

*Práctica 1.*

*Juan Cruz de Urquiza.*

*Introducción a los Sistemas Operativos.*

*2022.*

*Facultad de Informática, Universidad Nacional de La Plata.*

**1.**

- a. GNU/Linux, es el término que se debe emplear cuando queremos hacer referencia al SO GNU con el kernel Linux. Es un sistema operativo tipo Unix, pero libre, diseñado por miles de programadores, gratuito y de libre distribución.  
Existen diversas distribuciones (customizaciones) y es código abierto, lo que nos permite estudiarlo, personalizarlo, auditarlo, aprovecharnos de la documentación, etc...
- b. Otro sistema operativo existente es Windows, el cual es un sistema operativo de código cerrado, diseñado para uso comercial y personal, creado por Microsoft. Linux es un kernel de código abierto que admite una variedad de sistemas. Windows es muy estable y competente, además de estar ya consolidado en el mercado, siendo “estándar” para varias aplicaciones desarrolladas. Por otro lado, Linux es un sistema gratuito, personalizable y más seguro. Además, representa una gran cuota del mercado en lo que se refiere a servidores.  
Mac OS se basa en una base de código BSD, mientras que Linux es un desarrollo independiente de un sistema similar a Unix. Esto significa que estos sistemas son similares, pero no compatibles binarios. Además, Mac OS tiene muchas aplicaciones que no son de código abierto y se basan en bibliotecas que no son de código abierto.
- c. El sistema GNU es el sistema operativo similar a Unix, constituido en su totalidad por software libre. Fue diseñado para ser totalmente compatible con UNIX. El hecho de ser compatible con la arquitectura UNIX implica que GNU esté compuesto de pequeñas piezas individuales de software, muchas de las cuales ya estaban disponibles, que pudieron ser adaptadas y reutilizadas; otras en cambio tuvieron que ser reescritas. Consiste en paquetes de GNU (programas publicados específicamente por el proyecto GNU) además de software libre publicado por terceras partes.
- d. El sistema operativo GNU fue iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre. Para asegurar que el mismo fuera libre, se necesitó crear un marco regulatorio conocido como GPL (General Public License de GNU). En 1985, Stallman crea la FSF (Free Software Foundation), con el fin de financiar el proyecto GNU. En 1990, GNU ya contaba con un editor de textos (Emacs), un compilador (GCC) y gran cantidad de bibliotecas que componen un Unix típico, aunque faltaba el componente principal: el núcleo (Kernel). Si bien ya se venía trabajando en un núcleo conocido como TRIX, es

en 1988 que se decide abandonarlo debido a su complejidad (corría en hardware muy costoso). En este momento se decide adoptar como base el núcleo MACH para crear GNU Hurd, el cual tampoco prosperó. Desde 1991, Linus Torvalds ya venía trabajando en un Kernel denominado Linux, el cual se distribuiría bajo licencia GPL. En el año 1992, Torvalds y Stallman deciden fusionar ambos proyectos, y es allí donde nace GNU/Linux.

- e. La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- f. POSIX significa Portable Operating System Interface. Consiste en una familia de estándares especificadas por la IEEE con el objetivo de facilitar la interoperabilidad de sistemas operativos. Además, POSIX establece las reglas para la portabilidad de programas. Por ejemplo, cuando se desarrolla software que cumple con los estándares POSIX existe una gran probabilidad de que se pueda utilizar en sistemas operativos del tipo Unix. Si se ignoran tales reglas, es muy posible que el programa o librería funcione bien en un sistema dado pero que no lo haga en otro.

## 2.

- a. Una distribución GNU/Linux es un conjunto de aplicaciones reunidas que permiten brindar mejoras para instalar fácilmente un sistema operativo basado en GNU/Linux. Son "sabores" de GNU/Linux que, en general, se diferencian entre sí por las herramientas para configuración y sistemas de administración de paquetes de software para instalar. La elección de una distribución depende de las necesidades del usuario y de gustos personales. Existen numerosas distribuciones Linux. Cada una de ellas puede incluir cualquier cantidad de software adicional (libre o no), como algunos que facilitan la instalación del sistema y una enorme variedad de aplicaciones. La base de cada distribución incluye el núcleo Linux, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software. Existe una infinidad de distribuciones, con características propias. Algunas distribuciones conocidas de GNU/Linux son: Debian, Gentoo, Red Hat Linux, Slackware. Las diferencias entre las distribuciones están en la licencia que usan, en su modelo de desarrollo, en la cuota de uso o en aspectos importantes para los usuarios como el soporte del hardware.
- b. Las distribuciones GNU/Linux están compuestas por multitudes de programas de usuario que forman el sistema operativo completo y lo hacen usable, estos programas son muchos de los programados bajo

los proyectos de la fundación GNU. Este es el motivo de que las distribuciones GNU/Linux incluyan los términos de ambos proyectos. Las distribuciones GNU/Linux son una recopilación de programas de terceras partes, todos estos programas GNU y no GNU incluidos en los sistemas GNU/Linux están desarrollados por diferentes programadores ajenos a Linux y los creadores de la distribución. Los creadores de las distribuciones GNU/Linux son los encargados de juntar los diferentes programas de cada fuente y empaquetarlas como un sistema usable, estable y libre de errores.

- c. El Proyecto Debian es una asociación de personas que han hecho causa común para crear un sistema operativo libre llamado Debian, los cuales usan el núcleo de Linux. Viene con más de 59.000 paquetes (software precompilado y empaquetado en un formato amigable para una instalación sencilla en su máquina), un gestor de paquetes, y otras utilidades que hacen posible gestionar miles de paquetes en miles de ordenadores de manera tan fácil como instalar una sola aplicación. Todos ellos de forma gratuita.

Es un poco como una torre. En la base está el núcleo. Encima se encuentran todas las herramientas básicas. Después está todo el software que se ejecuta en su computadora. En la cima de la torre se encuentra Debian — organizando y encajando todo cuidadosamente para que todo el sistema trabaje junto.

El 16 de agosto de 1993 fundó oficialmente el proyecto Debian con un mensaje dirigido a desarrolladores que trabajaban en Linux. Con esta filosofía, el proyecto comenzó a dar sus primeros pasos aunque, eso sí, se avanzase de manera muy lenta. Entre 1994 y 1995, el Proyecto Debian comenzó a lanzar sus primeras versiones 0.9x además de modelar la filosofía del proyecto. Desde noviembre de 1994 a noviembre de 1995, la Free Software Foundation (a través del proyecto GNU) patrocinó la creación de Debian y dio el primer gran impulso y apoyo al proyecto; un año en el que Debian comenzó a tomar forma y captar colaboradores. En esta primera etapa, Ian Murdock ejercería de Líder del Proyecto Debian hasta que pasó el testigo a Bruce Perens en abril de 1996. La primera versión de Debian (la 1.1, con nombre en clave Buzz) se lanzó el 17 de junio de 1996. El modelo organizativo de Debian se hizo para que todo el proceso de desarrollo, mantenimiento y soporte se realizase con sumo cuidado; un detalle que hizo que se crease una comunidad muy disciplinada y, sobre todo, bien organizada a nivel mundial. Si bien el modelo definía muy bien las tareas, Debian estaba abierta a todo tipo de colaboraciones y, en la práctica, es de los pocos proyectos de distribución que no están cimentados sobre una empresa o entidad comercial.

- a. Cuando hablamos de un SO GNU hacemos referencia a tres elementos fundamentales:
- El kernel (núcleo).
  - El Shell (intérprete de comandos).
  - El FileSystem (sistema de archivos)
- b. El kernel (también conocido como núcleo) es la parte fundamental de un sistema operativo. Se podría definir como el corazón de este sistema operativo. Es a grandes rasgos el encargado de que el software y el hardware de una computadora puedan trabajar juntos. El Shell (intérprete de comandos) es el programa que recibe lo que se escribe en la terminal y lo convierte en instrucciones para el sistema operativo. Un intérprete de comandos es un programa que lee las entradas del usuario y las traduce a instrucciones que el sistema es capaz de entender y utilizar. El Filesystem se traduce como “sistema de archivos”; y es la forma en que dentro de un SO se organizan y se administran los archivos.

#### 4.

- a. El kernel de Linux es el elemento principal de los sistemas operativos Linux, y es la interfaz fundamental entre el hardware de una computadora y sus procesos. Los comunica entre sí y gestiona los recursos de la manera más eficiente posible. Se llama kernel porque se encuentra dentro del sistema operativo, al igual que las semillas de las frutas con cáscara dura, y controla todas las funciones principales del hardware, ya sea un teléfono, una computadora portátil, un servidor o cualquier otro tipo de equipo. La primera versión del Kernel de Linux fue la 0.01, lanzada en 1991 y la primera versión estable fue la 1.0.0. De ahí en adelante ha seguido su evolución con la ayuda de miles de desarrolladores alrededor del mundo y hasta ahora continúa en las versiones 4.
- b. El kernel cumple cuatro tareas:
- 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.

Si se implementa de forma correcta, es invisible para el usuario y funciona en su propio mundo pequeño, conocido como el espacio del kernel, donde asigna la memoria y supervisa el lugar en el que se almacena cada elemento. Lo que está a la vista del usuario, como los

exploradores y los archivos, se conoce como el espacio del usuario. Estas aplicaciones interactúan con el kernel a través de una interfaz de llamadas al sistema.

El kernel es el asistente personal atareado de un poderoso ejecutivo (el hardware). Debe transmitir los mensajes y las solicitudes (los procesos) de los empleados y el público (los usuarios) al ejecutivo, recordar qué elementos se almacenan y dónde (la memoria) y determinar quién puede acceder al ejecutivo en un momento dado y durante cuánto tiempo.

- c. La versión más actual de GNU/Linux es la 5.18.1, lanzada en mayo de 2022.

Antes de la serie de Linux 2.6.x, los números pares indicaban la versión “estable” lanzada. Por ejemplo una para uso de fabricación, como el 1.2, 2.4 ó 2.6. Los números impares, en cambio, como la serie 2.5.x, son versiones de desarrollo, es decir que no son consideradas de producción. Comenzando con la serie Linux 2.6.x, no hay gran diferencia entre los números pares o impares con respecto a las nuevas herramientas desarrolladas en la misma serie del kernel. Linus Torvalds dictaminó que éste sería el modelo en el futuro.

- d. En un sistema GNU/Linux, se podrían tener instalados 2 kernel con versiones diferentes, pero no podrían ejecutarse los 2 al mismo tiempo, uno estaría funcionando y el otro estaría simplemente almacenado en el disco duro.

Se pueden simular máquinas virtuales con sistemas operativos GNU/Linux, los cuales tendrían kernels, pero serían virtuales y no a nivel hardware.

- e. Como el kernel en realidad es código, tiene que estar almacenado dentro del sistema de archivos, para que cada vez que el sistema sea prendido, se cargue en memoria. En los sistemas con Debian/Ubuntu, el kernel puede ser encontrado en el directorio /boot.
- f. El kernel de GNU/Linux es monolítico, porque los controladores de dispositivos y las extensiones del núcleo normalmente se ejecutan en un espacio privilegiado, con acceso irrestricto al hardware, aunque algunos se ejecutan en espacio de usuario. Lo que lo hace híbrido es la capacidad de cargar y descargar funcionalidad a través de módulos.

## 5.

- a. El shell (-también conocido como el intérprete de comandos línea de comandos, terminal, consola-) es un programa que actúa como interfaz, para comunicar al usuario con el sistema operativo mediante una ventana que espera comandos textuales ingresados por el usuario en el teclado, los interpreta y los entrega al SO para su ejecución. La respuesta del SO es mostrada al usuario en la misma ventana. A continuación, la shell queda esperando más instrucciones. Se

interactúa con la información de la manera más simple posible, sin gráficas, solo el texto.

- b. Su función es ser el puente de comunicación entre el usuario usando el sistema operativo y la máquina. Gracias al shell, el usuario podrá ingresar comandos que provocarán cambios dentro del sistema operativo al ser ejecutados.
- c. Existen diferentes tipos de Shells en GNU/Linux, los cuales son: Korn Shell (pdksh, zsh), Bourne Shell (bash) y C Shell (tcsh). Estas se diferencian entre sí en la sintaxis de sus comandos y en la interacción con el usuario.
- d. Un comando externo es un archivo localizado en el árbol del sistema de archivos. Por ejemplo, cuando un usuario ejecuta el comando `ls`, el Shell pide al núcleo cargar en memoria el archivo `/usr/bin/ls`. Consideramos comandos externos los archivos que tengan uno de los formatos siguientes:
  - Archivos con formato binario ejecutable.
  - Archivos con formato de texto que representan un script de comandos (que puede estar escrito en Shell o en otro lenguaje).Un comando interno se integra en los procesos Shell (el Shell ejecuta la acción). Por lo tanto, no se corresponde en ningún caso a un archivo almacenado en disco. El comando interno se ejecuta por el Shell actual.
- e. La shell no forma parte del kernel básico del SO; sino que la misma “dialoga” con el kernel. Es iniciada por un proceso denominado “login”, y dado que cada usuario tiene asignado una shell por defecto, se inicia cada vez que un usuario comienza a trabajar en su estación de trabajo (es decir se “loguea” en una terminal). Dentro del contenido del archivo `/etc/passwd`, se puede ver cual es la shell que cada usuario tiene asignada por defecto.
- f. Sí, cada usuario puede tener un intérprete de comandos diferente, incluso un mismo usuario puede ir cambiando entre la terminal de su preferencia. Al instalar un sistema GNU/Linux en Oracle por ejemplo, a cada usuario se le asigna una terminal bash por defecto.

## 6.

- a. El sistema de archivos o file system es la forma en que dentro de un sistema de cómputo se organizan y administran los archivos. Esa administración comprende:
  - Métodos de acceso: cómo se acceden los datos contenidos en el archivo.
  - Manejo de archivos: cómo actúan los mecanismos para almacenar, referenciar, compartir y proteger los archivos.
  - Manejo de la memoria secundaria: Cómo se administra el espacio para los archivos en memoria secundaria.

- Mecanismos de integridad: con qué métodos se garantiza la incorruptibilidad del archivo.

La mayoría de los sistemas operativos poseen su propio sistema de archivos, los cuales estructuran la información que luego será representada, ya sea textual o gráficamente, utilizando un gestor de archivos.

- b. Hoy en día existen varios sistemas de archivos soportados por GNU/Linux, algunos de ellos son ext2, ext3, ReiserFs o XFS.
- c. Sí, es posible realizar particiones del disco duro las cuales posean como sistemas de archivos a FAT y NTFS.
- d. En el momento de instalación, GNU/Linux crea una estructura de directorios básica, definida por la Filesystem Hierarchy Standard Group llamada FHS, que define los directorios principales y sus contenidos en los sistemas operativos de la familia Unix. Se diseñó originalmente en 1994 para estandarizar el sistema de archivos de las distribuciones existentes. Todos los archivos y directorios aparecen bajo el directorio raíz /, aunque se encuentren en distintos dispositivos físicos.

La estructura básica del file system de FHS está dada por los directorios:

- /: Tope de la estructura de directorios. Es como el C:\ en Windows.
- Bin: Comandos binarios esenciales. Archivos binarios y ejecutables.
- Dev: Archivos y enlaces de dispositivos.
- Etc: Archivos de configuración del sistema específicos.
- Usr: Estructura secundaria. Aplicaciones de usuarios.
- Var: Datos variables de tamaño.
- Home: Directorio base que almacena archivos de usuarios.

## 7.

- a. Una partición de disco es el nombre que recibe cada división presente en una sola unidad física de almacenamiento de datos. Toda partición tiene su propio sistema de archivos. Todo sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico. Las particiones pueden utilizarse para varios fines: por una parte, se puede tener una dedicada a guardar datos sensibles con medidas de seguridad que no interfieran en el resto del sistema, así como copias de seguridad, y también pueden utilizarse para instalar diferentes sistemas operativos. En algunos de ellos, como los basados en GNU/Linux, se puede estructurar el disco en particiones para los diferentes tipos de archivo que utilice el sistema operativo. Es una división lógica en una unidad de almacenamiento, en la cual se alojan

y organizan los archivos mediante un sistema de archivos. Existen distintos esquemas de particiones para la distribución de particiones en un disco, los más conocidos y difundidos son MBR (Master Boot Record) y GPT (GUID Partition Table).

Hay 3 tipos diferentes de particiones:

- Partición primaria: Son las divisiones crudas o primarias del disco, solo puede haber 4 de éstas o 3 primarias y una extendida (depende de la tabla de particiones). A este tipo de particiones cualquier sistema operativo puede detectarlas y asignarles una unidad, siempre y cuando el sistema operativo reconozca su sistema de archivos.
- Partición extendida: También conocida como partición secundaria, es otro tipo de partición que actúa como una partición primaria; sirve para contener múltiples unidades lógicas en su interior. Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico. Solo puede existir una partición de este tipo por disco, y solo sirve para contener particiones lógicas. Por lo tanto, es el único tipo de partición que no soporta un sistema de archivos directamente.
- Partición lógica: Ocupa una porción de la partición extendida o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos (FAT32, NTFS, ext2,...) y se le ha asignado una unidad, así el sistema operativo reconoce las particiones lógicas o su sistema de archivos. Puede haber un máximo de 23 particiones lógicas en una partición extendida. Linux impone un máximo de 15, incluyendo las 4 primarias, en discos SCSI y en discos IDE 8963.

Particionar un disco duro puede tener tanto ventajas como desventajas.

Las ventajas principales son que se pueden tener distintos sistemas operativos en una misma unidad física de almacenamiento, se pueden generar copias de seguridad más simples de datos y archivos, proteger de malware a las otras particiones si una de ellas se ve afectada; se puede tener una mejor organización de archivos y documentos, e incluso utilizar diferentes sistemas de éstos en las distintas particiones.

Las desventajas fundamentales que existen al realizar particiones de disco duro es que las particiones no son invulnerables a ataques de malware o pérdida de datos, muchas veces al particionar se tiene un falso sentido de seguridad que todos los datos estén respaldados de forma segura cuando no es así. También, es bastante complejo realizar los particionados de forma adecuada, y se puede llegar a perder bastante espacio de almacenamiento al crear particiones innecesarias. Sumando a esto, tener la unidad de almacenamiento



principal particionada generalmente es innecesario para el usuario promedio.

- b. Las particiones dentro de GNU/Linux pueden ser identificadas de diferentes formas de acuerdo a la distribución usada. Generalmente, se encuentran dentro de la carpeta /dev, teniendo diferentes identificadores. Luego de /dev viene el identificador para las particiones, que empieza por /sd. A esto le sigue una letra del abecedario que identifica el disco, y un número que identifica la partición. De tal manera que, el primer disco duro siempre será /sda, y las particiones /sda1,/sda2, /sda3, el segundo disco será /sdb, y las particiones /sdb1,/sdb2,/sdb3, y así.

- c. Como mínimo se necesitan 2 particiones: una para / (que representa el sistema y los datos de éste) y otra para swap. De todas formas, es recomendable tener 3, agregando una partición para /home. Estas 3 particiones tienen que ser primarias, y el sistema de archivos que usan es el ext4 transaccional en términos generales.

La identificación para la partición que contenga / sería /dev/sda1, la partición que tiene el swap sería /dev/sda2 (esta no tiene sistema de archivos) y /dev/sda3 haría referencia a la partición que contenga a /home.

- d. Primer caso: Integración de nuevas adquisiciones.

Formación de una empresa nueva. La empresa recién formada no utiliza las mismas aplicaciones para las nóminas, el inventario y la facturación. Tiene previsto consolidar las dos empresas en un solo conjunto de aplicaciones, pero esta consolidación tardará un tiempo. Mientras tanto, debe reducir el coste de los centros de datos con rapidez. Por tanto, decide crear particiones lógicas para las aplicaciones utilizadas por la empresa recién adquirida. Instala un sistema operativo y las aplicaciones utilizadas por la empresa nueva en la partición lógica.

Segundo caso: Creación de varios entornos de cliente.

El usuario suministra servicios e-commerce de alta disponibilidad a diversos clientes. Proporciona recursos de sistema, aplicaciones y soporte técnico a cada cliente, y cada cliente puede configurar y utilizar independientemente las aplicaciones ejecutadas en los recursos de sistema suministrados. En un entorno de este tipo, es esencial aislar a los clientes para que sólo tengan acceso a sus recursos. Sin embargo, dedicar un servidor físico a cada cliente tiene un coste prohibitivo, y no permite aumentar ni disminuir fácilmente la cantidad de recursos de sistema utilizados por cada cliente. Por tanto, decide crear una partición lógica para cada cliente. Instala un sistema operativo y aplicaciones en cada partición lógica. A continuación, puede utilizar el particionamiento dinámico para añadir recursos a particiones lógicas o eliminar recursos de ellas según sea necesario.

Si un cliente deja de utilizar el servicio, puede suprimir la partición lógica de dicho cliente y reasignar los recursos a otras particiones lógicas.

- e. Para realizar particiones dentro de la unidad de almacenamiento principal de nuestro sistema, se utiliza software llamado particionador. Existen 2 tipos de particionadores: destructivos y no destructivos. Los particionadores destructivos permiten crear y eliminar particiones (fdisk es un ejemplo). Los particionadores no destructivos, además de crear y eliminar particiones permiten modificarlas (fips y gparted son ejemplos). Las distribuciones de GNU/Linux permiten al usuario hacer cambios en las particiones desde la interfaz de instalación del sistema operativo.

## 8.

- a. En las arquitecturas x86, el BIOS (Basic I/O System) es el responsable de iniciar la carga del sistema operativo a través del MBC (Master Boot Code del MBR, el cual es un sector reservado del disco físico ubicado en el cilindro 0, cabeza 0, sector 1). El BIOS carga el programa de booteo del sistema operativo desde el MBR y el gestor de arranque lanzado desde el MBC carga el kernel, el cual prueba y hace disponibles los dispositivos para después pasarle el control al proceso init. El proceso de arranque se ve como una serie de pequeños programas de ejecución encadenada.
- b. UEFI, por sus siglas en inglés Unified Extensible Firmware Interface, es una alianza entre varias compañías con el objetivo de modernizar el proceso de arranque, conformada por representantes de AMD, Apple, HP, Dell, IBM, Intel, Lenovo, Microsoft, entre otras empresas. Es propiedad del UEFI Forum, y aporta criptografía, autenticación de red y una interfaz gráfica. Define la ubicación del gestor de arranque y la interfaz entre éste y el firmware.
- c. El MBR, por sus siglas en inglés Master Boot Record, es el registro de arranque principal del sistema operativo, y es el primer sector de un dispositivo de almacenamiento de datos. El sector reservado que tiene, es el cilindro 0, sector 0, disco 1. Todos los discos duros tienen un MBR. El tamaño de éste coincide con el tamaño estándar de un sector: 512 bytes. En los primeros bytes está el Master Boot Code (código que utiliza la BIOS para arrancar el sistema). A partir del byte 446 está la tabla de particiones que es de 64 bytes. Los últimos 2 bytes corresponden a la firma. Es creado con algún utilitario.
- d. Las siglas GPT significan GUID partition table, y puede verse como una sustitución del MBR, aunque este se mantiene para tener la compatibilidad necesaria con el BIOS. El GPT usa un modo de direccionamiento lógico en vez del modo de direccionamiento

tradicional de cilindro, sector y disco. La cabecera GPT y la tabla de particiones están escritas al principio y al final del disco (hay redundancia).

- e. La funcionalidad de un gestor de arranque (bootloader en inglés) es la de cargar una imagen de Kernel (sistema operativo) de alguna partición para su ejecución. Se ejecuta luego del código del BIOS. Existen 2 modos de instalación: En el MBR o en el sector de arranque de la partición raíz o activa. Existen 2 tipos de bootloaders, el básico y el avanzado. Los bootloaders básicos sólo tienen una función: buscar una señal de arranque (cadena de bits) y cargarla. Los bootloaders avanzados tienen la capacidad de dejar elegir al usuario modos de operación a llevar a cabo y ejecutar diagnósticos de bajo nivel. Los gestores de arranque se encuentran en el MBR (solo la fase 1), y la fase 1.5 se encuentra en los siguientes 30 KB del disco (el llamado MBR gap). Los gestores de arranque para sistemas GNU/Linux más conocidos son: GRUB, LILO, NTLDR, GAG, YaST, entre otros.
- f. Para que un sistema operativo pueda ejecutarse, el sistema debe cargarse en la RAM de la computadora. Cuando una computadora se enciende por primera vez, lanza el bootstrap que reside en el chip Sistema Básico de Entrada y Salida (BIOS) o firmware. Las funciones principales del bootstrap son probar el hardware de la computadora y localizar y cargar el sistema operativo en la RAM. Puesto que el programa bootstrap está incorporado al chip BIOS, también se denomina control del BIOS. Durante la ejecución de las rutinas de firmware del BIOS, se llevan a cabo tres conjuntos de operaciones:
  - 1. Se ejecutan las autopruebas de Encendido (POSTs).
  - 2. Se completa la inicialización.
  - 3. El BIOS desplaza la dirección de inicio y la información de modo al controlador DMA, y luego carga el Registro de Inicio Maestro (MBR).
- g. El bootstrap es denominado el proceso de inicio de una máquina y carga del sistema operativo. El BIOS es el responsable de iniciar la carga del SO a través del MBR. Luego, se carga el programa de booteo (desde el MBR). El gestor de arranque lanzado desde el MBR carga el kernel. El proceso de arranque se ve como una serie de pequeños programas de ejecución encadenada.
- h. Si se ejecuta el proceso de shutdown en GNU/Linux, significa que la computadora va a apagarse, pero este proceso consta de varios pasos para poder realizarse de forma adecuada.

Cuando este proceso se inicia, todos los usuarios son notificados de que el sistema va a apagarse y las operaciones de inicio de sesión son bloqueadas. Todos los procesos que se estén llevando a cabo son notificados de que el sistema va a apagarse con una señal llamada SIGTERM. Esta señal les da a los programas tiempo para ser

guardados y terminados de forma limpia. El shutdown hace su trabajo señalando el proceso init, pidiéndole que cambie su runlevel (conjunto de procesos propios de Linux) al runlevel 0, el cual significa parada de sistema.

- i. Sí, es posible tener 2 sistemas operativos instalados en una misma computadora, teniendo GNU/Linux y Windows, por ejemplo. Esto es posible si se realiza una partición del disco duro de la computadora, donde en una de ellas se instale GNU/Linux y en otra el otro sistema operativo.

**9.**

- a. En GNU/Linux, todo es un archivo en sí mismo, y un archivo es un conjunto de información a la cual se le ha dado un nombre específico. Los nombres de los archivos están limitados a 256 caracteres, y no pueden contener el "/" en su nombre. Incluso los directorios son archivos, y la ruta para acceder a uno de ellos empieza siempre con /.

- b. Vi es un editor de texto de GNU/Linux. A diferencia de un procesador de texto, no ofrece herramientas para determinar visualmente cómo quedará el documento impreso, por esto carece de opciones como centrado o justificación de párrafos, pero permite mover, copiar, eliminar o insertar caracteres con mucha versatilidad. Es frecuentemente utilizado por programadores para escribir código fuente de software. Fue originalmente escrito por Bill Joy en 1976, tomando recursos de ed y ex, dos editores de texto para Unix, que trataban de crear y editar archivos, de ahí, la creación de vi.

Mcedit es un editor de texto independiente. Esta aplicación permite visualizar el contenido de ficheros y disfrutar de características como la de resaltar la sintaxis para ficheros de código fuente de ciertos lenguajes de programación, y la capacidad de trabajar tanto en modo ASCII como en modo Hexadecimal. Mcedit originalmente pertenece a Midnight Commander, el cual es un sistema de ficheros para sistemas Unix. Sin embargo, mcedit puede ser utilizado sin la necesidad de tener Midnight Commander, o puede ser reemplazado por otro editor de texto dentro del sistema de ficheros.

El comando cat es la abreviatura de concatenar. Esto se refiere al hecho de que cat puede ser utilizado para combinar múltiples archivos en un archivo, también se puede utilizar para crear nuevos archivos y para mostrar el contenido de los archivos existentes. Se utiliza a menudo como una forma sencilla de ver el contenido de un archivo de texto.

El comando more es un comando para ver pero no modificar el contenido de un archivo (o incluso un comando) y visualizarlo por páginas. Sólo permite la navegación hacia adelante y al pulsar una tecla, avanza hacia la siguiente página.

- c. *Resuelto en máquina.*

- d. El comando file lo que hace es informar al usuario el tipo de archivo seleccionado. Dependiendo del tipo de archivo referenciado el comando file mostrará en la terminal el tipo de archivo que se acaba de solicitar.

**10. Resuelto en máquina.**

**11.**

- a. Apaga el sistema. El tiempo para apagar el sistema es el parámetro.
- b. Reinicia el sistema. El tiempo para reiniciar el sistema es el parámetro.
- c. El comando halt detiene la CPU del ordenador. Los parámetros pueden ser para apagar las interfaces de red, escribir el registro wtmp, o no sincronizar antes de apagar.
- d. Busca un archivo en el directorio actual. Como parámetro va el nombre del archivo a buscar.
- e. Imprime información del sistema. El parámetro es para saber la opción a seleccionar (núcleo, versión, procesador).
- f. Escribe los mensajes del kernel. El parámetro -h es para que la información impresa sea más prolija, y el -w para que se pueda escribir un para filtrar los resultados.
- g. Lista todos los dispositivos PCI. Los parámetros son para cambiar el formato de muestra de la información.
- h. Sirve para programar un comando a cierta hora. Se pueden listar todos los trabajos programados y eliminarlos.
- i. Muestra las conexiones y los estados de la red. Con los parámetros se pueden ver programas, grupos y estadísticas.
- j. Monta un sistema de ficheros. Según los parámetros que se le ponga, realiza y escribe diferentemente los ficheros.
- k. Desmonta un sistema de ficheros. Según los parámetros que se le ponga, realiza y escribe diferentemente los ficheros.
- l. Muestra los 10 primeros caracteres de un archivo. Con los parámetros se puede personalizar para que imprima cierta cantidad de líneas, o imprima caracteres a excepción de algunos.
- m. Se utiliza para fijar un dispositivo de bucle. Los parámetros se utilizan para fijar el dispositivo, iniciar el cifrado y establecer el número de conversión de datos.
- n. Manda un mensaje a otro usuario del sistema. Tiene un parámetro para saber la versión de Linux.
- o. Se utiliza para dar formato a un dispositivo de almacenamiento de bloque con un determinado sistema de archivos. Con los parámetros se pueden gestionar las particiones, como eliminarlas y modificarlas.

**12.**

- a. /usr/sbin
- b. /usr/sbin
- c. /usr/sbin

- d.** No se encuentra en el sistema por defecto, hay que instalarlo por separado.
- e.** /bin
- f.** /bin
- g.** /bin
- h.** No se encuentra en el sistema por defecto, hay que instalarlo por separado.
- i.** No se encuentra en el sistema por defecto, hay que instalarlo por separado.
- j.** /bin
- k.** No se encuentra en el sistema por defecto, hay que instalarlo por separado.
- l.** /bin
- m.** No se encuentra en el sistema por defecto, hay que instalarlo por separado.
- n.** /bin
- o.** No se encuentra en el sistema por defecto, hay que instalarlo por separado.