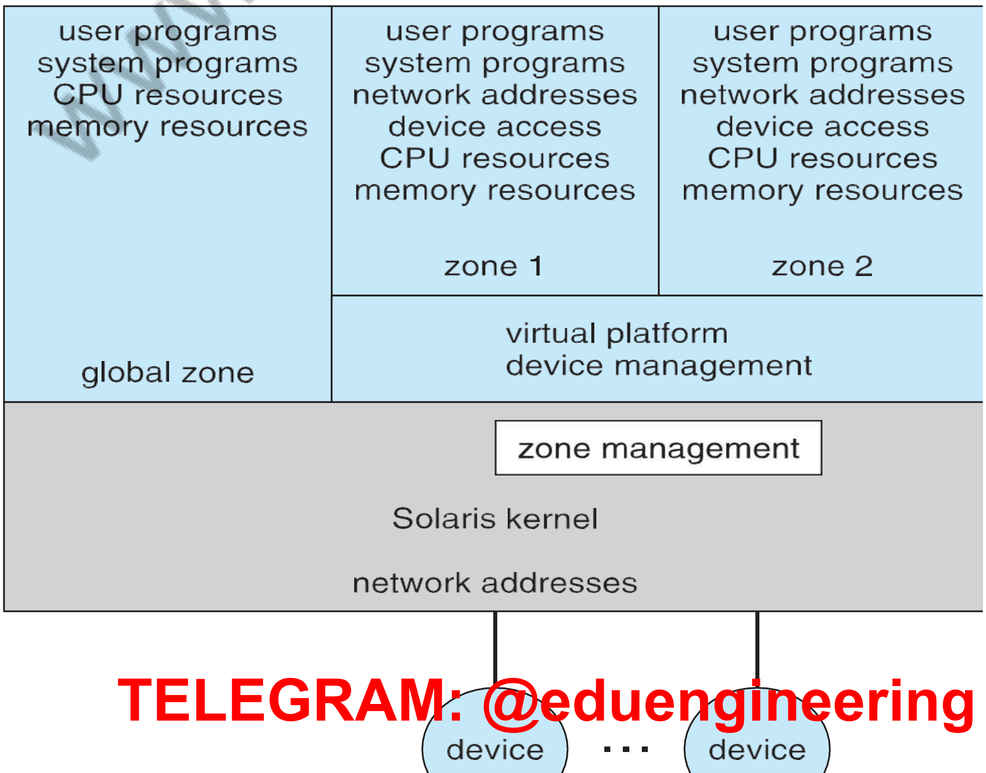
- Rather than providing system call interface, creater unand manage guest OSes.
- Can run on Type0 hypervisors but not on other Type1s
- Run in kernel mode
- Guests generally don't know they are running in a VM
- Implement device drivers for host HW because no other component can
- Also provide other traditional OS services like CPU and memory management



- Very little OS involvement in virtualization
- VMM is simply another process, run and managed by host
- ◗ Even the host doesn't know they are a VMM running guests
  - ◗ Tend to have poorer overall performance because can take advantage of some HW features
  - ◗ But also a benefit because require no changes to host OS

    - ◗ Student could have Type2 hypervisor on native host, run
    - ◗ Multiple guests, all on standard host OS such as Windows, Linux, MacOS

Solaris 10 with Two Zones



VIRTUALIZATION AND OPERATING-SYSTEM COMPONENTS

Now look at operating system aspects of virtualization

- CPU scheduling, memory management, I/O, storage, and unique VM migration feature

- How do VMM sschedule CPU use when guests believe they have dedicated CPUs?

  - How can memory management work when many guests

Require large amounts of memory?

OS Component – CPU Scheduling

Even single-CPU systems act like multiprocessor ones when virtualized

- One or more virtual CPUs per guest

- Generally VMM has one or more physical CPUs and number of threads to run on them.
- Guests configured with certain number of VCPUs
  - Can be adjusted throughout life of VM

When enough CPUs for all guests->VMM can allocate dedicated CPUs, each guest much like native operating system managing its CPUs

Usually not enough CPUs->CPU **over commitment**

VMM can use standard scheduling algorithms to put threads on CPUs

Some add fairness aspect

Cycle stealing by VMM and oversubscription of CPUs means guests don't get CPU cycles they expect.

- Consider timesharing scheduler in a guest trying to schedule 100ms time slices -> each may take 100ms, 1 second, or longer
- Poor response times for users of guest
- Time-of-day clocks incorrect
- Some VMMs provide application to run in each guest to fix time-of-day and provide other integration features

OS Component – Memory Management

Also suffers from over subscription -> requires extra management efficiency from VMM

For example, VMware ESX guests have a configured amount of physical memory, then ESX uses 3 methods of memory management

1. Double-paging, in which the guest page table indicates a page is in a physical frame but the VMM moves some of those pages to backing store

2. Install a pseudo-device driver in each guest (it looks like a device driver to the guest kernel but really just adds kernel-mode code to the guest)

- Balloon memory manager communicates with VMM and is told to allocate or deallocate memory to decrease or increase physical memory use of guest, causing guest OS to free or have more memory available

4. Deduplication by VMM determining if same page loaded more than once, memory mapping the same page into multiple guests

OS Component – I/O

Easier for VMMs to integrate with guests because I/O has lots of variation

Already somewhat segregated / flexible via device drivers

VMM can provide new devices and device drivers

But overall I/O is complicated for VMMs

- Many short paths for I/O in standard OSes for improved performance
- Less hypervisor needs to do for I/O for guests, the better
- Possibilities include direct device access, DMA pass-through, direct interrupt delivery

○ Again, HW support needed for these

Networking also complex as VMM and guests all need network access

- VMM can bridge guest to network (allowing direct access)
- And / or provide network address translation (NAT)
- NAT address local to machine on which guest is running, VMM provides address translation to guest to hide its address.

OS Component – Storage Management

- Both boot disk and general data access need be provided by VMM
- Need to support potentially dozens of guests per VMM (so standard disk partitioning not sufficient)
- Type 1 – storage guest root disks and config information within file system provided by VMM as a disk image
- Type 2 – store as files in file system provided by host OS
- Duplicate file -> create new guest
- Move file to another system -> move guest
- **Physical-to-virtual (P-to-V)** convert native disk blocks into VMM format
- **Virtual-to-physical (V-to-P)** convert from virtual format to native or disk format

VMM also needs to provide access to network attached storage (just networking) and other disk images, disk partitions, disks, etc.

OS Component – Live Migration

Taking advantage of VMM features leads to new functionality not found on general operating systems such as live migration

Running guest can be moved between systems, without interrupting user access to the guest or its apps

Very useful for resource management, maintenance downtime windows, etc

1. The source VMM establishes a connection with the target VMM

2. The target creates a new guest by creating a new VCPU, etc

3. The source sends all read-only guest memory pages to the target

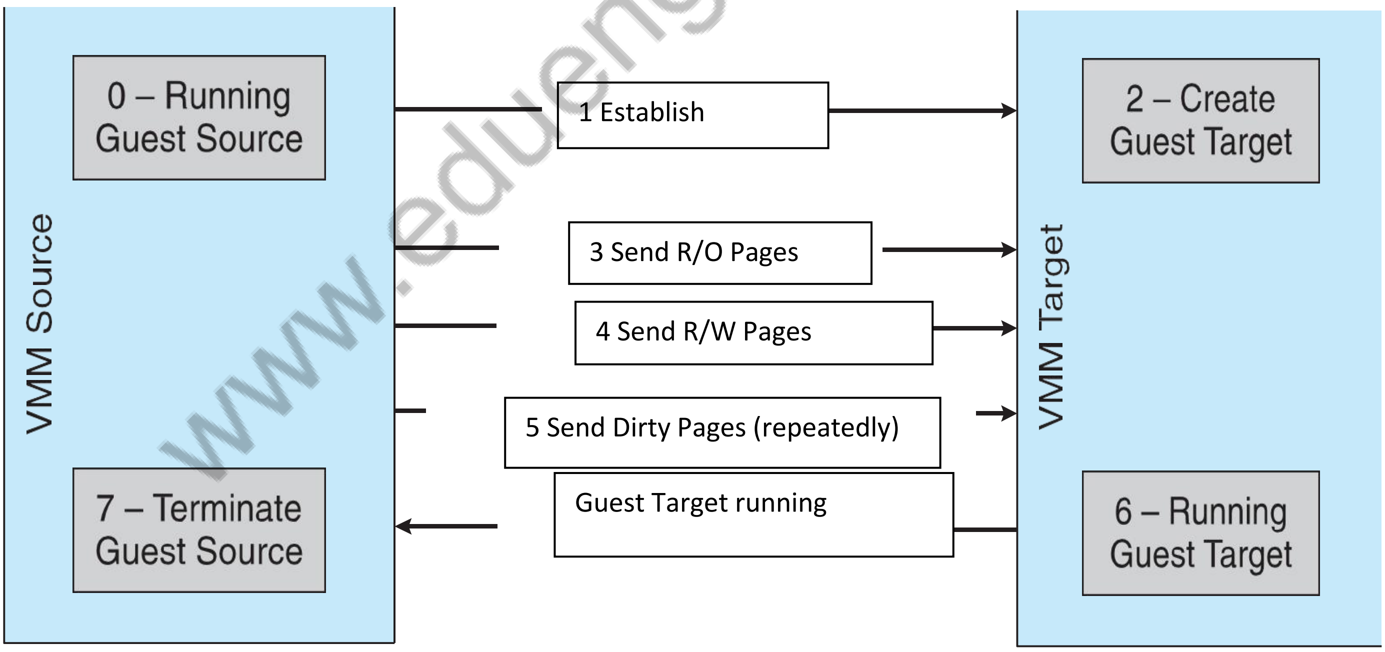
4. The source sends all read-write pages to the target, marking them as clean

5. The source repeats step 4, as during that step some pages were probably modified by the guest and are now dirty

6. When cycle of steps 4 and 5 becomes very short, source VMM freezes guest, sends VCPU's final state, sends other state details, sends final dirty pages, and tells target to start running the guest

Once target acknowledges that guest running, source terminates guest.

Live Migration of Guest Between Servers



BASIS FOR DEVELOPING THE OS

Create the illusion of having one or more objects to emulate the real object. It is closely related to abstraction. In developing the OS, abstraction provides simplification by combining multiple simple objects into a single complex object

Virtualization provides diversification and replication by creating the illusion of objects with desired characteristics.

The virtual infrastructure design comprises the design of the software components that form the virtual infrastructure layer. This layer supports running telco workloads and workloads that maintain the business continuity of services. The virtual infrastructure components include the virtualization platform hypervisor, virtualization management, storage virtualization, network virtualization, and backup and disaster recovery components.

This section outlines the building blocks for the virtual infrastructure, their components, and the networking to tie all the components together.

MOBILE OPERATING SYSTEM

A mobile operating system is an operating system that helps to run other application software on mobile devices. It is the same kind of software as the famous computer operating systems like Linux and Windows, but now they are light and simple to some extent.

The operating systems found on smartphones include Symbian OS, iPhone OS, RIM's BlackBerry, Windows Mobile, Palm WebOS, Android, and Maemo. Android, WebOS, and Maemo are all derived from Linux. The iPhone OS originated from BSD and NeXTSTEP, which are related to Unix.

It combines the beauty of computer and hand use devices. It typically contains a cellular built-in modem and SIM tray for telephony and internet connections. If you buy a mobile, the manufacturer company chooses the OS for that specific device.

Popular platforms of the Mobile OS

1. Android OS: The Android operating system is the most popular operating system today. It is a mobile OS based on the **Linux Kernel** and **open-source software**. The android operating system was developed by **Google**. The first Android device was launched in **2008**.

2. Bada (Samsung Electronics): Bada is a Samsung mobile operating system that was launched in 2010. The Samsung wave was the first mobile to use the bada operating system. The bada operating system offers many mobile features, such as 3-D graphics, application installation, and multipoint-touch.

3. BlackBerry OS: The BlackBerry operating system is a mobile operating system developed by **Research In Motion** (RIM). This operating system was designed specifically for BlackBerry handheld devices. This operating system is beneficial for the corporate users because it provides synchronization with Microsoft Exchange, Novell GroupWise email, Lotus Domino, and other business software when used with the BlackBerry Enterprise Server.

4. iPhone OS / iOS: The iOS was developed by the Apple inc for the use on its device. The iOS operating system is the most popular operating system today. It is a very secure operating system. The iOS operating system is not available for any other mobiles.

5. Symbian OS: Symbian operating system is a mobile operating system that provides a high-level of integration with communication. The Symbian operating system is based on the java language. It combines middleware of wireless communications and personal information management (PIM) functionality. The Symbian operating system was developed by **Symbian Ltd** in **1998** for the use of mobile phones. **Nokia** was the first company to release Symbian OS on its mobile phone at that time.

6. Windows Mobile OS: The window mobile OS is a mobile operating system that was developed by **Microsoft**. It was designed for the pocket PCs and smart mobiles.

7. Harmony OS: The harmony operating system is the latest mobile operating system that was developed by Huawei for the use of its devices. It is designed primarily for IoT devices.

8. Palm OS: The palm operating system is a mobile operating system that was developed by **Palm Ltd** for use on personal digital assistants (PADs). It was introduced in **1996**. Palm OS is also known as the **Garnet OS**.

9. WebOS (Palm/HP): The WebOS is a mobile operating system that was developed by **Palm**. It based on the **Linux Kernel**. The HP uses this operating system in its mobile and touchpads.

What is Apple iOS?

Apple iOS is a proprietary mobile operating system that runs on mobile devices such as the iPhone and iPad. Apple iOS stands for iPhone operating system and is designed for use with Apple's multitouch devices. The mobile OS supports input through direct manipulation and responds to various user gestures, such as pinching, tapping and swiping. The iOS developer kit provides tools that allow for iOS app development.

Apple iOS market share

As of 2022, the Apple iOS market share was 18.8% worldwide, making it the second most popular brand behind Samsung, according to IDC.

Apple iOS market share

As of 2019, the Apple iOS market share was 13.4% worldwide, making it the second most popular mobile OS behind Google Android, according to IDC.

Apple iOS features

- Wi-Fi, Bluetooth and cellular connectivity, along with VPN support.
- Integrated search support, which enables simultaneous search through files, media, applications and email.
- Gesture recognition supports -- for example, shaking the device to undo the most recent action.
- Push email.
- Safari mobile browser.
- Integrated front- and rear-facing cameras with video capabilities.
- Direct access to the Apple App Store and the iTunes catalog of music, podcasts, television shows and movies available to rent or purchase. iOS is also designed to work with Apple TV.
- Compatibility with Apple's cloud service, iCloud.
- Siri is Apple's virtual assistant that can set reminders, offer suggestions or interact with certain third-party apps. Siri's voice has been modified recently to make it sound more natural.
- Cross-platform communications between Apple devices through AirDrop.
- Support for Apple Watch, runs watchOS 9 but requires iPhone 8 or later running iOS 16 or later.
- Apple Pay, which stores users' credit card data and allows them to pay for goods and services directly with an iOS device.

- CarPlay allows users to interact with an iOS device while driving. CarPlay supports Siri voice controls, and users can access apps through a connected vehicle's touchscreen. CarPlay provides access to maps, phone, calendar, messaging, and music apps.
- The HomePod feature allows Siri to identify family members by voice, giving everyone a personalized experience. HomePod's handoff feature allows users to hand off music, podcasts and phone calls so that they can listen on another device.
- HomeKit allows iOS to be used as a tool for controlling home automation. HomeKit accessories include routers, lights, security cameras and more. The Home app allows you to control these devices from iOS.

What are the security and privacy features of Apple iOS?

iOS includes the following security features:

- **Apple ID support.** Users can sign into websites and apps using their existing Apple ID. Additionally, iOS supports signing in using Face ID or Touch ID, which use biometric authentication methods. Apple IDs are protected with two-factor authentication.
- **Privacy and security.** iOS supports fine-grained controls that prevent apps from gaining location information or accepting AirDrop content from unknown senders. Apps can also be blocked from using Wi-Fi or Bluetooth without users' permission. Additionally, iOS devices use a secure boot chain to ensure that only trusted (signed) code is executed during the boot process. This allows iOS devices to verify the integrity of any code running on the device.
- **Secure Enclave Support.** Secure Enclave is a hardware-based feature that stores cryptographic keys in an isolated location to prevent those keys from being compromised. Secure Enclave is not exclusive to iOS devices. It also works with Apple TV, Apple Watch, Mac computers and other Apple products.

Apple iOS version history

Apple iOS was originally known as iPhone OS. The company released three versions of the mobile OS under that name before iOS 4 debuted in June 2010. Apple released iOS 2 on July 11, 2008. It premiered alongside Apple's iPhone 3G. This operating system was followed on June 17, 2009 by iOS 3. The fourth version of iOS was released on June 21, 2010, along with the iPhone 4.

On Oct. 12, 2011, Apple released iOS 5, which expanded the number of available applications to over 500,000. This iOS version also added the Notification Center, a camera app, Siri and more.Unveiled on June 11, 2012, iOS 6 included a Maps application and the Passbook ticket storage and loyalty program application.

Released on Sept. 18, 2013, iOS 7 featured an entirely redesigned user interface. In September 2014, iOS 8 introduced Continuity, a cross-platform system that allows users of multiple Apple devices to pick up on one where they left off from another. Other new features included the Photos app and Apple Music.

ANDROID OPERATING SYSTEM

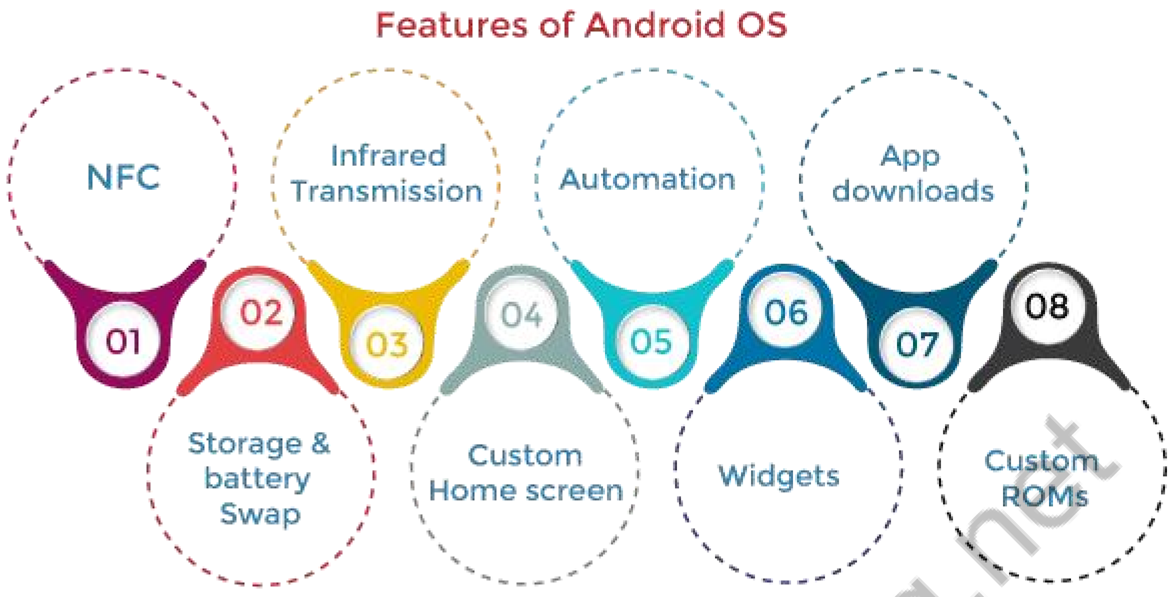
Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a partnership of developers known as the Open Handset Alliance and commercially sponsored by Google. It was disclosed in November 2007, with the first commercial Android device, the HTC Dream, launched in September 2008.

It is free and open-source software. Its source code is Android Open Source Project (AOSP), primarily licensed under the Apache License. However, most Android devices dispatch with additional proprietary software pre-installed, mainly Google Mobile Services (GMS), including core apps such as Google Chrome, the digital distribution platform Google Play and the associated Google Play Services development platform.

- About 70% of Android Smartphone runs Google's ecosystem, some with vendor-customized user interface and some with software suite, such as ***TouchWiz***and later ***One UI*** by Samsung, and ***HTC Sense***.
- Competing Android ecosystems and forksinclude Fire OS (developed by Amazon) or LineageOS. However, the "Android" name and logo are trademarks of Google which impose standards to restrict "uncertified" devices outside their ecosystem to use android branding.

Features of Android Operating System

Below are the following unique features and **characteristics of the android operating system, such as:**



1. Near Field Communication (NFC)

Most Android devices support NFC, which allows electronic devices to interact across short distances easily. The main goal here is to create a payment option that is simpler than carrying cash or credit cards, and while the market hasn't exploded as many experts had predicted, there may be an alternative in the works, in the form of Bluetooth Low Energy (BLE).

2. Infrared Transmission

The Android operating system supports a built-in infrared transmitter that allows you to use your phone or tablet as a remote control.

3. Automation

The ***Tasker*** app allows control of app permissions and also automates them.

4. Wireless App Downloads

You can download apps on your PC by using the Android Market or third-party options like ***AppBrain***. Then it automatically syncs them to your Droid, and no plugging is required.

5. Storage and Battery Swap

Android phones also have unique hardware capabilities. Google's OS makes it possible to upgrade, replace, and remove your battery that no longer holds a charge. In addition, Android phones come with SD card slots for expandable storage.

6. Custom Home Screens

While it's possible to hack certain phones to customize the home screen, Android comes with this capability from the get-go. Download a third-party launcher like ***Apex, Nova***, and you can add gestures, new shortcuts, or even performance enhancements for older-model devices.

7. Widgets

Apps are versatile, but sometimes you want information at a glance instead of having to open an app and wait for it to load. Android widgets let you display just about any feature you choose on the home screen, including weather apps, music widgets, or productivity tools that helpfully remind you of upcoming meetings or approaching deadlines.

8. Custom ROMs

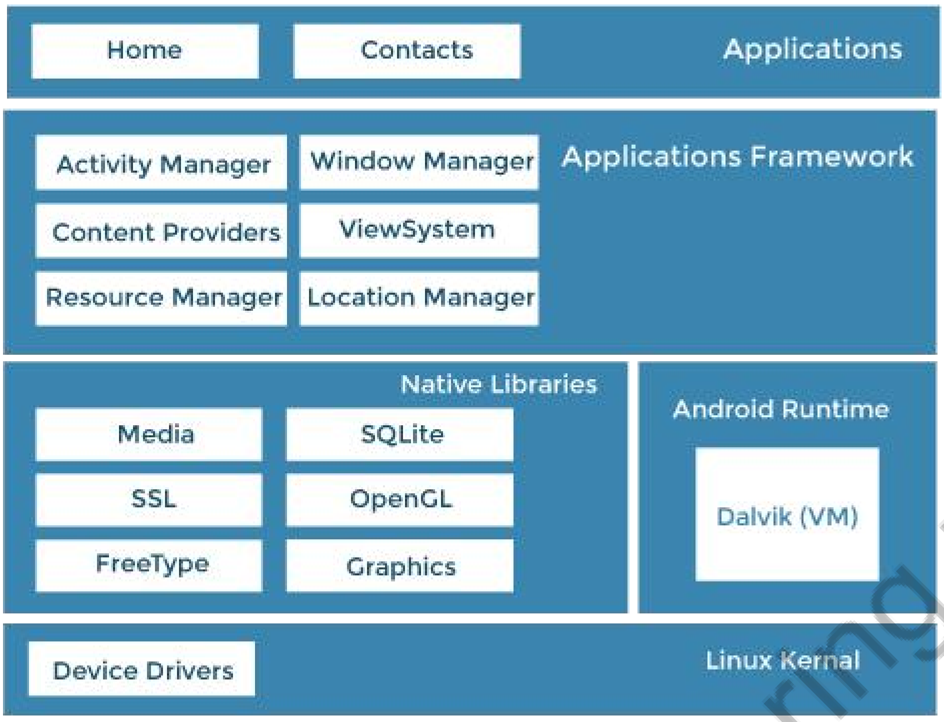
Because the Android operating system is open-source, developers can twist the current OS and build their versions, which users can download and install in place of the stock OS. Some are filled with features, while others change the look and feel of a device. Chances are, if there's a feature you want, someone has already built a custom ROM for it.

Architecture of Android OS

The android architecture contains a different number of components to support any android device needs. Android software contains an open-source Linux Kernel with many C/C++ libraries exposed through application framework services.

Among all the components, Linux Kernel provides the main operating system functions to Smartphone and Dalvik Virtual Machine (DVM) to provide a platform for running an android application. An android operating system is a stack of software components roughly divided into five sections and four main layers, as shown in the below architecture diagram.

- Applications
- Application Framework
- Android Runtime
- Platform Libraries
- Linux Kernel



1. Applications

An application is the top layer of the android architecture. The pre-installed applications like camera, gallery, home, contacts, etc., and third-party applications downloaded from the play store like games, chat applications, etc., will be installed on this layer.

It runs within the Android run time with the help of the classes and services provided by the application framework.

2. Application framework

Application Framework provides several important classes used to create an Android application. It provides a generic abstraction for hardware access and helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.

It includes different types of services, such as activity manager, notification manager, view system, package manager etc., which are helpful for the development of our application according to the prerequisite.

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications. The Android framework includes the following key services:

- **Activity Manager:** Controls all aspects of the application lifecycle and activity stack.
- **Content Providers:** Allows applications to publish and share data with other applications.

- **Resource Manager:** Provides access to non-code embedded resources such as strings, colour settings and user interface layouts.
- **Notifications Manager:** Allows applications to display alerts and notifications to the user.
- **View System:** An extensible set of views used to create application user interfaces.

3. Application runtime

Android Runtime environment contains components like core libraries and the Dalvik virtual machine (DVM). It provides the base for the application framework and powers our application with the help of the core libraries.

Like ***Java Virtual Machine*** (JVM), ***Dalvik Virtual Machine*** (DVM) is a register-based virtual machine designed and optimized for Android to ensure that a device can run multiple instances efficiently.

It depends on the layer Linux kernel for threading and low-level memory management. The core libraries enable us to implement android applications using the standard ***JAVA*** or ***Kotlin*** programming languages.

4. Platform libraries

The Platform Libraries include various C/C++ core libraries and Java-based libraries such as Media, Graphics, Surface Manager, OpenGL, etc., to support Android development.

- **app:** Provides access to the application model and is the cornerstone of all Android applications.
- **content:** Facilitates content access, publishing and messaging between applications and application components.
- **database:** Used to access data published by content providers and includes SQLite database, management classes.
- **OpenGL:** A Java interface to the OpenGL ES 3D graphics rendering API.
- **os:** Provides applications with access to standard operating system services, including messages, system services and inter-process communication.
- **text:** Used to render and manipulate text on a device display.
- **view:** The fundamental building blocks of application user interfaces.
- **widget:** A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- **WebKit:** A set of classes intended to allow web-browsing capabilities to be built into applications.

- **media:** Media library provides support to play and record an audio and video format.
- **surface manager:** It is responsible for managing access to the display subsystem.
- **SQLite:** It provides database support, and FreeType provides font support.
- **SSL:** Secure Sockets Layer is a security technology to establish an encrypted link between a web server and a web browser.

5. Linux Kernel

Linux Kernel is the heart of the android architecture. It manages all the available drivers such as display, camera, Bluetooth, audio, memory, etc., required during the runtime.

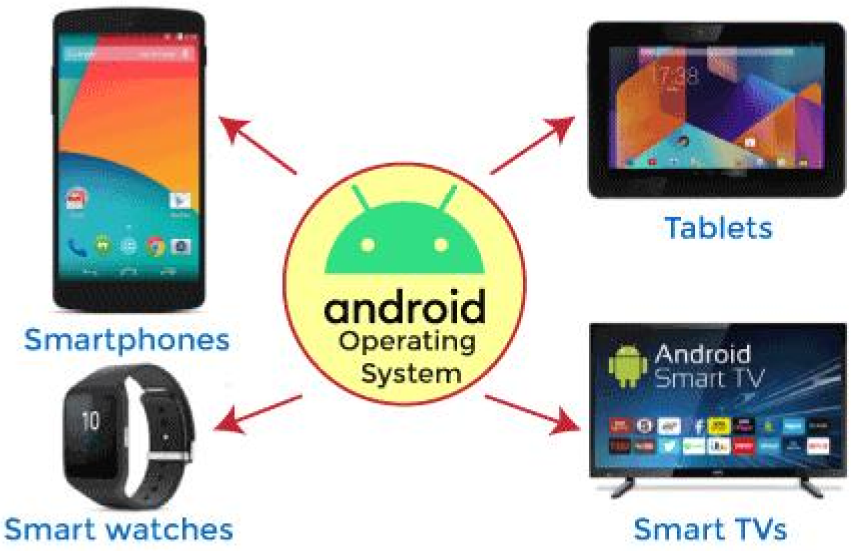
The Linux Kernel will provide an abstraction layer between the device hardware and the other android architecture components. It is responsible for the management of memory, power, devices etc. The features of the Linux kernel are:

- **Security:** The Linux kernel handles the security between the application and the system.
- **Memory Management:** It efficiently handles memory management, thereby providing the freedom to develop our apps.
- **Process Management:** It manages the process well, allocates resources to processes whenever they need them.
- **Network Stack:** It effectively handles network communication.
- **Driver Model:** It ensures that the application works properly on the device and hardware manufacturers responsible for building their drivers into the Linux build.

Android Applications

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as ***Google Play, SlideME, Opera Mobile Store, Mobango, F-droid*** or the ***Amazon Appstore***.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.



Android Emulator

The Emulator is a new application in the Android operating system. The Emulator is a new prototype used to develop and test android applications without using any physical device.

The android emulator has all of the hardware and software features like mobile devices except phone calls. It provides a variety of navigation and control keys. It also provides a screen to display your application. The emulators utilize the android virtual device configurations. Once your application is running on it, it can use services of the android platform to help other applications, access the network, play audio, video, store, and retrieve the data.

Advantages of Android Operating System

We considered every one of the elements on which Android is better as thought about than different platforms. Below are some important advantages of Android OS, such as:

- **Android Google Developer:** The greatest favourable position of Android is Google. Google claims an android operating system. Google is a standout amongst the most trusted and rumoured item on the web. The name Google gives trust to the clients to purchase Android gadgets.
- **Android Users:** Android is the most utilized versatile operating system. More than a billion individuals clients utilize it. Android is likewise the quickest developing operating system in the world. Various clients increment the number of applications and programming under the name of Android.
- **Android Multitasking:** The vast majority of us admire this component of Android. Clients can do heaps of undertakings on the double. Clients can open a few applications on the double and oversee them very. Android has incredible UI, which makes it simple for clients to do multitasking.

- **Google Play Store App:** The best part of Android is the accessibility of many applications. Google Play store is accounted for as the world's largest mobile store. It has practically everything from motion pictures to amusements and significantly more. These things can be effortlessly downloaded and gotten to through an Android phone.
- **Android Notification and Easy Access:** Without much of a stretch, one can access their notice of any SMS, messages, or approaches their home screen or the notice board of the android phone. The client can view all the notifications on the top bar. Its UI makes it simple for the client to view more than 5 Android notices immediately.
- **Android Widget:** Android operating system has a lot of widgets. This gadget improves the client encounter much and helps in doing multitasking. You can include any gadget relying on the component you need on your home screen. You can see warnings, messages, and a great deal more use without opening applications.

Disadvantages of Android Operating System

We know that the Android operating system has a considerable measure of interest for users nowadays. But at the same time, it most likely has a few weaknesses. Below are the following disadvantages of the android operating system, such as:

- **Android Advertisement pop-ups:** Applications are openly accessible in the Google play store. Yet, these applications begin demonstrating tons of advertisements on the notification bar and over the application. This promotion is extremely difficult and makes a massive issue in dealing with your Android phone.
- **Android require Gmail ID:** You can't get to an Android gadget without your email ID or password. Google ID is exceptionally valuable in opening Android phone bolts as well.
- **Android Battery Drain:** Android handset is considered a standout amongst the most battery devouring operating systems. In the android operating system, many processes are running out of sight, which brings about the draining of the battery. It is difficult to stop these applications as the lion's share of them is system applications.
- **Android Malware/Virus/Security:** Android gadget is not viewed as protected when contrasted with different applications. Hackers continue attempting to take your data. It is anything but difficult to target any Android phone, and each day millions of attempts are done on Android phones.