

Introducción a los Sistemas Operativos

Práctica 1

El objetivo de esta práctica es que el alumno se familiarice con los conceptos básicos del sistema operativo GNU/Linux, así como con su entorno y comandos principales.

1. Características de GNU/Linux:

(a) Mencione y explique las características más relevantes de GNU/Linux.

- Es un sistema operativo tipo Unix, pero libre.
- *Es la fusión de GNU, un proyecto de Richard Stallman, y Linux, un kernel de Linus Torvalds.*
- Es código abierto, lo que nos permite estudiarlo, personalizarlo, auditarlo, aprovecharnos de la documentación, etc...
Podemos ver cómo está hecho.
- Existen diversas distribuciones (customizaciones)

(b) Mencione otros sistemas operativos y compárelos con GNU/Linux en cuanto a los puntos mencionados en el inciso a.

Windows y MacOS son dos softwares propietarios o privados.

Propiedad de Microsoft y Apple respectivamente.

Es decir, no se puede acceder a su código fuente y no se permite su modificación.

Android es un sistema operativo prácticamente libre adquirido y desarrollado principalmente por Google. Al ser una distribución de Linux, su código fuente principal es libre y de código abierto. Pero la PlayStore y la gran mayoría de las aplicaciones son software propietario (muchas de ellas adrede por Google, muy importantes para la experiencia de usuario), lo que hace perder control y privacidad a los usuarios.

(c) ¿Qué es GNU?

GNU es un acrónimo recursivo que significa '*GNU's Not Unix*'

GNU se refiere a 4 libertades principales de los usuarios del software:

- Libertad de usar el programa con cualquier propósito
- Libertad de estudiar su funcionamiento
- Libertad para distribuir sus copias
- Libertad para mejorar los programa

(d) Indique una breve historia sobre la evolución del proyecto GNU.

Un proyecto iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre.

En 1985, Stallman crea la FSF (Free Software Foundation), con el fin de financiar el proyecto GNU.

En 1989 y para asegurar que el proyecto GNU fuera libre, se necesitó crear un marco regulatorio conocido como GPL (*General Public License*).

Le faltaba el componente principal, el Kernel.

Hubieron dos intentos fallidos (TRIX y MACH)

Como Linus Torvalds ya venía trabajando desde 1991 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) Explique qué es la multitarea, e indique si GNU/Linux hace uso de ella.

Un sistema operativo multitarea divide el tiempo de procesador disponible entre los procesos o subprocesos que lo necesitan, con el objetivo de estar ocioso el menor tiempo posible.

Permite que de manera aparente, se ejecuten varios programas a la vez.

Si se tiene un sistema con varios procesadores, hay multitarea real, lo que no quiere decir que con un solo procesador no haya multitarea:

Por debajo, se 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.

Linux tiene multitarea. Se basa en la llamada multitarea preventiva, es decir, es el sistema operativo (el kernel) quien se encarga de administrar y transferir el control y uso del procesador y los recursos de hardware entre los múltiples procesos que lo requieran.

(f) ¿Qué es POSIX?

POSIX (*acrónimo de Portable Operating System Interface, y X viene de UNIX*) es una norma escrita por la IEEE, que define una interfaz estándar del sistema operativo y el entorno, incluyendo un intérprete de comandos (o "shell").

El término fue sugerido por Richard Stallman en la década de 1980, en respuesta a la demanda del IEEE, que buscaba un nombre fácil de recordar.

POSIX no es un sistema operativo, sino que es un *estándar* cuya finalidad es facilitar la portabilidad de las aplicaciones, en el nivel del fuente, entre muchos sistemas.

2. Distribuciones de GNU/Linux

(a) ¿Qué es una distribución de GNU/Linux?

Nombre al menos 4 distribuciones de GNU/Linux y cite diferencias básicas entre ellas.

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

Cuatro distribuciones de GNU/Linux que puedo mencionar son:

Lihuen: Es una distribución Linux originalmente basada en GnuLinEx y desarrollada por la Facultad de Informática de la Universidad Nacional de La Plata, Argentina, con el objetivo de ser útil para entornos educativos y oficinas, orientada a usuarios con poca o ninguna experiencia en el uso de GNU/Linux.

Ubuntu: Es una distribución diseñada para ser fácil de usar y de instalar. Da una experiencia de usuario amigable y confiable.

Kali Linux: Es una distribución que se especializa en la seguridad informática y las pruebas de penetración. Está diseñado para profesionales de la seguridad y hackers éticos. Incluye una amplia gama de herramientas de seguridad preinstaladas, como herramientas de auditoría de redes y sistemas, análisis de vulnerabilidades, y más.

SteamOS: Es una distribución derivada de Arch Linux, basada en Linux y desarrollada por Valve como sistema operativo principal de la línea de videoconsolas Steam Machines.

Puede ser instalado en cualquier PC, aunque está orientado más a ser un media center que un SO de uso general. Está optimizado para los juegos.

(b) ¿En qué se diferencia una distribución de otra?

Cada distribución está pensada para un objetivo en particular y suele realizar modificaciones en el kernel, además de incluir un conjunto de programas, configuraciones y políticas alineadas con ese objetivo específico. Esto puede incluir la elección de un entorno de escritorio, un conjunto de herramientas de administración, controladores de hardware específicos y otras personalizaciones.

(c) ¿Qué es Debian? Acceda al sitio 1 e indique cuáles son los objetivos del proyecto y una breve cronología del mismo.

El **Proyecto Debian** es una asociación de personas que han hecho causa común para crear un sistema operativo (SO) libre. **Debian** es un Sistema Operativo.

Los sistemas Debian actualmente usan el kernel de Linux o de FreeBSD

(FreeBSD es un sistema operativo que incluye un núcleo y otro software.)

Debian lo producen cerca de un millar de desarrolladores activos, dispersos por el mundo que ayudan voluntariamente en su tiempo libre.

Debian comenzó en agosto de 1993 gracias a Ian Murdock, como una nueva distribución que se realizaría de forma abierta, en la línea del espíritu de Linux y GNU. Debian estaba pensado para ser creada de forma cuidadosa y concienzuda, y ser mantenida y soportada con el mismo cuidado. Comenzó como un grupo de pocos y fuertemente unidos hackers de Software Libre, y gradualmente creció hasta convertirse en una comunidad grande y bien organizada de desarrolladores y usuarios.

3. Estructura de GNU/Linux:

(a) Nombre cuales son los 3 componentes fundamentales de GNU/Linux.

- Kernel (Núcleo)
- Shell (Intérprete de comandos)
- FileSystem (Sistema de archivos)

(b) Mencione y explique la estructura básica del Sistema Operativo GNU/Linux

El Kernel es el componente principal del SO. Es software, que se encarga de administrar los recursos de hardware (memoria y CPU) entre los diferentes procesos que lo requieran.

El usuario se comunica con el SO mediante el Shell. En el caso de GNU/Linux es la consola de comandos, de tipo CUI (*Character User Interface*).

El shell en Windows sería el sistema de ventanas, que es de tipo GUI (Graphical User Interface)

4. Kernel:

(a) ¿Qué es?

El kernel es el componente principal del sistema operativo.

De hecho, en un sentido estricto, el kernel es el sistema operativo.

El kernel o núcleo, se encarga de administrar los recursos del hardware, y gestionar la comunicación con los dispositivos de E/S.

Es quien se ocupa de que el software y el hardware puedan trabajar juntos.

(a) Indique una breve reseña histórica acerca de la evolución del Kernel de GNU/Linux.

- En 1991 Linus Torvalds inicia la programación de un Kernel Linux basado en Minix (clon de Unix desarrollado por Tenenbaum 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
- Desarrollo continuado por miles de programadores alrededor del mundo
- En julio de 1996 se lanza la versión 2.0 y se define la nomenclatura de versionado. Se desarrolló hasta febrero de 2004 y terminó con la 2.0.40
- 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
- En 2001 se lanza la versión 2.4 y se deja de desarrollar a fines del 2010 con la 2.4.37.11
- La versión 2.4 fue la que catapultó a GNU/Linux como un SO estable y robusto. Durante este período es que se comienza a utilizar Linux más asiduamente
- A fines del 2003 se lanza la versión 2.6
- Esta versión ha tenido muchas mejoras para el SO dentro de las que se destacan soporte de hilos, mejoras en la planificación y soporte de nuevo hardware
- El 3 de agosto de 2011 se lanza la versión 2.6.39.4 anunciándose la misma desde meses previos como la última en su revisión
- El 17 de julio de 2011 se lanza la versión 3.01
No agrega mayores cambios. La decisión del cambio son los 20 años del SO y no superar los 40 números de revisión. Totalmente compatible con 2.6

(b) ¿Cuáles son sus funciones principales?

Sus funciones más importantes son la administración de memoria, CPU y la E/S

**(c) ¿Cuál es la versión actual? ¿Cómo se definía el esquema de versionado del Kernel en versiones anteriores a la 2.4?
¿Qué cambió en el versionado se impuso a partir de la versión 2.6?**

La última versión estable es la 6.4.11 (agosto de 2023).

Antes de la versión 2.6, el esquema de versionado del kernel de Linux utilizaba números pares e impares para indicar versiones estables y no estables, respectivamente. Por ejemplo, la versión 2.4 era una versión estable, mientras que la versión 2.5 era una versión de desarrollo.

Sin embargo, a partir de la versión 2.6, el esquema de versionado cambió y ya no se utilizan números pares e impares para indicar versiones estables y no estables.

(d) ¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?

Sí, es posible tener más de un kernel de GNU/Linux instalado en la misma máquina. Sin embargo, solo se puede usar un kernel a la vez. Cuando la máquina arranca, se puede elegir qué versión del kernel deseas usar.

(e) ¿Dónde se encuentra ubicado dentro del File System?

El Kernel de Linux se encuentra en el directorio /boot.

Este directorio contiene todo lo necesario para el proceso de arranque, excepto los archivos de configuración que no son necesarios en el momento del arranque.

Por lo tanto, el directorio /boot almacena datos que se utilizan antes de que el Kernel comience a ejecutar programas en modo usuario.

(f) ¿El Kernel de GNU/Linux es monolítico? Justifique.

El Kernel de GNU/Linux es **Monolítico Híbrido**.

Monolítico: Esto quiere decir que toda funcionalidad que debe implementar el sistema operativo se ejecuta en modo Kernel.

Absolutamente toda la lógica está en el kernel, por lo tanto evidentemente se pasa el CPU a modo kernel.

Se enfoca más en la performance. En teoría, es más veloz.

La otra opción al monolítico sería el Micro-Kernel. En esa organización, se intenta que el kernel sea lo más chico posible. Hay varias partes lógicas que dan apoyo al Kernel que se ejecutan en modo usuario.

Se enfoca más en la seguridad. En teoría, al estar en modo kernel menos tiempo, se reduce la posibilidad de que se den errores críticos para el sistema operativo (como la pantalla azul de la muerte, en Windows).

Híbrido: Esto quiere decir que se tiene la capacidad de agregar y eliminar módulos de código de manera dinámica del Kernel.

Por ejemplo, los drivers de una impresora, se descargan en el kernel y se ejecutan en ese modo.

5. Intérprete de comandos (Shell):

(a) ¿Qué es?

También conocido como CLI (Command Line Interface) es el modo de comunicación entre el usuario y el SO.

(b) ¿Cuáles son sus funciones?

Actúa como interfaz de usuario para comunicar al usuario con el sistema operativo mediante una consola gráfica o ventana.

Permite ejecutar comandos, explorar el árbol de directorios del sistema, crear, editar y eliminar archivos.

Incorpora características tales como control de procesos, redirección de entrada/salida, listado y lectura de archivos, protección, comunicaciones.

Proporciona un lenguaje de órdenes para escribir programas por lotes o scripts.

(c) Mencione al menos 3 intérpretes de comandos que posee GNU/Linux y compárelos entre ellos.

Bourne Shell (sh):

- Es el shell original de Unix.
- Se consideraba el mejor shell en términos de eficiencia.
- Proporciona una sintaxis para la expansión de parámetros y variables, redirección y comillas.

Korn Shell (ksh):

- Es un shell más antiguo desarrollado por David Korn en los Laboratorios Bell.
- Combina las características de muchos shells como Bourne shell, C shell, TC shell, etc.
- Proporciona funcionalidades interactivas al igual que lenguajes como C, pero se ejecuta más rápido y además dispone de funciones de edición de comandos más avanzadas.
- Es muy fácil de usar, por lo que los usuarios sin experiencia prefieren KSH1.

Bourne Again Shell (bash):

- Es un clon de Bourne Shell.
- Implementa esencialmente la misma gramática, expansión de parámetros y variables, redirección y comillas que Bourne Shell.
- Tiene licencia GNU, por lo que es de código abierto y está disponible de forma gratuita para el público en general.
- Tiene historial de comandos y órdenes internas para manipularla.

(d) ¿Dónde se ubican (path) los comandos propios y externos al Shell?

Los comandos internos, al ser propios de la Shell se encuentran en el mismo directorio que esta: /bin.

Los comandos externos, en un PATH a especificar.

Los comandos del shell en Linux se pueden clasificar en dos categorías:

Comandos internos:

Estos comandos están integrados en el shell. La ejecución de estos comandos es rápida ya que el shell no tiene que buscarlos en la ruta dada en la variable PATH, y tampoco es necesario generar ningún proceso para ejecutarlos.

Comandos externos:

Estos son comandos que no están integrados en el shell. Por lo tanto, para ejecutar estos comandos, el shell debe buscarlos en la ruta especificada por la variable PATH y puede ser necesario generar un proceso para su ejecución.

(e) ¿Por qué considera que el Shell no es parte del Kernel de GNU/Linux?

El Shell y el Kernel son dos componentes fundamentales de un sistema operativo, pero cumplen roles diferentes, por lo tanto es evidente que son dos cosas separadas.

El Kernel es la interfaz entre el hardware de una computadora y sus procesos, el Shell es la interfaz entre el usuario humano y el Kernel.

(f) ¿Es posible definir un intérprete de comandos distinto para cada usuario? ¿Desde dónde se define? ¿Cualquier usuario puede realizar dicha tarea?

Sí, es posible definir un intérprete de comandos distinto para cada usuario en un sistema Linux. Esto se define en el archivo `/etc/passwd`. Cada línea de este archivo representa una cuenta de usuario y el último campo de cada línea especifica el intérprete de comandos predeterminado para ese usuario.

Para cambiar el intérprete de comandos de un usuario, se puede usar el comando `chsh` (change shell). Este comando cambia la shell de inicio de sesión del usuario.

Sólo el usuario `root` o el propio usuario pueden cambiar el intérprete de comandos de un usuario. Los usuarios normales no pueden cambiar el intérprete de comandos de otros usuarios.

6. Sistema de Archivos (File System):

(a) ¿Qué es?

El File System o Sistema de Archivos es el cómo se organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento.

Sus principales funciones son la asignación de espacio a los archivos, la administración del espacio libre y del acceso a los datos.

(b) Mencione sistemas de archivos soportados por GNU/Linux.

ext2, ext3 y ext4:

Son los sistemas de archivos nativos de Linux. Cada uno es una evolución del anterior, siendo ext4 el más reciente y utilizado actualmente.

ReiserFS:

Es otro sistema de archivos nativo de Linux, conocido por ser eficiente con una gran cantidad de archivos pequeños.

NFS:

Es utilizado para sistemas de archivos distribuidos en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.

(c) ¿Es posible visualizar particiones del tipo FAT y NTFS en GNU/Linux?

Linux también ofrece soporte para sistemas de ficheros de Windows, como FAT, FAT32 y NTFS. Tanto para FAT como para FAT32, Linux tiene soporte completo y estable de escritura y lectura, mientras que para NTFS, y con las últimas versiones del kernel, solo se puede acceder de manera estable en modo lectura. Existen programas para la escritura, pero se recomienda precaución al hacerlo para no perder los datos.

FAT: El sistema de archivos FAT (File Allocation Table) fue creado por Microsoft en 1977. Se sigue utilizando hoy en día como el sistema de archivos preferido para los medios de disquetera y los dispositivos de almacenamiento portátiles de alta capacidad como las unidades flash y otros dispositivos de memoria de estado sólido como las tarjetas SD. FAT era el principal sistema de archivos utilizado en todos los sistemas operativos de consumo de Microsoft, desde MS-DOS hasta Windows ME.

NTFS: NTFS (New Technology File System) es un sistema de archivos desarrollado por Microsoft. Se introdujo por primera vez con Windows NT 3.1 en 1993 y en se sigue usando en la actualidad siendo el sistema de archivos principal en Windows 11.

(d) ¿Cuál es la estructura básica de los File System en GNU/Linux? Mencione los directorios más importantes e indique qué tipo de información se encuentra en ellos. ¿A qué hace referencia la sigla FHS?

La estructura básica del sistema de archivos en GNU/Linux se organiza en una estructura jerárquica, de tipo árbol. El nivel más alto del sistema de ficheros es `/` o directorio raíz. Todos los demás ficheros y directorios están bajo el directorio raíz.

El Estándar de Jerarquía del Sistema de Archivos (FHS, por sus siglas en inglés) es el estándar que establece y brinda el detalle de los nombres, contenidos, ubicaciones y permisos de los archivos y directorios. Es el conjunto de reglas que determinan una estructuración común de archivos y directorios en los sistemas Linux.

Los sistemas Linux residen bajo un árbol jerárquico de archivos, bastante parecido a como se estructura los sistemas Unix. En sus inicios, ese árbol jerárquico de directorios y archivos no estaban bajo ningún estándar, es decir, existían variaciones entre el de una distribución y otra.

El FHS se define como el estándar que establece y brinda el detalle de los nombres, contenidos, ubicaciones y permisos de los archivos y directorios. En otras palabras, es el conjunto de reglas que determinan una estructuración común de archivos y directorios en los sistemas Linux.

Directorios más importantes según FHS:

- **/** 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) Definición. Tipos de particiones. Ventajas y Desventajas.

Definición:

Una partición de disco es una división lógica en una unidad de almacenamiento, en la cual se alojan y organizan los archivos mediante un sistema de archivos. Cada partición tiene su propio sistema de archivos y generalmente, casi cualquier 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.

Tipos de particiones:

Existen tres tipos principales de particiones:

Primaria: Son las más elementales y un disco duro puede contener hasta 4 de estas. Es la partición desde la que puede arrancar el sistema operativo. Se almacena información de la misma en el MBR.

Extendida o Secundaria: Sirve para contener unidades lógicas en su interior. Se guardan ficheros en ellas. Solo puede existir una partición de este tipo por disco. No puede contener los archivos necesarios para arrancar el sistema.

Lógica: Son divisiones que se pueden llegar a encontrar dentro de una partición extendida. 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 de las Particiones:

- Permite funciones avanzadas como la instalación de múltiples sistemas operativos.
- Sirve para mantener los archivos del sistema operativo de manera separada, lo que es una buena práctica por cuestiones de seguridad, privacidad y organización.
- Se puede tener una partición con todos los datos necesarios para la restauración de todo el sistema.

Desventajas de las Particiones:

- Ofrece un falso sentido de la seguridad. Aunque tengamos los archivos divididos por particiones, el disco duro no deja de ser una unidad. Esto quiere decir que, si falla, igualmente perderemos todo lo que contenga.
- Con múltiples particiones puede ocurrir que en algunas de ellas nos quedemos sin espacio, pero en otras nos sobre.

**(b) ¿Cómo se identifican las particiones en GNU/Linux?
(Considere discos IDE, SCSI y SATA).**

En GNU/Linux, las particiones se identifican de la siguiente manera:

Los discos se identifican siempre con las letras: `/dev`` seguidas de un identificador para las particiones.

- Para los discos **IDE**, se utiliza ``hd`` seguido de una letra del abecedario que identifica el disco (por ejemplo, ``hda`` para el primer disco duro IDE), y un número que identifica la partición (por ejemplo, ``hda1`` para la primera partición del primer disco duro IDE)⁴.
- Para los discos **SCSI** y **SATA**, se utiliza ``sd`` seguido de una letra del abecedario que identifica el disco, y un número que identifica la partición. Por ejemplo, `/dev/sda1`` se refiere a la primera partición del primer disco SCSI o SATA.

Por lo tanto, el nombre completo de una partición en GNU/Linux incluirá el prefijo `/dev/``, seguido del tipo de disco (``hd`` para IDE, ``sd`` para SCSI o SATA), una letra que identifica el disco, y finalmente un número que identifica la partición en ese disco.

(c) ¿Cuántas particiones son necesarias como mínimo para instalar GNU/Linux? Nómbralas indicando tipo de partición, identificación, tipo de File System y punto de montaje.

El número mínimo de particiones necesarias para instalar GNU/Linux es uno, que sería la partición ROOT

(partición de tipo primaria, identificado como ' /dev/sda1' por ejemplo, y con punto de montaje en '/').

En una instalación típica de GNU/Linux es recomendable al menos dos particiones: la partición root y la partición de SWAP.

(La partición de intercambio en GNU/Linux, también conocida como swap, es una sección independiente del disco duro que se utiliza únicamente para el intercambio. Es un espacio en disco reservado para almacenar procesos inactivos cuando la memoria RAM física es baja)

Algunos recomiendan tener tres particiones: swap, root y home.

La partición home es donde se almacenan los datos del usuario.

(d) Ejemplifique diversos casos de particionamiento dependiendo del tipo de tarea que se deba realizar en su sistema operativo.

Estudiante Universitario:

- Partición primaria raíz ("/"): Para el sistema operativo y las aplicaciones.
- Partición primaria Swap.
- Partición primaria de datos personales: Almacenamiento de archivos personales.
- Partición primaria de trabajos universitarios: Para documentos y proyectos de la universidad.

Estudio de Contaduría:

- Partición primaria raíz: Sistema operativo y software contable.
- Partición primaria Swap.
- Partición primaria de datos contables: Almacenamiento de datos financieros.
- Partición primaria de respaldo: Para backup regular de los datos.

Fotógrafo Profesional:

- Partición primaria raíz: Sistema operativo y software de edición y herramientas.
- Partición primaria Swap.
- Partición primaria de respaldo: Para tener backup de los proyectos.
- Partición extendida de proyectos:

Para ir llevando de manera organizada los proyectos en curso, en particiones lógicas. Una partición lógica para cada proyecto.

**(e) ¿Qué tipo de software para particionar existe?
Menciónelos y compare.**

Para crear las particiones se utiliza software denominado particionador.

Existen 2 tipos:

- Destructivos: permiten crear y eliminar particiones (fdisk)
- No destructivo: permiten crear, eliminar y modificar particiones (fips, gparted)

La principal diferencia entre los dos tipos de software de particionamiento es que los destructivos borran todos los datos en una partición cuando se realiza una operación, mientras que los no destructivos conservan los datos durante las operaciones de particionamiento.

8. Arranque (bootstrap) de un Sistema Operativo:

(a) ¿Qué es el BIOS? ¿Qué tarea realiza?

El BIOS, o Sistema Básico de Entrada/Salida, es un software integrado en la placa madre de una computadora que se encarga de realizar una serie de tareas críticas para el funcionamiento del sistema. En términos simples, el BIOS es el primer programa que se ejecuta cuando se enciende una computadora y es responsable de verificar que todos los componentes de hardware estén funcionando correctamente antes de cargar el sistema operativo.

(b) ¿Qué es UEFI? ¿Cuál es su función?

El UEFI (Interfaz Extensible Firmware Unificada) es un software que reemplaza al BIOS y se encarga de controlar el hardware de una computadora cuando se enciende y de interactuar con el sistema operativo. A diferencia del BIOS, el UEFI ofrece una interfaz gráfica de usuario más moderna y actualizada, soporte para discos duros de gran capacidad, arranque más rápido y mejor seguridad.

(c) ¿Qué es el MBR? ¿Qué es el MBC?

El MBR (Master Boot Record) es un sector reservado del disco que se encuentra al principio del mismo (Cilindro 0, cabeza 0, sector 1). El tamaño estándar del MBR es el mismo que el de un sector de un disco: 512 bytes, de los cuales:

- Los primeros 445 bytes contienen el MBC
- Los siguientes 64 bytes corresponden a la tabla de particiones.
- Al final se cuenta con 2 bytes libres o para firmar el MBR

El MBC (Master Boot Code) es un pequeño código que permite cargar el SO. La última acción del BIOS es leer el MBC. Lo lleva a memoria y lo ejecuta.

(d) ¿A qué hacen referencia las siglas GPT? ¿Qué sustituye? Indique cuál es su formato.

Las siglas GPT hacen referencia a GUID Partition Table.

La GPT es un estándar para la colocación de la tabla de particiones en un disco duro físico. Es parte del estándar Extensible Firmware Interface (EFI) propuesto por Intel para reemplazar el viejo BIOS del PC, heredado del IBM PC original.

La GPT sustituye al Registro de Arranque Principal (MBR) usado con el BIOS .

La GPT tiene varias ventajas sobre el MBR, como una mayor capacidad de almacenamiento, una mayor redundancia y una

mayor flexibilidad en cuanto a la cantidad y el tamaño de las particiones.

(e) ¿Cuál es la funcionalidad de un “Gestor de Arranque”? ¿Qué tipos existen? ¿Dónde se instalan? Cite gestores de arranque conocidos.

El bootloader es un programa que se carga después del código del BIOS y su función principal es cargar una imagen del kernel del sistema operativo desde alguna partición para su ejecución. El bootloader es responsable de detectar y cargar el kernel del sistema operativo, así como de configurar el entorno necesario para su correcta ejecución.

En general, el bootloader se instala en el MBR, y toma el lugar del MBC.

Dos gestores de arranque conocidos en Linux son GRUB y LILO. GRUB es uno de los gestores de arranque más populares y se utiliza comúnmente en sistemas Linux. Es altamente configurable y puede detectar automáticamente otros sistemas operativos instalados en la computadora. LILO, por otro lado, es más simple que GRUB pero también menos configurable.

(f) ¿Cuáles son los pasos que se suceden desde que se prende una computadora hasta que el Sistema Operativo es cargado (proceso de bootstrap)?

- El BIOS (Basic I/O System) es el responsable de iniciar la carga del SO a través del MBR
- Se carga el programa de booteo (desde el MBR)
- El gestor de arranque lanzado desde el MBR carga el Kernel:
 - Prueba y hace disponibles los dispositivos.
 - Luego pasa el control al proceso init.

El proceso de bootstrap se ve como una serie de pequeños programas de ejecución encadenados.

Considerar que esto es de la década del 80. Como ya hemos visto, el GPT reemplazó al MBR y la UEFI reemplazó a la BIOS.

(g) Analice el proceso de arranque en GNU/Linux y describa sus principales pasos.

1. **BIOS:** Al encender la computadora, el BIOS realiza algunas revisiones de integridad del sistema. Busca, carga y ejecuta el programa de arranque (boot loader).

2. **MBR (Master Boot Record):** Se encuentra en el primer sector de un disco arrancable. Contiene información acerca de GRUB (o LILO en los sistemas antiguos). En pocas palabras, el BIOS carga y ejecuta el boot loader que se encuentra en el MBR.

3. **Gestor de Arranque GRUB (Grand Unified Bootloader) :**
Si tenemos varios kernels instalados en nuestro sistema, podemos seleccionar cuál de ellos deseamos ejecutar. GRUB tiene conocimiento acerca del sistema de archivos.

4. **Kernel:** En Linux, el flujo de control durante el arranque es desde el firmware al gestor de arranque y al núcleo (kernel). El núcleo inicia el planificador (para permitir la multitarea) y ejecuta el primer espacio de usuario y el programa de inicialización.

5. **Init:** Este proceso ejecuta secuencias de comandos necesarios para configurar todos los servicios y estructuras que no sean del sistema operativo, a fin de permitir que el entorno de usuario sea creado y pueda presentarse al usuario con una pantalla de inicio de sesión.

Cada proceso de arranque será diferente dependiendo de la arquitectura del procesador y el BIOS. Para los dispositivos con arranque mediante UEFI, el cargador de arranque se almacena como ejecutable EFI.

(h) ¿Cuáles son los pasos que se suceden en el proceso de parada (shutdown) de GNU/Linux?

El proceso de parada (shutdown) en GNU/Linux es una secuencia de pasos que se realiza para apagar el sistema de manera segura y ordenada.

- 1. Notificación a los usuarios:** Se notifica a todos los usuarios que el sistema se va a apagar.
- 2. Terminación de procesos en ejecución:** Se envía la señal de terminación (TERM) a todos los procesos en ejecución. Esto les da la oportunidad de terminar de manera segura y ordenada.
- 3. Parada de demonios (servicios):** Los demonios, o servicios, que están en ejecución se paran. Esto asegura que no haya ninguna actividad del sistema que pueda interrumpirse abruptamente durante el apagado.
- 4. Cambio al nivel de ejecución 0:** El sistema cambia al nivel de ejecución 0, que corresponde a la parada del sistema. Como resultado, se ejecutan todos los scripts de parada del directorio `/etc/rc.d/rc0.d`.
- 5. Apagado del hardware:** Finalmente, el sistema operativo envía una señal al hardware para que se apague.

Es importante notar que este proceso puede variar dependiendo de la distribución específica de GNU/Linux y la configuración del sistema.

(i) ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique.

Sí, es posible tener GNU/Linux y otro sistema operativo instalado en la misma PC. Esto se conoce como dual boot (arranque dual).

El dual boot permite a los usuarios instalar dos o más sistemas operativos en diferentes particiones del mismo disco duro. Cuando la computadora se enciende, el gestor de arranque (como GRUB en el caso de Linux) presenta un menú que permite seleccionar qué sistema operativo iniciar.

9. Archivos:

(a) ¿Cómo se identifican los archivos en GNU/Linux?

En GNU/Linux, los archivos se identifican de varias maneras:

1. Nombres de archivos:

Los nombres de archivos en GNU/Linux pueden tener hasta 255 caracteres, pudiendo utilizar los espacios y acentos.

Un punto importante es que se distingue entre mayúsculas y minúsculas. Por lo tanto, los archivos ``file.iso`` y ``File.iso`` son diferentes para el SO.

2. Ubicación en el sistema de archivos:

Todo el sistema Linux tiene por origen la raíz o root, representada por ``/`` y bajo ese directorio se localizan todos los demás archivos accesibles en el sistema operativo.

3. Tipo de archivo:

Linux no se maneja con las extensiones de los archivos como si lo hace Windows.

Si bien GNU/Linux a los archivos se les pueden colocar extensiones como `` .exe``, `` .txt``, `` .docx``, `` .jpg``, etc., pero estas extensiones se usan más por convención que por otra cosa.

El sistema es capaz de reconocer el tipo de archivo por su contenido y no por su extensión.

Por ejemplo, un archivo con extensión `` .txt`` en GNU/Linux es simplemente un archivo de texto, al igual que en otros sistemas operativos. Sin embargo, a diferencia de otros sistemas operativos como Windows, donde la extensión del archivo determina qué programa se utiliza para abrirlo, en GNU/Linux el sistema operativo puede determinar qué programa usar basándose en el contenido real del archivo.

Los tipos de archivos existentes en un sistema GNU/Linux son los siguientes:

- Archivo regular: tipo común que contiene datos solamente.
- Directorios: los directorios son utilizados para separar un grupo de archivos de otros.
- Dispositivos: todo componente de hardware instalable es llamado dispositivo.
- Enlaces: los enlaces son archivos utilizados para hacer referencia a otro archivo ubicado en otro lugar.

(b) Investigue el funcionamiento de los editores vi y mcedit, y los comandos cat y more

VI

El editor vi es un editor de texto que maneja en memoria el texto entero de un archivo. Es el editor clásico de UNIX (se encuentra en todas las versiones).

Puede usarse en cualquier tipo de terminal con un mínimo de teclas, lo cual lo hace difícil de usar al enfrentarse por primera vez al mismo.

MODOS DE VI:

Existen tres modos o estados de vi:

- **Modo comando:** este es el modo en el que se encuentra el editor cada vez que se inicia. Las teclas ejecutan acciones (comandos) que permiten mover el cursor, ejecutar comandos de edición de texto, salir de vi, guardar cambios, etc.
- **Modo inserción o texto:** este es el modo que se usa para insertar el texto. Existen varios comandos que se pueden utilizar para ingresar a este modo.
- **Modo línea o ex:** se escriben comandos en la última línea al final de la pantalla.

MCEDIT

Midnight Commander es un administrador de archivos o gestor de ficheros, bastante ortodoxo, para GNU/Linux y también existe para Windows, que se encuentra licenciado bajo GPL de GNU y por lo tanto califica como Software Libre. Es, básicamente, una aplicación para nuestra terminal o tty rica en funciones que nos permite hacer lo siguiente: copiar, mover, eliminar archivos y directorios, buscar archivos y ejecutar comandos en la subshell.

Posee un visor interno y editor incluido. Ese editores Mcedit

Mcedit es un ejecutable independiente, el cual nos permite visualizar y editar contenido de archivos/ficheros resaltando su sintaxis de ciertos lenguajes de programación y la capacidad de trabajar tanto en modo ASCII como en modo hexadecimal.

CAT

El comando ``cat`` es uno de los comandos más útiles que se pueden aprender en Linux. Deriva su nombre de la palabra "concatenar" y permite crear, fusionar o imprimir archivos en la pantalla de salida estándar o en otro archivo.

Algunos usos comunes del comando ``cat``:

- **Crear un archivo:** Usando el comando ``cat``, puedes crear un archivo rápidamente y agregarle texto. Para hacer eso, usa el operador ``>`` para redirigir el texto en el archivo. Por ejemplo: ``cat > filename.txt``. Así se crea el archivo y puedes comenzar a rellenarlo con texto.
- **Ver el contenido de un archivo:** Este es uno de los usos más básicos del comando ``cat``. Sin necesidad de ninguna opción, el comando leerá el contenido de un archivo y lo mostrará en la consola. Por ejemplo: ``cat filename.txt``.
- **Redirigir contenido:** En lugar de mostrar el contenido de un archivo en la consola, puedes redirigir la salida a otro archivo usando la opción ``>``.
- **Concatenar archivos:** El comando ``cat`` en Linux permite concatenar y mostrar el contenido de archivos. Se utiliza para visualizar, unir y crear archivos.

MORE

El comando ``more`` se utiliza para ver (pero no modificar) el contenido de un archivo de texto o la salida de un comando en la terminal, página por página. Es especialmente útil cuando se ejecuta un comando que produce una gran cantidad de salida, como ``ls`` o ``du``.

Algunos ejemplos de cómo se utiliza el comando ``more``:

- **Ver el contenido de un archivo:** Para ver el contenido de un archivo, simplemente se escribe ``more`` seguido del nombre del archivo. Por ejemplo: ``more filename.txt``.
- **Ver la salida de un comando:** Puedes usar ``more`` para ver la salida de un comando página por página¹. Por ejemplo, si se quiere ver la lista de archivos en un directorio que tiene muchos archivos, puedes usar ``ls | more``.
- **Navegación:** Cuando se está viendo el contenido de un archivo o la salida de un comando con ``more``, se puede presionar la barra espaciadora para avanzar a la siguiente página. Si se quiere salir antes de llegar al final, puedes presionar ``q``.

Es importante mencionar que existe un comando similar llamado ``less``, que proporciona más funcionalidades que ``more``, como la capacidad de moverse hacia atrás en el archivo (algo que no puedes hacer con ``more``)¹.

(c) Cree un archivo llamado “prueba.exe” en su directorio personal usando el vi. El mismo debe contener su número de alumno y su nombre.

Pasos:

- 1- Escribir en la terminal “vi prueba.exe” para crear el archivo
- 2- Apretar ‘i’ para entrar en modo de inserción
- 3- Escribir legajo y nombre
- 4- Escape para salir del modo inserción
- 5- Escribir ‘:wq’ y apretar Enter para guardar
- 6- En la consola ingresar ‘cat prueba.exe’ para ver si todo salió correctamente

(d) Investigue el funcionamiento del comando file. Pruébelo con diferentes archivos. ¿Qué diferencia nota?

El comando file permite identificar el tipo de archivo con el que estás trabajando.

Como ya se vio anteriormente, los archivos en linux llevan extensión por convención, pero el sistema operativo no hace uso real de ello.

Tranquilamente un archivo puede llamarse "archivo.png" y ser un archivo de audio, o un ejecutable.

El comando `file` no se guía por la extensión del archivo, sino que analiza su contenido, más concretamente su número mágico, para intentar reconocer qué tipo de archivo es.

Un número mágico, también conocido como magic number, es un conjunto de bytes que indica inequívocamente el tipo de archivo que estamos analizando. Es decir, el magic number permite determinar, sin ningún tipo de margen de error, el formato del fichero ante el cual nos encontramos.

El número mágico se puede encontrar en la versión hexadecimal del código fuente de cualquier archivo y corresponde a los primeros bytes del programa.

10. Indique qué comando es necesario utilizar para realizar cada una de las siguientes acciones. Investigue su funcionamiento y parámetros más importantes:

(a) Cree la carpeta ISO2023

`mkdir ISO2023`

Este comando permite a los usuarios crear directorios (carpetas). Posibilita la creación de varios directorios a la vez, así como establecer los permisos para los directorios.

(b) Acceda a la carpeta (cd)

`cd [ruta]`

`cd /home/nicolas/documents/ISO2023`

`cd` (de change directory). Con la ayuda de este comando, podemos navegar por todos nuestros directorios en nuestro sistema.

(c) Cree dos archivos con los nombres iso2023-1 e iso2023-2 (touch)

`touch [nombre_del_archivo]`

`touch iso2023-1`

`touch iso2023-2`

Este comando se usa para crear cualquier tipo nuevo de archivo en sistemas Linux.

(d) Liste el contenido del directorio actual (ls)

Este comando enumera el contenido del directorio que desee, archivos y otros directorios anidados. Cuando se usa sin opciones ni argumentos, `ls` muestra una lista en orden alfabético de los nombres de todos los archivos en el directorio de trabajo actual.

(e) Visualizar la ruta donde estoy situado (pwd)

`pwd`

Este comando sirve para localizar la ruta del directorio de trabajo en el que te encuentras actualmente.

(f) Busque todos los archivos en los que su nombre contiene la cadena “iso*” (find)

find iso*

El comando find sirve para buscar archivos y directorios según sus permisos, tipo, fecha, propiedad, tamaño, etc.. También se puede combinar con otras herramientas como grep o sed

(g) Informar la cantidad de espacio libre en disco (df)

df

Se usa para mostrar la cantidad en porcentaje y KB de espacio libre en disco disponible.

(h) Verifique los usuarios conectado al sistema (who)

who

El comando who nos da información de los usuarios que están conectados al sistema y también otras informaciones como cuándo se arrancó el sistema y cuál es el nivel de ejecución del sistema.

(i) Acceder a el archivo iso2023-1 e ingresar Nombre y Apellido vi iso2023-1

Los pasos son los mismos que en el inciso (c)

(j) Mostrar en pantalla las últimas líneas de un archivo (tail). tail iso2023-1

El comando se utiliza para mostrar las últimas líneas de un archivo (de texto) o para restringir la salida de un comando de Linux a un ámbito concreto. A través de un parámetro se puede restringir a un número de líneas o a una cantidad de bytes.

11. Investigue su funcionamiento y parámetros más importantes:

(a) shutdown:

Se utiliza para apagar o reiniciar el sistema de manera programada o inmediata. Puede utilizarse para apagar o reiniciar el sistema de forma segura.

(b) reboot:

Este comando reinicia el sistema de forma inmediata.

(c) halt: Detiene el sistema de forma inmediata, apagando todos los procesos y dejando la máquina en un estado de apagado.

(d) locate: Busca archivos y directorios en todo el sistema de archivos basándose en una base de datos previamente creada para una búsqueda más rápida que "find".

(e) uname: Muestra información sobre el sistema operativo, como el nombre del kernel y la versión del sistema.

(f) dmesg: Muestra registros del kernel, lo que permite ver mensajes y eventos del sistema que ocurren durante el inicio y funcionamiento del sistema.

(g) lspci: Muestra información sobre los dispositivos PCI conectados al sistema, como tarjetas gráficas y adaptadores de red.

(h) at: Programa la ejecución de comandos en un momento específico en el futuro.

(i) netstat: Muestra información sobre conexiones de red, tablas de enrutamiento, estadísticas de interfaces y más.

(j) mount: Se utiliza para montar sistemas de archivos, como unidades USB o particiones, en el sistema para que sean accesibles.

(k) umount: Desmonta sistemas de archivos previamente montados, liberando recursos y desconectando dispositivos.

(l) head: Muestra las primeras líneas de un archivo de texto o la salida de un comando, por defecto las primeras 10 líneas.

(m) losetup: Se utiliza para configurar y controlar bucles de dispositivo, que permiten trabajar con archivos de imagen de disco.

(n) write: Permite enviar mensajes a otros usuarios conectados al sistema.

(ñ) mkfs: Se utiliza para crear un sistema de archivos en una partición o dispositivo.

(o) fdisk: Herramienta para gestionar particiones de disco. Debe usarse con cuidado ya que puede afectar la estructura de datos en el disco.

12. Indique en qué directorios se almacenan los comandos mencionados en el ejercicio anterior.

Todos los comando anteriormente mencionados se encuentran en el directorio **/bin**

El directorio `/bin` en Linux es un subdirectorio estándar del directorio raíz que contiene los programas ejecutables (es decir, listos para ejecutarse) que deben estar disponibles para lograr una funcionalidad mínima con el propósito de arrancar (es decir, iniciar) y reparar un sistema.

Este directorio contiene binarios para uso de todos los usuarios. Los archivos en este directorio son necesarios para el funcionamiento del sistema y pueden ser utilizados por todos los usuarios del sistema operativo.