Practica 1

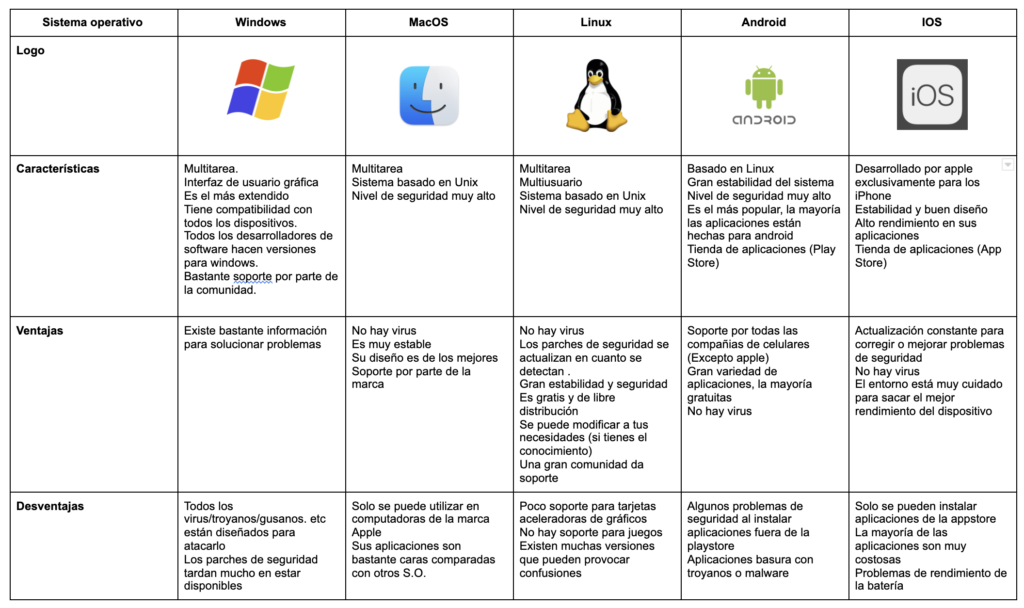
1. Características de GNU/Linux:

* Es multiusuario, multitarea y multiprocesador
* Es altamente portable
* Posee diversos intérpretes de comandos, de los cuales algunos

son programables

* Permite el manejo de usuarios y permisos
* Todo es un archivo (hasta los dispositivos y directorios). Cada directorio puede estar en una partición diferente (/temp, /home, etc.)
* Es case sensitive
* Es código abierto

1. Comparación con otros sistemas operativos:



1. GNU = GNU No es Unix. GNU es un sistema operativo de tipo Unix, así como una gran colección de programas informáticos que componen al sistema, desarrollado por y para el Proyecto GNU y auspiciado por la Free Software Foundation. Está formado en su totalidad por software libre, mayoritariamente bajo términos de copyleft.
2. El proyecto para desarrollar el sistema GNU se denomina «Proyecto GNU». El Proyecto GNU se concibió en 1983 como un modo de retomar el espíritu cooperativo que prevalecía en la comunidad informática en sus comienzos, posibilitar la cooperación eliminando los obstáculos impuestos por los dueños de software privativo.
3. En 1971, cuando comenzó su carrera en el MIT, Richard Stallman formaba parte de un grupo de trabajo que usaba exclusivamente software libre. Incluso las compañías informáticas distribuían con frecuencia software libre. Los programadores eran libres de cooperar unos con otros y lo hacían a menudo.

Al inicio de los años ochenta casi todo el software era privativo, lo que significa que tenía dueños que prohibían e impedían la cooperación entre usuarios. De ahí que fuera necesario el Proyecto GNU.

A principios de 1990 ya se habían encontrado o programado los componentes principales excepto uno, el núcleo. En 1991 Linus Torvalds programó Linux, un núcleo similar a Unix, y en 1992 lo convirtió en software libre. La combinación de Linux con el sistema GNU, que ya estaba prácticamente completo, formó un sistema operativo completo: el sistema GNU/Linux. Se estima que existen decenas de millones de personas que en la actualidad usan sistemas GNU/Linux, habitualmente mediante distribuciones de GNU/Linux.

1. Un sistema operativo multitarea divide el tiempo de procesador disponible entre los procesos o subprocesos que lo necesitan. El sistema está diseñado para la multitarea preferente; asigna un segmento de tiempo de procesador a cada subproceso que ejecuta. El subproceso que se está ejecutando actualmente se suspende cuando transcurre su segmento de tiempo, lo que permite que se ejecute otro subproceso. Cuando el sistema cambia de un subproceso a otro, guarda el contexto del subproceso adelantado y restaura el contexto guardado del siguiente subproceso de la cola.
2. POSIX es una serie de normas, estándares y convenciones creadas por la comunidad de usuarios de Unix. Se utilizan con el fin de mejorar la compatibilidad entre diferentes sistemas operativos Unix y proporcionar una base para que las aplicaciones sean más portables entre distintos sistemas operativos.

2.

1. Una distribución GNU/Linux es una colección de paquetes de software, recopilados y mantenidos por un grupo de personas. Esta tiene un núcleo Linux. Algunas distribuciones:

* #31 - Slackware (1992): Es la distribución más antigua aún mantenida activamente. La configuración se hace a mano, editando “documentos de texto escondidos en lugares al azar”. Extremadamente estable y segura.
* #126 - S.u.S.E. (1992): Pionera en la incorporación de componentes gráficos “de fábrica”. Adquirida por Novell en 2004. Firmó un controversial acuerdo con Microsoft, eludiendo la protección brindada por GPL v2.
* #2 - Debian (1993): Programas y documentación hoy y siempre 100% gratuitos. Ultra estable. Muy buena para usuarios y/o para servidores. Su ciclo de desarrollo de las nuevas versiones es un tanto lento. Cuenta con más de 50.000 paquetes de software listos para ser instalados. Órganos de gobierno súper democráticos.
* #43 - Red Hat (1995): Posiblemente la compañía asociada a GNU/Linux más conocida del mundo. RHEL, orientada a servidores, cuesta desde $349 dólares en su versión más elemental hasta $1.299 dólares en la más completa.

1. El proyecto Debian se enfoca en desarrollar un sistema operativo completo y funcional compuesto por software libre, y sus objetivos principales incluyen:

* Software Libre: Debian se compromete a proporcionar un sistema completamente basado en software libre. Esto significa que todos los programas incluidos en Debian deben cumplir con la definición de software libre establecida por la Free Software Foundation (FSF). Esta filosofía promueve la libertad de los usuarios para ejecutar, estudiar, modificar y distribuir el software.
* Estabilidad y Calidad: Debian se enorgullece de su enfoque en la estabilidad y la calidad del sistema. Esto se logra mediante rigurosas pruebas y un proceso de liberación estable, lo que hace que Debian sea una elección popular para servidores y estaciones de trabajo que requieren una alta confiabilidad.
* Amplia Variedad de Arquitecturas: Debian se desarrolla para una amplia variedad de arquitecturas de hardware, lo que lo hace versátil y adecuado para una amplia gama de dispositivos y plataformas.
* Comunidad y Colaboración: Debian es un proyecto comunitario donde miles de voluntarios de todo el mundo contribuyen al desarrollo, documentación y soporte del sistema operativo.

Cronología breve del proyecto Debian:

* Agosto de 1993: Debian se inicia como un proyecto personal por Ian Murdock. El nombre "Debian" es una combinación de su nombre con el de su entonces novia, Debra.
* Septiembre de 1993: Se lanza la primera versión de Debian, Debian 0.01.
* 1994-1996: Se establece una estructura organizativa y se lanzan varias versiones. Debian se gana una reputación por su enfoque en la calidad y la ética del software libre.
* 1997: Se establece la Debian Free Software Guidelines (Directrices de Software Libre de Debian), que definen los principios que guiarán la inclusión de software en Debian.
* 2000: Debian 2.2, también conocido como "Potato", se convierte en la primera versión oficialmente lanzada con la nueva arquitectura AMD64.
* 2005: Se lanza Debian 3.1, apodado "Sarge", con soporte oficial para múltiples arquitecturas y herramientas de instalación mejoradas.
* 2013: Debian 7.0, llamado "Wheezy", se lanza con mejoras en seguridad, actualizaciones de software y soporte para sistemas de archivos Btrfs.
* 2017: Debian 9.0, conocido como "Stretch", se lanza con mejoras en la seguridad, nuevas versiones de software y compatibilidad con Secure Boot.
* 2019: Debian 10.0, llamado "Buster", se lanza con características como el administrador de paquetes APT 1.8 y mejoras en el soporte para sistemas embebidos.

3.

* 1. Los tres componentes fundamentales de GNU/Linux son:
* Kernel (Núcleo): El núcleo de Linux es el componente esencial del sistema operativo que se encarga de gestionar los recursos de hardware y proporcionar servicios básicos como la administración de procesos, la administración de memoria y la comunicación con dispositivos de hardware.
* GNU Utilities (Herramientas GNU): Estas son las utilidades y programas proporcionados por el proyecto GNU (GNU's Not Unix) que complementan el kernel de Linux para formar un sistema operativo funcional. Incluyen comandos de línea de comandos, bibliotecas, herramientas de compilación y otras utilidades esenciales.
* Entorno de Usuario: Este componente incluye las aplicaciones y programas que los usuarios ejecutan para realizar tareas específicas. Puede incluir aplicaciones de oficina, navegadores web, editores de texto, reproductores multimedia y una amplia variedad de software de usuario final.
  1. La estructura básica del sistema operativo GNU/Linux se organiza de la siguiente manera:
* Ejecuta programas y gestiona dispositivos de hardware
* Es el encargado de que el software y el hardware puedan

trabajar juntos

* Sus funciones más importantes son la administración de

memoria, CPU y la E/S

* En sí, y en un sentido estricto, es el sistema operativo
* Es un núcleo monolítico hibrido: Los drivers y código del Fernel se ejecutan en modo

Privilegiado. Lo que lo hace hibrido es la capacidad de cargar y descargar funcionalidad a través de módulos

* Este licenciado bajo la licencia GPL v2

4.

a) El kernel es definido como el núcleo o corazón del sistema operativo, y se encarga principalmente de mediar entre los procesos de usuario y el hardware disponible en la máquina, es decir, concede el acceso al hardware, al software que lo solicite, de una manera segura; y el procesamiento paralelo de varias tareas.

Un poco de historia:

* En 1991 Linus Torvalds inicia la programación de un Kernel Linux basado en Minix (clon de Unix desarrollado por Tenembaum en 1987 con el fin de crear un S.O. de uso didáctico)
* El 5 de octubre de 1991, se anuncia la primera versión “oficial” de Linux (0.02)
* En 1992 se combina su desarrollo con GNU, formando GNU/Linux
* La versión 1.0 apareció el 14 de marzo de 1994
* En mayo de 1996 se decide adoptar a Tux como mascota oficial de Linux
* En julio de 1996 se lanza la versión 2.0
* En enero de 1999 se lanza la versión 2.2, que provee mejoras de portabilidad entre otras y se desarrolla hasta febrero de 2004 terminando en la versión 2.2.26
* El 17 de julio de 2011 se lanza la versión 3.0

B ) Funciones del kernel:

* Gestión de la memoria: supervisa cuánta memoria se utiliza para almacenar qué tipo de elementos, así como el lugar en que los guarda.
* Gestión de los procesos: determina qué procesos pueden usar la unidad central de procesamiento (CPU), cuándo y durante cuánto tiempo.
* Controladores de dispositivos: actúa como mediador o intérprete entre el hardware y los procesos.
* Seguridad y llamadas al sistema: recibe solicitudes de servicio por parte de los procesos.

C La última versión de la serie Linux kernel 6.2, Linux kernel 6.2.16, estuvo disponible el 17 de mayo de 2023. Antes de la versión 2.6, los números impares indicaban desarrollo, los pares producción. A partir de la versión 2.6 del kernel de Linux, se introdujo un nuevo esquema de versionado que utilizaba cuatro números, como por ejemplo, 2.6.32.1.

D Es posible tener más de un kernel de GNU/Linux instalado en la misma máquina, y esta práctica se conoce como "multiboot" o "multiarranque".

E El kernel se encuentra en /sbin/init.

F Tipos de kernel:

* Kernel monolítico. Un kernel grande para todas las tareas. Es el único responsable de la gestión de la memoria y de los procesos, de la comunicación entre procesos y proporciona funciones de soporte de drivers y hardware. Los sistemas operativos que usan el kernel monolítico son Linux, OS X y Windows.
* Microkernel. El microkernel se ha diseñado intencionadamente de un tamaño pequeño para que en caso de fallo no paralice todo el sistema operativo. No obstante, para que pueda asumir las mismas funciones que un kernel grande, está dividido en varios módulos. Como ejemplo de aplicación solo existe el componente Mach de OS X, ya que hasta ahora no hay ningún sistema operativo con microkernel.
* Kernel híbrido. La combinación del kernel monolítico y el microkernel se denomina kernel híbrido. En este caso, el kernel grande se hace más compacto y modulable. Otras partes del kernel pueden cargarse dinámicamente. Esto ya ocurre en cierta medida en Linux y OS X.

El kernel de GNU/Linux se considera principalmente monolítico porque la mayoría de sus funciones esenciales se ejecutan en un solo espacio de memoria y como parte de un solo programa. Sin embargo, su capacidad para cargar módulos de kernel de manera dinámica agrega elementos de un kernel híbrido, lo que le brinda flexibilidad y modularidad.

5 Shell

A Modo de comunicación entre el usuario y el SO. Ejecuta programas a partir del ingreso de comandos, cada usuario puede tener una interfaz o Shell.

B El shell en un sistema operativo tiene las siguientes funciones principales:

* Interacción con el Usuario: Proporciona una interfaz de línea de comandos para que los usuarios interactúen con el sistema operativo.
* Ejecución de Comandos: Interpreta y ejecuta comandos ingresados por el usuario.
* Administración de Procesos: Permite crear, administrar y controlar procesos en el sistema.
* Gestión de Archivos: Facilita la navegación y administración de archivos y directorios.
* Redirección y Automatización: Permite redirigir la entrada/salida y automatizar tareas a través de scripts.
* Personalización: Permite a los usuarios personalizar su entorno y crear alias de comandos.
* Historial de Comandos: Mantiene un registro de comandos previamente ejecutados.
* Gestión de Variables de Entorno: Controla variables que afectan el comportamiento del sistema y programas.

C

Bash (Bourne-Again SHell):

Popularidad: Bash es el shell más comúnmente utilizado en sistemas GNU/Linux y es el shell predeterminado en muchas distribuciones.

Características Destacadas: Bash es altamente personalizable, admite autocompletado avanzado (tabulación), redirección y tuberías (piping) de comandos, y ofrece una amplia gama de características de scripting y programación, como bucles y condicionales.

Historial de Comandos: Almacena un historial de comandos ejecutados por el usuario, lo que facilita la recuperación y reejecución de comandos anteriores.

Uso Común: Ideal para usuarios avanzados, administradores de sistemas y desarrolladores de scripts debido a su versatilidad y potencia.

Zsh (Z Shell):

Características Destacadas: Zsh es conocido por su robusto sistema de autocompletado, que sugiere y completa automáticamente comandos y argumentos, lo que lo hace muy eficiente para usuarios que pasan mucho tiempo en la línea de comandos.

Plugins y Temas: Zsh es altamente personalizable y admite una variedad de plugins y temas que permiten a los usuarios adaptar su entorno de shell según sus preferencias.

Historial de Comandos: Al igual que Bash, Zsh mantiene un historial de comandos para facilitar la búsqueda y reejecución de comandos anteriores.

Uso Común: Zsh es popular entre los usuarios avanzados y aquellos que buscan una experiencia de línea de comandos altamente personalizable y eficiente.

Fish (Friendly Interactive SHell):

Características Destacadas: Fish se centra en la usabilidad y la amigabilidad. Ofrece una experiencia de línea de comandos más intuitiva y fácil de usar para usuarios principiantes. Su autocompletado es especialmente amigable.

Colorido por Defecto: Fish utiliza colores y resaltado de sintaxis por defecto, lo que hace que los mensajes y la salida de comandos sean más legibles y atractivos visualmente.

Historial de Comandos: Al igual que los otros shells, Fish mantiene un historial de comandos, pero su búsqueda de historial es más intuitiva y rápida.

Uso Común: Fish es ideal para usuarios principiantes y aquellos que desean una experiencia de línea de comandos más amigable y visualmente atractiva.

D Los comandos internos del shell, como cd, echo, pwd, son parte del propio shell y se ejecutan directamente desde su código. No se encuentran en ubicaciones específicas en el sistema de archivos.

Los comandos internos son intrínsecos al funcionamiento del shell y no se buscan en directorios del PATH.

Los comandos externos son programas o scripts que no están incorporados en el shell, y se almacenan en directorios específicos del sistema de archivos.

Para ejecutar un comando externo, el shell busca en los directorios enumerados en la variable de entorno PATH.

E El shell y el kernel son componentes distintos de un sistema operativo. El kernel es el núcleo del sistema operativo y se encarga de la gestión de hardware y servicios fundamentales, mientras que el shell es una interfaz de usuario que permite a los usuarios interactuar con el sistema operativo y ejecutar comandos. Su naturaleza y funciones son diferentes, y es por eso que se consideran componentes separados en un sistema GNU/Linux.

F En la mayoría de los sistemas GNU/Linux es posible definir un intérprete de comandos distinto para cada usuario. Esto se logra mediante la modificación del archivo de configuración de inicio de sesión del usuario correspondiente. El intérprete de comandos se inicia automáticamente cuando un usuario inicia sesión en el sistema.

El archivo de configuración de inicio de sesión más comúnmente utilizado es el archivo ~/.bashrc para el shell Bash (Bourne-Again SHell), que es el shell predeterminado en muchas distribuciones de GNU/Linux.

6. Sistema de archivos

* 1. Un sistema de archivos es el sistema de almacenamiento de un dispositivo de memoria, que estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta. El objetivo principal de esta organización es que el usuario pueda identificar los archivos sin lugar a error y acceder a ellos lo más rápido posible. Organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento (fat, ntfs ext2, ext3, reiser, etc.). El adoptado por GNU/Linux es el Extended (v2, v3, v4).
  2. Linux soporta gran variedad de sistemas de ficheros, desde sistemas basados en discos, como pueden ser ext2, ext3, ReiserFS, XFS, JFS, UFS, ISO9660, FAT, FAT32 o NTFS, a sistemas de ficheros que sirven para comunicar equipos en la red de diferentes sistemas operativos, como NFS (utilizado para compartir recursos entre equipos Linux) o SMB (para compartir recursos entre máquinas Linux y Windows).
  3. Sí, es posible visualizar particiones del tipo FAT (File Allocation Table) y NTFS (New Technology File System) en sistemas GNU/Linux.
  4. Directorios más importantes según FHS (Filesystem Hierarchy Standard):
* **/** Tope de la estructura de directorios. Es como el C:
* **/home** Se almacenan archivos de usuarios (Mis documentos)
* **/var** Información que varía de tamaño (logs, BD, spools)
* **/etc** Archivos de configuración
* **/bin** Archivos binarios y ejecutables
* **/dev** Enlace a dispositivos
* **/usr** Aplicaciones de usuarios

7. Particiones

a) Una partición es una forma de dividir lógicamente el disco físico, es decir, tener varias particiones es como tener varios discos duros en un solo disco duro físico, cada uno con su sistema de archivos y funcionando de manera diferente.

Tipos de particiones:

* Partición primaria: división cruda del disco (puede haber 4 por disco). Se almacena información de la misma en el MBR.
* Partición extendida: sirve para contener unidades lógicas en su interior. Solo puede existir una partición de este tipo por disco. No se define un tipo de FS directamente sobre ella.
* Partición lógica: ocupa la totalidad o parte de la partición extendida y se le define un tipo de FS. Las particiones de este tipo se conectan como una lista enlazada.

Ventajas:

* Facilidad de Reinstalación: Separar los archivos de Windows de la información personal facilita la reinstalación de Windows sin perder datos personales.
* Copias de Seguridad: Las particiones ofrecen un espacio dedicado para copias de seguridad, lo que ayuda a proteger los datos importantes.
* Mayor Seguridad: Si una partición se infecta con virus, tener información importante en otra partición puede evitar pérdidas catastróficas.
* Mejor Organización: Las particiones permiten una organización más eficiente de los datos según importancia, tipo o tamaño.
* Instalación de Varios Sistemas Operativos: Las particiones son esenciales para crear sistemas de arranque dual o múltiple con diferentes sistemas operativos.
* Uso de Máquinas Virtuales: Para quienes trabajan con máquinas virtuales, las particiones separadas ayudan a mantener separados los contenidos del PC principal y las máquinas virtuales.

Desventajas:

* Experiencia Más Lenta: Un exceso de particiones puede dificultar la búsqueda de archivos, requiriendo una organización meticulosa.
* Posibilidad de Errores: Con más particiones, existe una mayor probabilidad de que surjan errores.
* Desorden en los Volúmenes: Es fácil crear particiones innecesarias o confusas durante el proceso de particionamiento, especialmente en unidades pequeñas.
* Innecesario para Usuarios Básicos: Aquellos que utilizan el PC para tareas simples pueden no encontrar utilidad en crear particiones, ya que puede ser una opción compleja y poco necesaria para ellos.

1. En GNU/Linux, las particiones y dispositivos de almacenamiento se identifican mediante un esquema de nomenclatura que utiliza el sistema de archivos udev y la estructura del sistema de archivos /dev. La identificación de las particiones depende del tipo de dispositivo de almacenamiento que estés utilizando, ya sea IDE, SCSI, SATA u otro.

* Los discos IDE generalmente se nombran como /dev/hdX, donde X es una letra que representa el disco IDE específico, comenzando desde a (por ejemplo, /dev/hda, /dev/hdb, /dev/hdc, etc.). Las particiones en un disco IDE se numeran secuencialmente, por lo que una partición en el primer disco IDE se nombra como /dev/hda1, la segunda partición como /dev/hda2, y así sucesivamente.
* Los discos SCSI y dispositivos SCSI emulados (como la mayoría de los dispositivos SATA modernos) se nombran como /dev/sdX, donde X es una letra que representa el disco SCSI específico, comenzando desde a (por ejemplo, /dev/sda, /dev/sdb, /dev/sdc, etc.). Las particiones en un disco SCSI se numeran de la misma manera que en los discos IDE, por lo que la primera partición en el primer disco SCSI se nombra como /dev/sda1, la segunda como /dev/sda2, y así sucesivamente.
* Los discos SATA se manejan de la misma manera que los discos SCSI en GNU/Linux. Por lo tanto, se nombran como /dev/sdX, donde X representa el disco SATA específico, comenzando desde a (por ejemplo, /dev/sda, /dev/sdb, /dev/sdc, etc.).

1. Como mínimo es necesario una partición (para el /), es recomendable crear al menos 2 (/ y SWAP).
2. Diversos casos de particionamiento dependiendo la tarea:

* Uso Doméstico Básico:
  + Partición raíz (/): 50 GB (Ext4, punto de montaje /).
  + Partición de intercambio (swap): 2 GB.
* Desarrollo y Uso Personal:
  + Partición raíz (/): 30 GB (Ext4, punto de montaje /).
  + Partición /home: Resto del espacio (Ext4, punto de montaje /home).
  + Partición de intercambio (swap): 2 GB.
* Servidor Web o Servidor de Aplicaciones:
  + Partición raíz (/): 20 GB (Ext4, punto de montaje /).
  + Partición /var: 10 GB (Ext4, punto de montaje /var).
  + Partición /home: Resto del espacio (Ext4, punto de montaje /home).
  + Partición de intercambio (swap): 4 GB.
* Máquina Virtual o Ambiente de Pruebas:
  + Partición raíz (/): 20 GB (Ext4, punto de montaje /).
  + Partición /var: 10 GB (Ext4, punto de montaje /var).
  + Partición de intercambio (swap): 4 GB.

1. El software que se usa para particionar, se denomina particionas, existen dos tipos:

* Destructivos: permiten crear y eliminar particiones (fdisk).
* No destructivo: permiten crear, eliminar y modificar particiones (\_ps, gparted).

8.

1. La **BIOS** (Basic Input/Output System, por sus siglas en inglés) es un conjunto de software de bajo nivel que se encuentra en la placa base (o placa madre) de una computadora. Su tarea principal es proporcionar una interfaz entre el hardware de la computadora y el sistema operativo, permitiendo que el sistema se inicie correctamente y realice operaciones esenciales.
2. **UEFI** (Unified Extensible Firmware Interface).

* Define la ubicación de gestor de arranque
* Define la interfaz entre el gestor de arranque y el firmware
* Expone información para los gestores de arranque con:
  + Información de hardware y configuración del firmware
  + Punteros a rutinas que implementan los servicios que el firmware ofrece a los bootloaders u otras aplicaciones UEFI
  + Provee un BootManager para cargar aplicaciones UEFI (e.j: Grub) y drivers desde un UEFI filesystem
  + El booloader ahora es un tipo de aplicaci\_on UEFI:
    - El Grub ser\_a una aplicaci\_on UEFI, que reside en el UEFI

filesystem donde est\_an los drivers necesarios para arrancar el sistema operativo (FAT32)

* Para el Grub deja de ser necesario el arranque en varias etapas.

1. **MBR** es el registro principal de arranque o registro de arranque maestro como también se conoce (Master boot record), es un sector de 512 bytes al principio del disco duro que contine una secuencia de comandos necesarios para cargar un sistema operativo. Es decir, es el primer registro del disco duro, el cual contiene un programa ejecutable y una tabla donde están definidas las particiones del disco duro.

**MBC** (master boot code) es el código de inicio maestro es una de las varias partes del master boot record (registro de inicio maestro). Realiza el primer conjunto de funciones importantes en el proceso de arranque. En el registro de inicio maestro genérico, el código de master boot code consume 446 bytes del registro de inicio maestro total de los 412 bytes; el espacio restante es utilizado por la tabla de particiones (64 bytes) y la firma de disco de 2 bytes

1. Utiliza el sistema **GPT** (GUID partition table) para solucionar limitaciones del MBR, como la cantidad de particiones GPT especifica la ubicación y formato de la tabla de particiones en un disco duro.
2. El **gestor de arranque** es un pequeño programa que es capaz de cargar el sistema operativo propiamente y, finalmente, transferirle el control. El sistema se autoiniciará y puede cargar los controladores de dispositivos y otros programas que son necesarios para el normal funcionamiento del sistema operativo.

Existen varios gestores de arranque (boot loaders) disponibles para sistemas operativos, y la elección de uno en particular depende de la configuración y preferencias, gestores de arranque más conocidos:

* GRUB (GRand Unified Bootloader): GRUB es uno de los gestores de arranque más populares en sistemas basados en Linux. Es versátil y compatible con múltiples sistemas operativos. GRUB permite seleccionar entre diferentes sistemas operativos instalados en la misma computadora y ofrece una interfaz de línea de comandos para realizar configuraciones avanzadas.
* GRUB2: Es una versión más reciente y mejorada de GRUB. GRUB2 es el gestor de arranque predeterminado en muchas distribuciones de Linux modernas y admite características como el soporte para unidades GPT y la autenticación segura (Secure Boot).
* LILO (LInux LOader): LILO fue un gestor de arranque ampliamente utilizado en sistemas Linux en el pasado, pero ha sido reemplazado en gran medida por GRUB. Aún puede encontrarse en sistemas más antiguos.
* Syslinux: Syslinux es un gestor de arranque ligero y versátil diseñado principalmente para sistemas Linux. Se utiliza comúnmente en medios de arranque USB y CD/DVD.
* Windows Boot Manager: Este gestor de arranque es utilizado por sistemas Windows y permite seleccionar entre diferentes versiones de Windows instaladas en la misma computadora. También puede arrancar otros sistemas operativos si están configurados correctamente.
* rEFInd: Este gestor de arranque está diseñado para sistemas UEFI y es especialmente popular en entornos de doble arranque (dual boot) con macOS y Linux. Proporciona una interfaz gráfica elegante y es altamente configurable.
* Clover: Clover es un gestor de arranque específicamente diseñado para sistemas Hackintosh, que son computadoras no Apple que ejecutan macOS. Facilita el arranque de macOS en hardware no oficial de Apple.
* Boot Camp: Es el gestor de arranque incluido en macOS que permite a los usuarios de Mac seleccionar entre macOS y Windows en computadoras con procesadores Intel.
* rEFIt: Antecesor de rEFInd, rEFIt es un gestor de arranque para Mac que permite arrancar múltiples sistemas operativos en computadoras con hardware Intel.

1. El proceso de inicio de una maquina y carga del sistemaoperativo y se denomina Bootstrap. En las arquitecturas x86, el BIOS (Basic I/O System) es el responsable de iniciar la carga del SO a través del MBC:

* Esta grabado en un chip (ROM, NVRAM)
* En otras arquitecturas también existe, pero se lo conoce con otro nombre: Power on Reset + IPL en mainframe o OBP (OpenBoot PROM): en SPARC

Cuando enciende la computadora, el programa de booteo en el MBR se carga y luego lanza el kernel del sistema operativo. El kernel inicia, configura los dispositivos y pasa el control al proceso init. Este proceso de arranque es una secuencia de pequeños programas que se ejecutan en cadena para poner en funcionamiento el sistema operativo.

1. El proceso de arranque en GNU/Linux consta de estos pasos principales:

* POST y Selección de Dispositivo de Arranque: La BIOS/UEFI realiza pruebas de hardware y selecciona el dispositivo de arranque.
* Carga del Gestor de Arranque: El gestor de arranque, como GRUB, se carga y permite seleccionar el sistema operativo.
* Carga del Kernel: El kernel de Linux se carga en la memoria.
* Init o systemd: Se inicia el proceso init o systemd, que gestiona el inicio de servicios.
* Montaje de Sistemas de Archivos: Se montan los sistemas de archivos.
* Ingreso al Modo Usuario: El sistema espera que los usuarios inicien sesión.
* Inicio de Sesión: Los usuarios inician sesión en sus cuentas.
* Una vez completado este proceso, el sistema GNU/Linux está listo para su uso.

1. El proceso de parada (shutdown) en GNU/Linux es un proceso ordenado que apaga el sistema de manera segura. Aquí están los principales pasos que se suceden en el proceso de parada de GNU/Linux:

* **Envío de Señales de Terminación**: El sistema operativo envía señales de terminación (como SIGTERM) a todos los procesos en ejecución, notificándoles que deben finalizar sus tareas y guardar cualquier trabajo pendiente.
* **Espera de Procesos**: El sistema espera un tiempo para que los procesos respondan a las señales de terminación. Los procesos tienen la oportunidad de realizar operaciones de cierre adecuadas, como guardar archivos y finalizar conexiones de red.
* **Envío de Señal de Finalización Forzada**: Si un proceso no responde a la señal de terminación o si está bloqueado, el sistema puede enviar una señal de finalización forzada (SIGKILL) para detenerlo de manera inmediata.
* **Desmontaje de Sistemas de Archivos**: Se desmontan los sistemas de archivos montados, asegurando que todos los datos se hayan escrito en los dispositivos de almacenamiento y que no haya datos en búfer pendientes de escritura.
* **Apagado de Dispositivos**: El sistema apaga todos los dispositivos hardware de manera controlada, como unidades de disco, interfaces de red y otros periféricos.
* **Cierre de Sesiones de Usuarios**: Se cierran las sesiones de usuario activas. Los usuarios son notificados y se les pide que cierren sesión.
* **Finalización de Servicios y Procesos del Sistema**: Se detienen los servicios y procesos del sistema, como demonios y servicios en segundo plano.
* **Apagado del Kernel**: El kernel del sistema recibe la señal de apagado y procede a apagar el hardware de manera segura.
* **Apagado del Hardware**: Finalmente, el hardware se apaga completamente, lo que incluye el corte de energía o el apagado del sistema, dependiendo de la configuración.

Es importante llevar a cabo este proceso de parada de manera ordenada para evitar la pérdida de datos o daños en el sistema de archivos. La secuencia exacta de estos pasos puede variar ligeramente según la distribución de Linux y la configuración del sistema, pero en general, se busca garantizar que el sistema se apague de manera segura y que todos los procesos y datos estén en un estado coherente.

1. Es posible tener instalados en una PC tanto GNU/Linux como otro sistema operativo, como Windows o macOS. Esta configuración se conoce como "dual boot" o "arranque dual".

9. Archivos

a) En GNU/Linux, los archivos se identifican por su nombre y ubicación en el sistema de archivos. Cada archivo tiene un nombre único y se encuentra en una ruta específica. Los permisos y el propietario también son parte de su identificación.

* 1. **Editores y comandos:**
* vi (Vim): vi es un editor de texto en línea de comandos muy potente y versátil que se encuentra en la mayoría de las distribuciones de Linux. Se utiliza para editar archivos de texto en la terminal. Vim, una versión mejorada de vi, incluye muchas funciones adicionales, como resaltado de sintaxis, autocompletado y una amplia gama de atajos de teclado. Para usarlo, simplemente escribe vi seguido del nombre del archivo que deseas editar. Puedes entrar en el modo de edición presionando la tecla i, realizar tus cambios y luego guardar y salir presionando Esc y escribiendo :wq para guardar y salir.
* mcedit (Editor de Midnight Commander): mcedit es el editor de texto integrado en Midnight Commander, un administrador de archivos para la terminal. Ofrece una interfaz de edición de texto similar a un editor de texto de ventana única con funciones básicas como cortar, copiar, pegar y búsqueda. Para abrir un archivo en mcedit, simplemente escribe mcedit seguido del nombre del archivo.
* cat (Concatenate): El comando cat se utiliza para mostrar el contenido de archivos de texto en la terminal. Por ejemplo, para mostrar el contenido de un archivo llamado archivo.txt, puedes escribir cat archivo.txt. cat también se utiliza para concatenar (unir) archivos y redireccionar la salida a otro archivo o tubería.
* more: El comando more se utiliza para paginar la salida de archivos de texto en la terminal, lo que permite ver el contenido uno por uno en lugar de mostrar todo el contenido a la vez. Puedes usar las teclas de flecha hacia arriba y hacia abajo o la barra espaciadora para navegar a través del contenido. Para salir de more, simplemente presiona la tecla q.

10.

a) Crear la carpeta ISO2017:

Comando: mkdir ISO2017

Función: Crea un nuevo directorio con el nombre "ISO2017".

b) Acceder a la carpeta (cd):

Comando: cd ISO2017

Función: Cambia el directorio de trabajo actual al directorio "ISO2017".

c) Crear dos archivos con los nombres iso2017-1 e iso2017-2:

Comando: touch iso2017-1 iso2017-2

Función: Crea dos archivos vacíos con los nombres especificados.

d) Listar el contenido del directorio actual:

Comando: ls

Función: Lista los archivos y directorios en el directorio actual.

e) Visualizar la ruta donde estoy situado:

Comando: pwd

Función: Muestra la ruta completa del directorio actual.

f) Buscar todos los archivos en los que su nombre contiene la cadena "iso":\*

Comando: find . -name 'iso\*'

Función: Busca todos los archivos cuyos nombres contienen "iso" seguido de cualquier número de caracteres en el directorio actual y sus subdirectorios.

g) Informar la cantidad de espacio libre en disco:

Comando: df

Función: Muestra información sobre el espacio en disco utilizado y disponible en todas las particiones montadas.

h) Verificar los usuarios conectados al sistema:

Comando: who

Función: Muestra una lista de los usuarios que están actualmente conectados al sistema.

i) Acceder al archivo iso2017-1 e ingresar Nombre y Apellido:

Comando: nano iso2017-1 o vi iso2017-1 (usa el editor de tu elección)

Función: Abre el archivo "iso2017-1" en un editor de texto para que puedas ingresar el Nombre y Apellido.

j) Mostrar en pantalla las últimas líneas de un archivo (tail):

Comando: tail nombre\_del\_archivo

Función: Muestra las últimas líneas de un archivo de texto. Puedes usar tail seguido del nombre del archivo que deseas ver. Por ejemplo, tail iso2017-1 mostrará las últimas líneas del archivo "iso2017-1".

11.

a) shutdown: Este comando se utiliza para apagar o reiniciar el sistema. Los archivos ejecutables de "shutdown" suelen encontrarse en /sbin/ o /usr/sbin/.

b) reboot: Es un comando que reinicia el sistema. El archivo ejecutable de "reboot" también suele estar en /sbin/.

c) halt: El comando "halt" se utiliza para apagar el sistema. El archivo ejecutable de "halt" se encuentra en /sbin/.

d) locate: "locate" se usa para buscar archivos en el sistema. El ejecutable y la base de datos de "locate" suelen estar en /usr/bin/.

e) uname: Este comando proporciona información del sistema, como el nombre del kernel y la versión. El ejecutable "uname" se encuentra en /bin/.

f) dmesg: "dmesg" muestra mensajes del kernel que se han registrado durante el arranque del sistema y mientras está en funcionamiento. El ejecutable "dmesg" suele estar en /bin/.

g) lspci: El comando "lspci" muestra información sobre los dispositivos PCI en el sistema. El ejecutable "lspci" se encuentra en /usr/bin/.

h) at: "at" se usa para programar tareas para que se ejecuten en un momento específico. El ejecutable "at" se encuentra en /usr/bin/.

i) netstat: Muestra información sobre conexiones de red, tablas de enrutamiento, estadísticas de interfaz, entre otros. El ejecutable "netstat" suele estar en /bin/.

j) mount: "mount" se utiliza para montar sistemas de archivos en directorios específicos. El ejecutable "mount" se encuentra en /bin/.

k) umount: "umount" se utiliza para desmontar sistemas de archivos previamente montados. El ejecutable "umount" suele estar en /bin/.

l) head: "head" muestra las primeras líneas de un archivo de texto. El ejecutable "head" se encuentra en /usr/bin/.

m) losetup: "losetup" se utiliza para configurar y administrar dispositivos loopback, que permiten el montaje de archivos como dispositivos de bloque. El ejecutable "losetup" suele estar en /sbin/.

n) write: "write" se utiliza para enviar mensajes a otros usuarios que están conectados al sistema. El ejecutable "write" suele estar en /usr/bin/.

ñ) mkfs: "mkfs" se utiliza para crear sistemas de archivos en dispositivos de bloque. El ejecutable "mkfs" suele estar en /sbin/.

o) fdisk: "fdisk" se utiliza para gestionar las particiones de los discos duros. El ejecutable "fdisk" se encuentra en /sbin/.