

1a) GNU/Linux es un Sistema Operativo gratuito y de libre distribución, diseñado por miles de programadores y de código abierto, lo que nos permite estudiarlo y personalizarlo.

1b) Windows es un SO comercial, diseñado por la compañía Microsoft. No se distribuye libremente, ni con su código fuente, por lo que tiene un costo asociado y no está permitida su modificación.

1c) GNU (GNU is Not Unix) es un sistema operativo formado en su totalidad por software libre. Sigue un diseño tipo Unix y se mantiene compatible con este SO, pero se distingue de Unix por ser software libre y por no contener código de Unix.

1d) Historia de GNU: El proyecto fue iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre. Para asegurar que fuera libre, se necesitó crear un marco regulatorio, el GPL (General Public License de GNU). Stallman crea la fundación FSF (Free Software Foundation) para financiar el proyecto.

En 1990, GNU ya contaba con editor de textos (Emacs), compilador (GCC) y una gran cantidad de bibliotecas, pero faltaba el componente principal: el núcleo (Kernel).

Se venía trabajando en un núcleo (TRIX), pero se decide abandonarlo por su complejidad y costo. Luego, se decide adoptar el núcleo MACH para crear GNU Hurd, pero no prosperó.

Linus Torvalds venía trabajando desde 1991 en un Kernel denominado Linux, y en 1992, él y Stallman deciden fusionar ambos proyectos, y allí nace GNU/Linux.

1e) Un SO multitarea divide el tiempo de procesador disponible entre los procesos que lo necesitan.

Básicamente, permite que varios procesos se ejecuten al mismo tiempo, compartiendo uno o más procesadores. Los SO multitarea son capaces de dar servicio a más de un proceso a la vez para permitir la ejecución de muchos más programas. GNU/Linux es un SO multitarea.

1f) POSIX (Portable Operating System Interface UNIX) es una norma escrita por la IEEE. Define una interfaz estándar del SO y el entorno, incluyendo un intérprete de comandos (shell), y programas de utilidades comunes para apoyar la portabilidad de aplicaciones a nivel código fuente.

2ab) Una distribución de GNU/Linux es una customización de GNU/Linux formada por una versión de kernel y determinados programas con sus configuraciones.

2c) Debian es una distribución de GNU/Linux. Nació en 1993 de la mano del proyecto Debian, fundado por Ian Murdock.

La dedicación de Debian al software libre, su base de voluntarios, naturaleza no comercial y su modelo de desarrollo abierto lo distingue de otras distribuciones de GNU. Debian se caracteriza por tener el SO más estable posible, en lugar de las últimas novedades de GNU/Linux.

3a) Los 3 componentes fundamentales de GNU/Linux son: el núcleo (Kernel), el intérprete de comandos (Shell), y el File System (Sistema de Archivos).

3b)

- Kernel: El Kernel es el núcleo del SO. Es el encargado de que el software y el hardware puedan trabajar juntos, pues ejecuta programas, y gestiona los recursos del sistema y dispositivos de hardware. Sus funciones más importantes son la administración de la memoria, del tiempo de CPU que los programas y procesos utilizan, y de que como usuarios podamos acceder a los dispositivos de E/S de forma cómoda.
- Shell: El Shell, o intérprete de comandos, es el modo de comunicación entre el usuario y el SO. Gracias a él podemos dar órdenes para que nuestro SO realice las tareas que necesitamos. No debemos confundirlo con el SO, pues este es solo un programa que hace de mediador entre el usuario y el Kernel del sistema. El Shell puede ser tanto gráfico (Windows) como de texto (Bash). Cada usuario puede tener una shell, se pueden personalizar y son programables.
- File System: Los Sistemas de Archivos estructuran la información guardada en una unidad de almacenamiento (normalmente un disco duro) de una computadora, que luego será representada ya sea textual o gráficamente por medio de un gestor de archivos. La mayoría de SO tienen su propio File System.

4a) El Kernel es el núcleo del SO. Es el encargado de que el software y el hardware puedan trabajar juntos, pues ejecuta programas, y gestiona los recursos del sistema y dispositivos de hardware.

4b) Sus funciones más importantes son la administración de la memoria, del tiempo de CPU que los programas y procesos utilizan, y de que como usuarios podamos acceder a los dispositivos de E/S de forma cómoda.

4c) La última versión estable es la 6.4.13 (30-08-2023). Se definía como A.B.C.D

- A: Denota versión. Cambia con menor frecuencia.
- B: Denota mayor revisión.
- C: Denota menor versión. Solo cambia cuando hay nuevas características.
- D: Cambia cuando se corrige un grave error sin agregar nueva funcionalidad.

Antes de la versión 2.6, los números impares B indicaban desarrollo, y los impares producción. Luego dejó de representar eso.

4d) Solo es posible tener un único Kernel de GNU/Linux instalado en una máquina.

4e) El Kernel reside en memoria principal como parte del proceso que se esté ejecutando.

4f) El Kernel es un núcleo monolítico híbrido. Todo lo que necesita para trabajar está en el mismo Kernel. El código del Kernel se ejecuta en modo privilegiado, y lo que lo hace híbrido es la capacidad de cargar y descargar funcionalidad a través de módulos.

5a) El Shell, o intérprete de comandos, es el modo de comunicación entre el usuario y el SO. Gracias a él podemos dar órdenes para que nuestro SO realice las tareas que necesitamos.

5b) Algunas de sus funciones son la ejecución de programas a partir del ingreso de comandos, control de procesos, re-dirección de E/S, listado y lectura de ficheros, protección, comunicaciones y un lenguaje de órdenes para escribir programas por lotes.

5c)

- Bourne Shell (sh): Usa sintaxis de comandos de los primeros sistemas UNIX. Se lo encuentra como `"/bin/sh"`.
- C Shell (csh): Usa sintaxis similar al lenguaje C. Se lo encuentra como `"/bin/csh"`.
- Bourne Again Shell (bash): El más usado en el mundo de GNU/Linux. Se lo encuentra como `"/bin/bash"`.
- Tsch: Versión extendida de C Shell. Se lo encuentra como `"/bin/tcsh"`.

5d) `"/bin/shell"`.

5e) Porque el Shell es solo un programa que hará de mediador entre nosotros y el kernel del sistema.

5f) Los gustos de los usuarios son los que deciden qué intérprete de comandos se usará, por esto es posible configurar para cada usuario un intérprete de comandos

distintos y no afectar el funcionamiento de programas como el ls, cat o cp.

6a) Los Sistemas de Archivos organizan la forma en que se almacenan los archivos en dispositivos de almacenamiento.

6b) GNU/Linux soporta una gran cantidad de tipos de sistema de archivos: adfs, affs, autofs, coda, coherent, cramfs, devpts, efs, ext2, ext3, hfs, hpfs, iso9660, jfs, minix, msdos, ncpfs, nfs, ntfs, proc, qnx4, reiserfs, romfs, smbfs, sysv, tmpfs, udf, ufs, umsdos, vfat, xenix, xfs.

6c) Si. En estos momentos existen distribuciones de GNU/Linux que pueden realizar operaciones de lectura y escritura sobre ellas.

6d) En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada Ruta. La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una barra ("/") o barra invertida ("\") y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

Directorios más importantes:

- "/": Es la raíz del sistema de directorios. Aquí se monta la partición principal de Linux EXT.

- `/etc`: Contiene archivos de configuración de la mayoría de programas.
- `/home`: Contiene los archivos personales de los usuarios.
- `/bin`: Contiene comandos básicos y muchos programas.
- `/sbin`: Comandos administrativos.
- `/dev`: Contiene archivos simbólicos que representan partes del hardware, tales como discos duros, memoria, etc...
- `/mnt`: Contiene subdirectorios donde se montan otras particiones de disco duro, CD-ROMs, etc...
- `/tmp`: Ficheros temporales o de recursos de programas.
- `/usr`: Programas y librerías instalados con la distribución.
- `/usr/local`: Programas y librerías instalados por el administrador.
- `/lib`: Librerías varias y módulos del kernel.
- `/var`: Datos varios como archivos de log de programas, bases de datos, contenidos del servidor web, copias de seguridad, etc...
- `/proc`: Información temporal sobre los procesos del sistema.

El FHS (Filesystem Hierarchy Standard) define los directorios principales y sus contenidos en el sistema operativo Linux, GNU/Linux y otros sistemas de la familia Unix. Se diseñó originalmente en 1994 para estandarizar el sistema de archivos de las distribuciones GNU/Linux, basándose en la tradicional organización de directorios de los sistemas Unix.

7a) Particionar es una forma de dividir lógicamente el disco físico. Existen 3 tipos diferentes:

- Primaria: Son las divisiones primarias del disco. Solo pueden haber 4 de estas. Un disco físico completamente formateado consiste en una partición primaria que ocupa todo el espacio del disco, y posee un sistema de archivos (FAT32, NTFS, ext2...). Prácticamente cualquier SO puede detectarlas.
- Extendida: Sirve para contener infinidad de unidades lógicas en su interior. Fue ideada para romper con 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.
- Unidad Lógica: Ocupa un trozo de 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.

Ventajas:

- Se puede guardar una copia de seguridad de los datos del usuario en otra partición del mismo disco, para evitar la pérdida de información importante.
- Si tienes un error/problema en una de ellas, las demás no se verán afectadas.
- Puedes tener diferentes sistemas operativos en tu PC, totalmente independientes unos de otros.
- Puedes tener tus archivos de datos en particiones totalmente independientes.
- Puedes borrar/cambiar el contenido de una partición, sin que esto afecte a las demás.

7b)

- Discos duros (en general: /dev/hdx#, donde x es el disco y # es la partición):
 - Primer disco duro: (todo el disco) /dev/hda
 - Particiones primarias:
 - Primera partición primaria: /dev/hda1
 - Segunda partición primaria: /dev/hda2
 - Tercera partición primaria: /dev/hda3
 - Cuarta partición primaria: /dev/hda4
 - Particiones lógicas:
 - Primera partición lógica: /dev/hda5
 - Sucesivamente: /dev/hda#
 - Segundo disco duro: (todo el disco) /dev/hdb
 - Particiones primarias:
 - Primera partición primaria: /dev/hdb1
 - Segunda partición primaria: /dev/hdb2
 - Tercera partición primaria: /dev/hdb3
 - Cuarta partición primaria: /dev/hdb4
 - Particiones lógicas:
 - Primera partición lógica: /dev/hdb5
 - Sucesivamente: /dev/hdb#
- Discos SCSI
 - Primer disco SCSI: /dev/sda
 - Segundo disco SCSI: /dev/sdb
 - Y así sucesivamente...
- Primer CD-ROM SCSI: /dev/scd0, también conocido como /dev/sr0

7c) Para instalar GNU/Linux, es necesaria como mínimo 1 partición (para el "/"), pero es recomendable crear al menos 2,

para usar la segunda como partición SWAP, la cual es un espacio del disco duro utilizada por el sistema operativo como memoria virtual o almacenamiento temporal cuando no hay espacio suficiente en la memoria RAM.

7d)

- **partman:** Herramienta original de Linux para particionar discos. Esta «navaja suiza» también puede ajustar el tamaño de las particiones, crear sistemas de ficheros (como se llama en Windows a “formatear”) y asignarlos a sus respectivos puntos de montaje.
- **fdisk:** Es la herramienta original de Linux para particionar discos, buena para expertos. Sea cuidadoso si tiene una partición de FreeBSD en su máquina. Los núcleos instalados traen soporte para este tipo de partición, pero la manera en que fdisk la representa, puede (o no) ser un poco diferente. Para más información, revise el CÓMO de Linux+FreeBSD.
- **cdisk:** Una herramienta para particionar a pantalla completa, muy fácil de usar. Recomendada para la mayoría de los usuarios. cdisk no reconoce las particiones de FreeBSD, y nuevamente, los dispositivos mostrados en pantalla pueden ser un tanto diferentes a los que realmente tiene.

8a) El BIOS (Basic I/O System) es el responsable de iniciar la carga del SO a través del MBC. Está grabado en un chip.

8b) El UEFI (Unified Extensible Firmware Interface) define la interfaz entre el gestor de arranque y el firmware. Reemplaza la antigua interfaz del BIOS estándar.

8c) El MBR (Master Boot Record) es el "sector cero" de un dispositivo de almacenamiento, como un disco duro. Se emplea para el arranque del SO con bootstrap o para almacenar una tabla de particiones.

El MBC (Master Boot Code) es un pequeño trozo de código que ocupa los primeros bytes del MBR, el cual es cargado por el BIOS para comenzar el proceso de carga.

8d) El GPT (GUID Partition Table) es un sistema para solucionar limitaciones del MBR, como la cantidad de particiones. Especifica la ubicación y formato de la tabla de particiones en un disco duro.

8e) Un gestor de arranque (bootloader) es un programa sencillo diseñado exclusivamente para preparar todo lo que necesita el SO para funcionar. Normalmente se utilizan cargadores de arranque multietapas, en los que varios programas pequeños se suman los unos a los otros, hasta que el último de ellos carga el SO.

GRUB (Grand Unified Bootloader) es un gestor de arranque múltiple que se usa comúnmente para iniciar dos o más SO instalados en una misma computadora.

LILO (Linux Loader) es un gestor de arranque de Linux que permite iniciar este SO junto con otras plataformas en la misma computadora. Funciona en una variedad de sistemas de archivos.

8fg) Al encender una computadora, ésta se comprueba a sí misma usando un código llamado POST para verificar que los elementos imprescindibles para su funcionamiento están presentes y disponibles (RAM, por ej).

Si la prueba se supera con éxito, se ejecuta un programa denominado "bootstrap loader" para buscar un sector de arranque en alguno de los dispositivos de almacenamiento disponibles. Un sector de arranque es el primer sector de un disco y contiene un pequeño programa (gestor de arranque) capaz de cargar un SO. El bootstrap loader tiene una lista de lugares en que buscar un sector de arranque.

Cuando se encuentra el sector de arranque, se ejecuta el gestor de arranque. En el caso concreto de LILO, se muestra una selección de SO a cargar, y se procede a cargar el SO escogido por el usuario, o el que se haya predeterminado por defecto.

Una vez que LILO ha cargado el kernel de Linux, le pasa el control a este. El kernel monta el sistema de ficheros raíz ("/"), y ejecuta el programa "init" que se encarga de arrancar el resto de procesos que la máquina debe ejecutar (comprobar y montar el sistema de archivos, iniciar los daemons, entre otros (ver "/etc/inittab").

Una vez iniciados todos los servidores y procesos de entrada de usuario, podemos estar delante de una consola de texto, una consola gráfica, o una pantalla llena de opciones para ejecutar.

8h) A grandes rasgos, cuando se ejecuta el proceso de parada, se notifica a los usuarios de este hecho, y además se bloquea el sistema para que nadie más pueda acceder.

Acto seguido, se envía la señal SIGTERM a todos los procesos no definidos en `/etc/inittab`, provocando que terminen su ejecución de modo ordenado.

Poco después, se envía una señal de SIGKILL para que los procesos que no hayan atendido a SIGTERM concluyan también, pero de una manera estricta.

El proceso de parada lleva a cabo su cometido enviando una señal a init para que cambie a uno de estos niveles de ejecución, dependiendo de lo que se desea conseguir:

- 1: Apagar o detener el sistema (nivel 0, opción "-r")
- 2: Entrar en modo monousuario (nivel 1, opción por defecto)
- 3: Reiniciar el sistema (nivel 6, "-h")

Una vez cambiado el nivel de ejecución, se invoca a halt, reboot, o poweroff según corresponda.

8i) Si, ya que un disco rígido puede particionarse y en cada partición tener un sistema de archivos distinto, sería como tener varios discos distintos, uno con cada SO, por lo tanto se necesitará un gestor de arranque.

9a) La base del sistema de archivos de Linux, es obviamente el archivo, que no es otra cosa que la estructura empleada por el sistema operativo para almacenar información en un

dispositivo. Un archivo puede contener cualquier tipo de información, y tiene un nombre que debe cumplir unas ciertas reglas:

- Puede tener entre 1 y 255 caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada / y no es recomendable emplear los caracteres especiales (= ^ ~ ' " ` * ; - ? [] () ! & ~ < >).
- Se pueden utilizar números exclusivamente si así se desea. Las letras mayúsculas y minúsculas se consideran diferentes, y por lo tanto no es lo mismo carta.txt que Carta.txt ó carta.Txt.

Se puede emplear un cierto criterio de "tipo" para marcar las distintas clases de ficheros empleando una serie de caracteres al final del nombre, y que indiquen el tipo de fichero del que se trata. Así, los ficheros de texto, HTML, las imágenes PNG o JPEG tienen extensiones .txt, .htm (o .html), .png y .jpg (o .jpeg) respectivamente.

Pese a esto, Linux sólo distingue 3 tipos de archivos:

- Archivos o ficheros ordinarios, son los mencionados anteriormente.
- Directorios (o carpetas), es un archivo especial que agrupa otros ficheros de una forma estructurada.
- Archivos especiales, son la base sobre la que se asienta Linux, puesto que representan los dispositivos conectados a un ordenador, como puede ser una impresora. De esta forma introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos

dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

9b)

- Vim: Es un editor con dos modos: edición y comandos. En el modo de edición el texto que ingrese será agregado al texto, en modo de comandos las teclas que oprima pueden representar algún comando de Vi.
- MC: Es un editor de texto como el EDIT del DOS.
- Cat: Se utiliza para concatenar archivos y mostrarlos por la salida estándar (normalmente la pantalla).
- More: es un filtro para paginar texto, mostrando una pantalla cada vez.

10b) cd: El comando cd nos permite cambiar de directorio, igual que en DOS. Por ejemplo, "cd /" nos lleva al directorio raíz, que es de donde surgen todos los directorios del sistema.

10c) touch: El comando touch nos permite crear un archivo en el directorio en el que estamos situados actualmente. Por ejemplo, "touch archivo.txt" crea un archivo con ese nombre y formato en la carpeta actual.

10d) ls: Con el comando ls podemos listar los archivos dentro de un determinado directorio. Por ejemplo, si ejecutamos "ls" sin ninguna opción, nos listará los archivos del directorio actual. Si, en cambio, ejecutamos "ls carpeta", nos listará los archivos del directorio de nombre "carpeta".

Además, puede recibir varias opciones:

- "-l": Muestra mucha más información sobre los archivos, como el usuario y el grupo al que pertenece, el tamaño, los permisos, etc.
- "-a": Muestra todos los archivos, incluyendo los ocultos.
- "-t": Ordena los archivos por fecha de modificación.

10e) pwd: El comando pwd es muy sencillo, nos muestra la ruta de directorios en la que estamos situados en este momento. Por ejemplo, `"/home/user"`

10f) find: El comando find permite encontrar archivos, utilizando diversas técnicas. En principio, si se le pasa como parámetro únicamente una determinada ruta, por ejemplo `"find /home/user"`, el comando buscará todos los archivos y directorios que se encuentren a partir de esa ruta. Utilizando algunos otros parámetros es posible buscar los archivos por diversos criterios:

`find . -name "hola.txt"`: Encuentra todos los archivos llamados `hola.txt` que se encuentren a partir del directorio actual. Las comillas no son obligatorias, pero son recomendables si se quieren usar opciones más complejas.

`find . -size 50k`: Busca los archivos que ocupan 50 kilobytes a partir del directorio actual. Si se utiliza `find . size 20c`, buscará los archivos que ocupen 20 bytes.

10g) df: Si se ejecuta el comando df sin ninguna opción, se nos dará una estadística sobre el espacio total, el ocupado y el libre de todas las unidades de disco montadas.

10h) who: El comando who nos permite ver los usuarios que están utilizando el sistema, incluyendo la fecha en la que se conectaron al sistema.

10j) tail: El comando tail nos permite ver la cola de un archivo, es decir sus últimas líneas (últimas 10 por omisión). Esto, aunque a simple vista no parezca útil, resulta de gran utilidad para ver archivos grandes como, por ejemplo, logs (archivos donde se guardan mensajes de estado y/o de errores).

11a) shutdown: Para cerrar y apagar el sistema. shutdown -h now cierra y apaga el sistema; shutdown -r now cierra y reinicia el sistema.

11b) reboot: Reiniciar el sistema.

11c) halt: Apagar el sistema.

11d) locate: Realiza búsquedas en una base de datos interna que se actualiza periódicamente y nos permite hacer búsquedas bastante más rápidas que find. (los resultados no siempre estarán actualizados).

11e) uname: Indica nombre y versión del sistema operativo. Usado con la opción -a muestra toda la información.

11f) gmesg: Para ver rápidamente el log del último booteo del sistema.

11g) lspci: Es una utilidad que nos da información de todos los buses PCI en el sistema y de todos los dispositivos conectados en él.

11h) at: Ejecuta comandos a una hora determinada.

11i) netstat: Muestra un listado de las conexiones activas de un ordenador, tanto entrantes como salientes.

11j) mount: En Linux, antes de que un disco duro (o cualquier otro dispositivo de almacenamiento) pueda ser utilizado, se necesita que el sistema "coloque" ese dispositivo dentro de un directorio. Este directorio casi siempre suele ser el directorio /mnt, pero en realidad se puede utilizar cualquiera.

11k) umount: Cuando se desea que el sistema "remueva" un dispositivo, se utiliza este comando.

11l) head: Muestra las 10 primeras líneas de un fichero específico.

11m) losetup: Activa y controla los dispositivos de bucle (loop).

11n) write: Permite comunicarse con otro usuario, copiando líneas de su terminal a la de él.

11ñ) mkfs: Se emplea para construir un sistema de ficheros de Linux sobre un dispositivo, usualmente una partición de un disco duro.

11o) fdisk (manejar con cuidado): es un programa guiado por menús para la creación y manipulación de tablas de partición.