Un sistema operativo administra memoria, cpu y dispositivos. Está obligado a garantizar el correcto funcionamiento del sistema.

# POSIX

Serie de normas, estándares y convenciones con el objetivo de promover la portabilidad de las aplicaciones y el software entre distintos S.O. Define una serie de interfaces y comandos que deben estar en todos los Sistemas que cumplan con el estándar

# Libertades de GNU

* De usar el programa con cualquier propósito
* Estudiar su funcionamiento
* Distribuir sus copias
* Mejorar sus programas

# Kernel Linux:

* Monolítico hibrido porque los drivers y el código del kernel se ejecutan en modo privilegiado y tiene la capacidad de cargar y descargar módulos que proveen funcionalidad

# Directorios más importantes según Filesystem Hierarchy Standard

* / : Raiz
* /home : Archivos de usuario
* /var : Logs, BD, Spools, información que varía de tamaño
* /etc : Archivos de configuración
* /bin : Archivos binarios y ejecutables
* /sbin : Archivos binarios y ejecutables para administrar el sistema
* /dev : Enlace a dispositivos. Denominaciones
  + Basada en buses (IDE):
    - /dev/hdaX
    - /dev/hdbX
  + Basada en identificación de los buses (SATA, SCSI):
    - /dev/sdaX
    - /dev/sdbX
  + Tambien hay otras nomenclaturas, basadas en UUID, labels, etc.
* /usr : Aplicaciones de usuarios
* /boot : Archivos necesarios para el arranque del sistema (Acá está el kernel)

# Archivos de configuracion

* /etc/passwd: : Info básica del usuario, nombre, ID, grupo, nombre completo, directorio de inicio y Shell. (Root es UID 0)
* /etc/group: Grupos y sus usuarios, nombre e ID
* /etc/shadow: Contraseñas de usuarios encriptadas

# Master boot record

* Sector reservado del disco físico (cilindro 0, cabeza 0, sector 1)
* Existe un MBR en todos los discos pero sólo uno es designado como “Primary Master Disk”
* El tamaño coincide con el estándar del sector, 512 Bytes
  + Los primeros bytes son el Master Boot Code
  + A partir del byte 446 está la tabla de particiones, de 64 bytes
  + Al final existen 2 bytes libres o para firmar el MBR
* La ultima acción del BIOS es leer el MBC (llevarlo a memoria y ejecutarlo)
* Si se tiene un sistema instalado se usa el bootloader de MBC
* Debido a su tamaño se restringe a 4 la cantidad de particiones primarias (o 3 primarias y una extendida con sus particiones lógicas). Sólo las primarias pueden marcarse como booteables

# Instalación de Linux

* Cómo mínimo es necesario una sola partición para el directorio raíz
* Pero es recomendable crear al menos 2 (la raíz y el swap)

# Bootloader

* Su finalidad es cargar una imagen de kernel de alguna partición para su ejecución
* Se ejecuta luego del código del BIOS
* Existen 2 modos de instalación
  + En el MBR
  + En el sector de arranque de la partición raíz o activa
* GPT reemplaza al esquema MBR
* Puede llegar a cargar un entorno previo a la carga del sistema

# Proceso de arranque o bootsrap

1. POST y Carga del bootloader: El BIOS/UEFI realiza el post y carga el gestor de arranque
2. Selección de kernel
3. Carga del kernel y del initramfs: El kernel y el initramfs (un sistema de archivos temporal en la RAM) se cargan en memoria. Initramfs contiene controladores y modulos necesarios para montar el sistema de archivos raíz
4. Inicialización del kernel: El kernel inicializa los dispositivos y monta el sistema de archivos raíz
5. Ejecucion de init/systemd: El kernel lanza el primer proceso que suele ser systemd, el cual se encarga de realizar la mayoría de las tareas de inicialización del sistema
6. Inicio de sesión

# UEFI

* Define una interfaz entre el sistema operativo y el firmware. Reemplazando la BIOS.
* Define la ubicación del gestor de arranque y la interfaz entre el mismo y el firmware
* Expone
  + información de hardware y configuración del firmware
  + Punteros a rutinas de servicios

# Comandos comunes para gestión de usuarios

* Useradd nombre : Agrega al usuario y modifica el /etc/passwd
* Passwd nombre : Cambia la contraseña del usuario y modifica /etc/shadow
* Usermod nombre : modifica el grupo y directorio home
* Userdel nombre : Elimina el usuario
* Groupdel nombre : Elimina el grupo

# Permisos de archivos

* Notación octal (RWX)
* Se dividen en 3:
  + Los primeros 3 bits son del dueño del archivo
  + Los segundos 3 bits del grupo
  + Los terceros 3 bits de otros usuarios
* Se pueden cambiar con el comando chmod
* El comando chown y chgrp permiten cambiar el usuario y grupo dueño de un archivo

# SystemV

* Es sincrónico
* Se configura a través del archivo /etc/inittab
* Pasos del inicio de un sistema GNU/Linux:
  1. Se empieza a ejecutar el código del BIOS
  2. El BIOS ejecuta el POST
  3. El BIOS lee el sector de arranque (MBR)
  4. Se carga el gestor de arranque (MBC)
  5. El bootloader carga el kernel y el initrd
  6. Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales
  7. El kernel ejecuta el proceso init y se desmonta el initrd
  8. Se lee el /etc/inittab
  9. Se ejecutan los scripts apuntados por el runlevel 1
  10. El final del runlevel 1 le indica que vaya al runlevel por defecto
  11. Se ejecutan los scripts apuntados por el runlevel por defecto
  12. El sistema está listo para usarse

## Init

* Es ejecutado por el kernel
* Su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO
* Posee el PID 1 y se encuentra en /sbin/init
* En SystemV se lo configura a través del archivo /etc/inittab
* No tiene padre y es el padre de todos los procesos
* Es el encargado de montar los file systems y de hacer disponible los demás dispositivos

## RunLevels

* Son el modo en el que arranca Linux (3 en redhat y 2 en debian). El proceso de arranque se divide en niveles y cada uno es responsable de iniciar o detener una serie de servicios
* Códigos numéricos que indican distintos estados operativos en los que puede encontarse un sistema Linux
* Según el estándar, los niveles son:
  + 0 : Halt (Parada)
  + 1 : Single user mode (monousuario)
  + 2 : Multiusuario sin soporte de red
  + 3 : Multiusuario completo por consola
  + 4 : No se utiliza
  + 5 : Multiusuario completo con login gráfico basado en X
  + 6 : Reboot
* Se encuentran definidos en /etc/inittab, el cual guarda los ids, niveles de ejecución, acciones y procesos que serán ejecutados en cada nivel de ejecución
* El comando “init” permite cambiar el runlevel actual. Pero no es permanente
* Los scripts que se ejecutan están en /etc/init.d
* En /etc/rcX.d hay links a los archivos del /etc/init.d. Donde la X es el runlevel en el que se va a ejecutar. El nombre del link puede iniciar con S (start) o K (kill) el cual define cómo se va a lanzar el script. Y además tienen un número que identifica el orden en el que van a ser iniciados.

# Upstart

* Primer reemplazo para SystemV
* Permite la ejecución de trabajos en forma asíncrona a través de eventos.
* Estos trabajos se denominan Jobs, y definen servicios o tareas a ser ejecutadas por init
* Son scripts de texto plano que definen las acciones o tareas a ejecutar ante determinados eventos
* Cada job es definido en el /etc/init (.conf)
* Pueden ser de dos tipos
  + Task: Ejecución finita
  + Service: Ejecucion indeterminada
* No es compatible con SystemD ni con SystemV
* No existe el archivo /etc/inittab

# SystemD

* Sistema que centraliza la administración de demonios y librerías del sistema
* Mejora el paralelismo del booteo
* Puede ser controlado por “Systemctl” (iniciar, reiniciar, detener, recargar configuración, ver el estado, etc.)
* Es compatible con SystemV si es llamado como init
* El demonio “SystemD” reemplaza al proceso init (Pasa a tener el PID 1)
* Los runlevels son reemplazados por targets
* No existe el archivo /etc/inittab
* Las unidades de trabajo son denominadas units de tipo
  + Service: Controla un servicio particular
  + Socket: Encapsula IPc, un socket del sistema o file system FIFO.
    - Mecanismo de iniciación bajo demanda: Cuando el socket recibe una conexión, spawnea el servicio y le pasa el socket
  + Target: Agrupa units o establece puntos de sincronización durante el booteo
  + Snapshot: Almacena el estado de un conjunto de unidades que puede ser reestablecido más tarde
  + Etc.
* Las units pueden estar activas o inactivas

## Cgroups

* Forma de organizar un grupo de procesos en forma jerarquica en SystemD
* Puede limitar el uso de recursos
* Agrupa un conjunto de procesos relacionados

# FStab

* Define qué particiones se montan al arranque
* Su configuración se encuentra en /etc/fstab
* Opciones:
  + User : Cualquier usuario puede montar la partición
  + Auto : Monta la partición al inicio
  + RO: Read only
  + RW: Read / write
  + Etc.

# Procesos

* Cada proceso es un programa en ejecución
* Tiene un PID y un PPID (Process ID, Parent Process ID)
* Pueden tener el UID del usuario que lo inició, el tiempo que lleva iniciado, comando utilizado, Shell que usa, cantidad de memoria. Etc.
* Para ejecutar un comando en background se puede usar “comando &”
* Todos los procesos tienen 3 archivos abiertos. Los cuales se identifican con un “file descriptor”
  + STDIN : Entrada estándar, permite que el proceso reciba datos de entrada. Usa el file descriptor 0
  + STOUT : Salida estándar, se utiliza para la salida normal del proceso. Usa el file descriptor 1
  + STDERR : Salida de error estándar, se utiliza para la salida de mensajes de error o advertencias. Usa el file descriptor 2

# Empaquetamiento

* “tar -czf nombre.tar.gz directorio” -> Empaquetar y comprimir todos los archivos del directorio
* “tar -xzf nombre.tar.gz” -> Desempaquetar y descomprimir un archivo
* “tar -cf nombre.tar directorio” -> Empaquetar todos los archivos del directorio
* “tar -xf nombre.tar” -> Desempaquetar todos los archivos
* En resumen:
  + Empaquetar: -czf (empaqueta y comprime) y -cf (empaqueta sin comprimir).
  + Extraer: -xzf (extrae y descomprime) y -xf (extrae sin descomprimir).

En el año 1992, Torvalds y Stallman deciden fusionar los proyectos Linux y GNU con el fin de crear GNU/Linux.

Existen distribuciones que incluyen paquetes propietarios y otras que son 100% libres.

Una de las principales diferencias entre algunas distribuciones es el formato de sus paquetes instalables.

Orden de secuencia de booteo:

1 - Se ejecuta el código de la BIOS.

2 - El hardware lee el sector de arranque.

3 - Se carga el gestor de arranque.

4 - Se carga el kernel.

La versión estable actual de Linux es la 6.10.7 .

El shell de GNU/Linux está formado por archivos binarios ejecutables como cualquier otro programa.

Las distribuciones GNU/Linux se diferencian en las aplicaciones que instalan y la versión del kernel que trae entre otras cosas.

Existen múltiples distribuciones GNU/Linux para distintos propósitos (servidores, maquinas de escritorio, tablets, celulares, etc).

El runlevel 6 hace referencia a un modo reboot

/user/bin no forma parte del FHS (file hierarchy system)

La imagen del kernel Linux se encuentra dentro de /boot según el FHS.

POSIX es un estándar cuya finalidad es facilitar la portabilidad de las aplicaciones en el nivel fuente, incluso en el nivel fuente, entre muchos sistemas. Es un conjunto de estándares definiendo APIs para los sistemas operativos que lo cumplen.

El kernel de Linux es monolítico híbrido. Que sea monolítico implica que todo el sistema operativo se ejecuta en el espacio de kernel. Que sea híbrido indica que permite modularidad, es decir, permite componentes como controladores de dispositivos o sistemas de archivos cargados o descargados de manera dinámica, sin reiniciar el kernel.

Existe un MBR por disco

Una distribución GNU/Linux se diferencia de otra en el conjunto de aplicaciones que se instalan por defecto, el gestor de programas/paquetes que utilizan, la versión de kernel a la que da soporte.

Una máquina con arranque UEFI utiliza GPT como formato para su tabla de particiones.

En una tabla de particiones puede haber definido como máximo particiones primarias y una extendida. Las particiones primarias son las principales particiones del disco, teniendo acceso directo al sistema operativo. La partición extendida actúa como contenedor para particiones lógicas.

El FHS hace referencia a un estándar para organizar archivos y directorios archivos del file system.

La GPL es una licencia de derechos de autor.

Si sabemos el nombre de un proceso, podemos matarlo con el comando killall.

La funcionalidad del kernel se basa en administrar el uso de CPU y administrar la memoria principal.