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

GNU: Es un Sistema Operativo tipo Unix (Unix like), pero libre, S.O. diseñado por miles de programadores, S.O. gratuito y de libre distribucion (se baja desde la Web, CD, etc.). Existen diversas distribuciones (customizaciones). Es de codigo abierto, lo que nos permite estudiarlo, personalizarlo, auditarlo, aprovecharnos de la documentacion, etc...

Caracteristicas Generales:

• Es multiusuario

• Es multitarea y multiprocesador

• Es altamente portable

• Posee diversos interpretes de comandos, de los cuales algunos son programables

• Permite el manejo de usuarios y permisos

• Todo es un archivo (hasta los dispositivos y directorios)

• Cada directorio puede estar en una partici´on diferente (/temp, /home, etc.)

• Es case sensitive

• Es codigo abierto

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

/ Windows / Linux

Uso estándar: Interfaz gráfica de usuario / Líneas de comandos

Acceso remoto: Servidor de terminales; el cliente tiene que instalarse y configurarse / Solución integrada (terminal y shell)

Software y características: Soporta programas habituales; posibilidad de utilizar aplicaciones de Microsoft / No ofrece portabilidad para todos los programas; gran cantidad de aplicaciones disponibles

Soporte de hardware: El nuevo hardware está diseñado normalmente para los sistemas Windows / Por lo general, pueden utilizarse los controladores de hardware para las distribuciones de Linux más tarde

Seguridad : Elevado potencial de errores de usuario; interfaz integrada como posible punto de ataque / Los usuarios habituales no tienen acceso a los ajustes básicos del sistema; las vulnerabilidades conocidas se solucionan rápidamente

Asistencia : Asistencia a largo plazo para todas las versiones / La asistencia varía en función de la distribución y de la versión

Documentación El sistema y sus aplicaciones están muy bien documentadas, algo que difiere de los componentes de la API y de los formatos de los datos // Se conoce el código fuente completo del sistema, las API, las bibliotecas y las aplicaciones; la mayoría de manuales y de páginas informativas están en inglés

c. ¿Qué es GNU?.

GNU: Es un Sistema Operativo tipo Unix (Unix like), pero libre, S.O. diseñado por miles de programadores, S.O. gratuito y de libre distribucion (se baja desde la Web, CD, etc.). Existen diversas distribuciones (customizaciones). Es de codigo abierto, lo que nos permite estudiarlo, personalizarlo, auditarlo, aprovecharnos de la documentacion, etc...

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

GNU = GNU No es Unix

Iniciado por Richard Stallman en 1983 con el fin de crear un Unix libre (el sistema GNU).Para asegurar que el mismo fuera libre, se necesito 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 tıpico.Faltaba el componente principal → El N´ucleo (Kernel) GNU.Si bien ya se venıa trabajando en un nucleo conocido como TRIX, es en 1988 que se decide abandonarlo debido a su complejidad (corrıa en hardware muy costoso).En este momento se decide adoptar como base el nucleo MACH para crear GNU Hurd, el cual tampoco prospero.Linus Torvalds ya ven´ıa trabajando desde 1991 en un Kernel denominado Linux, el cual se distribuir´ıa bajo licencia GPL .En el año 1992, Torvalds y Stallman deciden fusionar ambos proyectos, y es all´ı donde nace GNU/Linux. GNU/Linux pertenece al desarrollo del software libre

e. Explique qué es la multitarea, e indique si GNU/Linux hace uso de ella

La multitarea es la característica de los sistemas operativos modernos que permite que varios procesos o aplicaciones se ejecuten aparentemente al mismo tiempo, compartiendo uno o más procesadores.

Linux es de multitarea y multiprocesador, porque puede haber varias tareas.

f. ¿Qué es POSIX?

POSIX es una norma escrita por la IEEE, que define una interfaz estándar del sistema operativo y el entorno, incluyendo un intérprete de comandos. 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.

2. a. ¿Qué es una distribución de GNU/Linux? Nombre al menos 4 distribuciones de GNU/- Linux y cite diferencias básicas entre ellas

Una distribucion es una customizacion de GNU/Linux formada por una version de kernel y determinados programas con sus configuraciones.

Distribuciones de Linux :

- Ubuntu: Ubuntu es una distribución de Linux basada en Debian. Está desarrollada por Canonical y una comunidad de desarrolladores. Tiene tres ediciones oficiales: Desktop, Server y Core, que pueden ejecutarse tanto en ordenador como en una máquina virtual. Más del 33 % de los sitios web Linux utilizan Ubuntu, según los datos de W3Techs. Su crecimiento desde 2010 ha sido asombroso. También es la distribución más popular entre los proyectos de cloud computing.

Debian: Debian es un sistema operativo (OS) de código abierto. Esta distribución fue anunciada inicialmente por Ian Murdock en 1993 como «Debian Linux Release». El proyecto Debian es una comunidad de desarrolladores y usuarios que mantienen el OS GNU basado en software de código abierto. Actualmente, los sistemas Debian utilizan el kernel de Linux o el kernel de FreeBSD. Sin embargo, también están trabajando en ofrecer Debian para otros kernels. Principalmente, GNU Hurd.

CentOS: CentOS es una distribución de Linux basada en el código fuente de la distribución comercial Red Hat Enterprise Linux (RHEL). Se lanzó en 2004 y está respaldado por una creciente comunidad. Es una apuesta segura para los que buscan un código de gran calidad. Pero CentOS 8 será su última versión. En 2019, Red Hat anunció que CentOS Linux sería reemplazado por CentOS Stream —una plataforma de desarrollo upstream para RHEL—. A partir de este cambio de estrategia han aparecido nuevas alternativas open source; como Rocky Linux, fundado por Gregory Kurtzer, fundador del proyecto CentOS.

Red Hat Enterprise Linux (RHEL): Red Hat Enterprise Linux (RHEL) es una distribución comercial de Linux desarrollada por Red Hat. Tiene una versión de servidor y una versión de escritorio. Como utiliza un software de código abierto, publicado bajo Licencia Pública General, ponen el código a disposición del público a través de CentOS. Red Hat ha patrocinado el proyecto CentOS desde 2014.

b. ¿En qué se diferencia una distribución de otra?

Existen múltiples distribuciones Linux diferentes. Muchos tienen diferentes filosofías: algunos, como Fedora, se niegan a incluir software de código cerrado, mientras que otros, como Mint, incluyen material de código cerrado para que sea más fácil para los usuarios. Incluyen diferentes software predeterminados, como Ubuntu incluye Unity, Ubuntu incluye otros entornos de escritorio, Fedora incluye GNOME Shell y Mint incluye Cinnamon o MATE.

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

Debian es un sistema operativo y una distribución de Software Libre. El proyecto Debian es una asociación de personas que han hecho causa común para crear un sistema operativo (SO) 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. a. Nombre cuales son los 3 componentes fundamentales de GNU/Linux

Directorios mas importantes segun FHS (Filesystem Hierarchy Standard)

• / Tope de la estructura de directorios. Es como el C:\

• /home Se almacenan archivos de usuarios (Mis documentos)

• /var Informacion que varıa de tamaño (logs, BD, spools)

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

/Raíz el equivalente al C: de windows

/bin Binarios y ejecutables del sistema.

/boot Aquí se encuentra lo mas importante del sistema; el kernel y los archivos necesarios para que el sistema funcione correctamente.

/dev Archivos que se relacionan con los diferentes dispositivos que pueden estar funcionando en una PC (discos duros, terminales, sonido, video, lectores dvd/cd, etc) /etc Uno de los mas importantes, porque se encuentran todas las configuraciones del sistema.

/home Aquí cada usuario posee un directorio donde se guarda toda su configuración, los programas instalados y documentos personales.

/usr Jerarquía secundaria para datos compartidos de solo lectura (Unix system resources).

/root Similar al /home pero para el usuario root (solo puede ser visto por el administrador)

/var Archivos variables, como son logs, bases de datos, directorio raíz de servidores HTTP y FTP, colas de correo, archivos temporales, etc.

4. a. ¿Qué es? Indique una breve reseña histórica acerca de la evolución del Kernel de GNU/Linux.

En informática, un núcleo o kernel es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado.

Historia:

En 1991 Linus Torvalds inicia la programacion de un Kernel Linux basado en Minix (clon de Unix desarrollado por Tenembaum en 1987 con el fin de crear un S.O. de uso didactico). El 5 de octubre de 1991, se anuncia la primera version “oficial” de Linux (0.02) . En 1992 se combina su desarrollo con GNU, formando GNU/Linux . La version 1.0 aparecio el 14 de marzo de 1994.Desarrollo continuado por miles de programadores al rededor del mundo

En mayo de 1996 se decide adoptar a Tux como mascota oficial de Linux. En julio de 1996 se lanza la version 2.0 y se define la nomenclatura de versionado. Se desarrollo hasta febrero de 2004 y termino con la 2.0.40. En enero de 1999 se lanza la version 2.2, que provee mejoras de portabilidad entre otras y se desarrolla hasta febrero de 2004 terminando en la version 2.2.26. En 2001 se lanza la version 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 mas asiduamente

A fines del 2003 se lanza la version 2.6. Esta version ha tenido muchas mejoras para el SO dentro de las que se destacan soporte de hilos, mejoras en la planificacion y soporte de nuevo hardware. El 3 de agosto de 2011 se lanza la version 2.6.39.4 anunciandose la misma desde meses previos como la ´ultima en su revision . El 17 de julio de 2011 se lanza la version 3.01. No agrega mayores cambios. La decision del cambio son los 20 años del SO y no superar los 40 numeros de revision . Totalmente compatible con 2.6. La última version estable es la 5.13.12 (agosto de 2021)

b. ¿Cuáles son sus funciones principales?

El kernel cumple cuatro tareas:

1.Gestión de la memoria: supervisa cuánta memoria se utiliza para almacenar qué tipo de elementos, así como el lugar en que los guarda.

2.Gestión de los procesos: determina qué procesos pueden usar la unidad central de procesamiento (CPU), cuándo y durante cuánto tiempo.

3.Controladores de dispositivos: actúa como mediador o intérprete entre el hardware y los procesos.

4.Seguridad y llamadas al sistema: recibe solicitudes de servicio por parte de los procesos.

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 version estable actual es la 5.13.12. Antes de la versión 2.6, los

números impares indicaban desarrollo, los pares producción

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

Podemos tener en un sistema Linux por ejemplo, dos kernels instalados cada uno de una versión distinta. Eso sí, solo ejecutaríamos uno a la vez, el otro quedaría instalado y almacenado en el disco. También podríamos emular varios kernels a partir de un mismo sistema operativo base y ejecutarlos a la vez, aunque serían kernels virtuales y no estarían funcionando realmente a nivel sistema

e. ¿Dónde se encuentra ubicado dentro del File System?

El kernel se encuentra en /boot junto con archivos necesarios para que el sistema funcione correctamente.

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

Es un núcleo monolítico híbrido:

• Los drivers y código del Kernel se ejecutan en modo privilegiado

• Lo que lo hace híbrido es la capacidad de cargar y descargar funcionalidad a través de módulos

5. Intérprete de comandos (Shell):

a. ¿Qué es?

Un intérprete de órdenes o de comandos, es un programa informático que tiene la capacidad de traducir las órdenes que introducen los usuarios, mediante un conjunto de instrucciones facilitadas por él mismo directamente al núcleo y al conjunto de herramientas que forman el sistema operativo.​

b. ¿Cuáles son sus funciones?

Sirve de modo de comunicación entre el usuario y el SO. Ejecuta programas a partir del ingreso de comandos

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

entre ellos.

Bourne Shell (sh), Korn Shell (ksh), Bourne Again Shell(bash)(autocompletado, history, alias)

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

- usr/bin/

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

La confiabilidad de la Shell y la capacidad de ser remplazada son los principales motivos para no tener la Shell en 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, cada usuario puede tener una interfaz o shell. Esto se define al crear el usuario, o en archivo /etc/passwd, el archivo /etc/passwd solo puede ser modificado por un usuario root.

6. Sistema de Archivos (File System)

a. ¿Qué es?

Un sistema de archivos o sistema de ficheros, en informática, es un componente que controla cómo se almacenan y recuperan los datos. Sin un sistema de archivos, los datos colocados en un medio de almacenamiento serían un gran cuerpo de datos sin manera de saber dónde termina un dato y comienza el siguiente.

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

El adoptado por GNU/Linux es el Extended (v2, v3, v4), pero soporta gran variedad de sistemas de ficheros, como pueden ser ext2, ext3, ReiserFS, XFS, JFS, UFS, ISO9660, FAT, FAT32 o NTFS, etc.

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

Linux 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. En modo escritura todavía está en fase experimental y no es estable.

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?

Los sistemas de ficheros de Linux y Unix se organizan en una estructura jerárquica, de tipo árbol. El nivel más alto del sistema de ficheros es / o directorio raíz. ... Por debajo del directorio raíz ( / ) hay un importante grupo de directorios común a la mayoría de las distribuciones de GNU/Linux.

Directorios más importantes según FHS (Filesystem Hierarchy Standard)

• / Tope de la estructura de directorios. Es como el C:\

• /home Se almacenan archivos de usuarios (Mis documentos)

• /var Informaci´on que varía de tamaño (logs, BD, spools)

• /etc Archivos de configuración

• /bin Archivos binarios y ejecutables

• /dev Enlace a dispositivos

• /usr Aplicaciones de usuarios

FHS -> Filesystem Hierarchy Standard (Jerarquía estándar de un sistema de archivos)

7. Particiones:

a. Definición. Tipos de particiones. Ventajas y Desventajas.

Una partición es una división lógica en una unidad de almacenamiento

Tipos de particiones: las primarias, las extendidas o secundarias, y las lógicas.

VENTAJAS:

• La principal ventaja de particionar un disco duro es el tener una partición exclusiva para los archivos personales del usuario, es ideal a la hora de instalar de cero el sistema operativo. En una de las particiones se mantiene el sistema operativo y en la otra los archivos del usuario (documentos, e-mails, descargas, música, etc).

• Si se necesita formatear e instalar de cero el sistema por cualquier inconveniente, simplemente se procede a formatear la partición que contiene el sistema operativo, dejando intacta la otra. Esto nos ahorra mucho tiempo, dado que no debemos guardar toda esa información en otra unidad de almacenamiento.

DESVENTAJAS:

• Si se usa una de las particiones como respaldo, no es muy seguro que digamos. Si se rompe el disco duro, se pueden ver afectadas ambas particiones. Recuerde que ambas están en el mismo disco físico, aunque en diferentes áreas de éste.

• Tampoco particionar aumenta la velocidad del disco duro, de hecho, puede producir un mayor desgaste del mecanismo interno del disco. Es es porque cuando se particiona un disco duro en dos, se crean dos áreas de trabajo. Es decir, dos anillos concéntricos, siendo el anillo interior el de la unidad C, y el anillo externo generalmente el de la unidad D.

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

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

Minimo 1 particion para el directorio de raíz y otra para el swap. /swap La partición de intercambio es donde se extiende la memoria del sistema al dedicar parte del disco duro a esta. / La barra / sola representa la raíz del árbol del filesystem.

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

Básicamente se pueden considerar 3 tipos:

• Emulación:

• Emulan hardware

• Tienen que implementar todas las instrucciones de la CPU

• Es muy costosa y poco eficiente

• Permite ejecutar arquitecturas diferentes a las soportadas por el hardware

• Virtualización completa:

• Permiten ejecutar SO huéspedes en un sistema anfitrión (host)

• Utilizan en el medio un hypervisor o monitor de máquinas virtuales

• El SO huésped debe estar soportado en la arquitectura anfitriona

• Es más eficiente que la emulación (Intel-VT y AMD-V)

• Paravirtualizacion:

• Permite correr SOs modificados exclusivamente para actuar en entornos virtualizados

• Mayor eficiencia que la virtualización

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

Para crearlas, 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) ← generalmente las distribuciones permiten hacerlo desde la interfaz de instalación

8. Arranque (bootstrap) de un Sistema Operativo:

a. ¿Qué es el BIOS? ¿Qué tarea realiza?

El BIOS (Basic Input Output System) es un software de bajo nivel que se halla en el motherboard. Cuando se arranca la computadora el BIOS se ejecuta, realizando el POST (Power-on self-test), que incluye rutinas que, entre otras actividades, fijan valores de las señales internas, y ejecutan test internos.

b. ¿Qué es UEFI? ¿Cuál es su función?

UEFI (Unified Extensible Firmware Interface) o Interfaz de Firmware Extensible Unificada es una tecnología que controla el hardware de la computadora cuando se prende y que sustituye a BIOS. es el encargado de gestionar el hardware de la computadora y controlar los distintos periféricos, la memoria RAM, las revoluciones de los ventiladores, etc.

c. ¿Qué es el MBR? ¿Que es el MBC?

-MBR (master boot record)

-MBC (master boot code)

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

las siglas hacen referencia a Guid partition table. Puede verse como una sustitución del MBR como era pensado en la BIOS. GPT usa modo de direccionamiento lógico (LBA, logical block addressing) en vez del modo cilindro-cabeza-sector usado con el MBR. El MBR “heredado” se almacena en el LBA 0.

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

Es el bootloader que este es la de cargar una imagen de Kernel (sistema operativo) de alguna partición para su ejecución. Los tipos que existen son GRUB, LILO, NTLDR, GAG, YaST, etc.

Existen 2 modos de instalación:

• En el MBR (puede llegar a utilizar MBR gap)

• En el sector de arranque de la partición raíz o activa (Volume Boot Record)

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

Carga el programa de booteo (desde el MBR)

• El gestor de arranque lanzado desde el MBC carga el Kernel:

• Prueba y hace disponibles los dispositivos

• Luego pasa el control al proceso init

• El proceso de arranque se ve como una serie de pequeños programas de ejecución encadenada

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

El proceso de arranque de un sistema linux consta de seis etapas:

BIOS: Sistema básico de configuración. La bios inicia MBR.

MBR: Registro de arranque maestro que inicia el Grub.

GRUB: Cargador de arranque que ejecuta el Kernel.

Kernel: Inicia el /sbin/init.

Init: Lanza los programas de Runlevel.

RunLevel: Aquí encontramos las aplicaciones por nivel de ejecución (la ruta puede variar, pero normalmente » /etc/rc.d/rc\*.d/».

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

A grandes rasgos, cuando se lanza shutdown, se notifica a los usuarios de este hecho y, además, se bloquea el sistema para que nadie más pueda acceder —creando el archivo /etc/nologin—, exceptuando el root. Acto seguido, se envía la señal SIGTERM a todos los procesos no definidos en inittab para el siguiente run level, provocando que terminen su ejecución de modo ordenado. Poco después, se envía una señal SIGKILL para que los procesos que no hayan atendido a SIGTERM concluyan también su ejecución —pero en este caso no de una manera “limpia”—. shutdown lleva a cabo su cometido enviando una señal a init para que cambie a uno de estos niveles de ejecución, en función del efecto que se desee conseguir: Apagar o detener el sistema (nivel 0, opción “-r”). Entrar en modo monousuario (nivel 1, opción por defecto). Reiniciar el sistema (nivel 6, “-h”). Una vez cambiado el nivel de ejecución, si procede, se invoca a halt, reboot o poweroff según sea necesario.

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

Es posible. Para ello debemos tener particionado nuestro disco duro o tener varios discos duros instalados en la PC. De esta manera podemos instalar copias de los sistemas operativos en las distintas partes de los discos

9. Archivos:

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

Los archivos se identifican por su nombre (obviamente), en Linux no hay ningún formato estándar para los nombres de los archivos, en general, pueden tener cualquier carácter menos "/" y están limitados a 256 caracteres de longitud.

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

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

mcedit: editor de archivos interno de GNU Midnight Commander.

mcedit es un enlace a mc , el ejecutable principal de GNU Midnight Commander. Al ejecutar GNU Midnight Commander con este nombre, se solicita mirar al editor interno y abrir el archivo especificado en la línea de comandos. El editor se basa en la versión de terminal de cooledit - editor independiente para X Window System.

El comando "cat" (concatenar) se usa con mucha frecuencia en Linux. Lee los datos del archivo y da su contenido como salida. Nos ayuda a crear, visualizar, concatenar archivos. Así que veamos algunos comandos de gato de uso frecuente.

El comando "more" es una utilidad de línea de comandos para ver el contenido de un archivo o archivos una vez, pantalla a la vez. Admite la navegación hacia adelante y hacia atrás a través de un archivo y se utiliza principalmente para ver el contenido de un archivo. También admite la búsqueda de cadenas o expresiones regulares y la apertura del archivo en el punto actual en un editor de texto.

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.

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

El comando file es una utilidad que realiza una serie de pruebas (test) para determinar el tipo y formato de un archivo.

**La invocación del comando file tiene el siguiente formato:**

* **file [PARÁMETROS] *ARCHIVO*...**

Los parámetros posibles, según la especificación [POSIX](https://es.wikipedia.org/wiki/POSIX), son:

**-d →** Realiza las pruebas de sintaxis y de números mágicos del sistema. Esta es la opción default, sino se utiliza *-m* o *-M*

**-h →** Si el archivo a analizar es un enlace simbólico, lo identifica como tal.

**-i →** No brindar una clasificación adicional más allá de la del tipo básico.

**-m *archivo →*** Realiza una prueba adicional de números mágicos con el archivo indicado

**-M *archivo →*** Similar a *-m*, salvo que no realiza las pruebas de sintaxis y de números mágicos por defecto del sistema.

En los sistemas operativos [GNU/Linux](https://es.wikipedia.org/wiki/GNU/Linux), este comando fue enriquecido con muchos más parámetros, entre los cuales se encuentran:

**-b →** No imprime el nombre del archivo en cada resultado.

**-i →** Muestra el tipo mime junto con la codificación utilizada.

**-e *prueba →*** Excluye de realizar la prueba indicada. Entre las opciones que podemos indicarle, se encuentran:

**apptype**

Tipo de aplicación EMX (solo para EMX).

**ascii**

Varios tipos de archivos de texto. Esta prueba intenta determinar la codificación, más allá de la indicada dentro del propio archivo.

**encoding**

Varios tipos de codificaciones para la prueba suave de números mágicos.

**tokens**

Busca cadenas conocidas dentro de los archivos de texto.

**cdf**

Muestra detalles de los archivos CDF (Compound Document Files). POr ejemplo SVG, XHTML, etc.

**compress**

Analiza y busca dentro de los archivos comprimidos.

**elf**

Muestra detalles de archivos ELF.

**soft**

Consulta de archivos mágicos.

**tar**

Analiza archivos [Tar](https://es.wikipedia.org/wiki/Tar).

**--mime-type →** similar a *-i*, salvo porque solo muestra el tipo mime.

**-z →** Examina los archivos comprimidos.

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 redictadoISO = mkdir "nombre de carpeta"

* **-m, --mode=modo**. Establece los permisos de los directorios creados a modo.
* **-p, --parents**. Crea los directorios padre que falten para cada argumento directorio. No hace caso de argumentos que correspondan a directorios existentes, por ejemplo si existe un directorio /a, entonces mkdir /a es un error, pero mkdir -p /a no lo es.
* **--verbose**. Muestra un mensaje para cada directorio creado.

(b) Acceda a la carpeta (cd) = cd "nombre de carpeta" o cd .. para volver atrás en el arbol del filesystem.

(c) Cree dos archivos con los nombres redictadoiso-1 y redictadoiso-2 (touch) = touch [opcion] archivo…

los argumentos pueden ser -a, -c, -d fecha

-a = cambia el tiempo de acceso

-c= No crea archivos que no existian antes.

-d fecha ( formato = MMDDHHMMAAAA = mes dia hora minutos año) = cambia la fecha del archivo

si se llama sin argumentos edita o crea los archivos.

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

Lista el contenido de un determinado directorio o, si no le indicamos el directorio, lista el contenido del directorio actual. Además de listar el contenido se le pueden pasar distintos parámetros para extender la información que muestra.

ls -a

Nos muestra los archivos y directorios dentro del directorio actual, incluyendo los archivos y directorios ocultos.

ls -t

Ordena los archivos por fecha de modificación.

ls -X

Ordena los archivos por extensión.

ls -l

Muestra toda la información: usuario, grupo, [permisos](https://www.servidoresadmin.com/comando-chmod-en-linux-modificacion-de-permisos-en-linux/), tamaño, fecha y hora de creación.

ls -lh

Muestra la misma información que ls -l pero con las unidades de tamaño en KB, MB, etc.

ls -R

Muestra el contenido de todos los [subdirectorios](http://es.wikipedia.org/wiki/Directorio) de forma recursiva.

ls -S

Ordena los resultados por tamaño de archivo.

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

con el argumento -P vemos la ruta fisica ejemplo:

*estamos en -> / var / run*

***pwd →***  */ var / run*

***pwd -P →*** */ run*

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

find [ruta] [expresión\_de\_búsqueda] [acción]

find. -name “\*iso\*”

**ejemplos de uso:**

busca desde raiz archivos que tengan de nombre "reporte"

**find / -name reporte**

**find / -iname Reporte (lo mismo, pero sin tomar en cuenta mayúsculas y minúsculas)**

El uso de expresiones regulares en lo que se busca es válido:

**find / -name "[0-9]\*" (todo lo que empieze con un dígito)**

**find / -name "[Mm]\*" (todo lo que empieze con un la letra M o m)**

**find / -name "[a-m]\*.txt" (todo lo que empieze entre a y m y termine en ".txt")**

Busca bajo /home todos los archivos que pertenezcan al usuario mario:

**find /home -user mario**

(lo mismo y que contengan con "enero" como en reporte\_enero2011)

**find /home -user mario -name "\*enero\*"**

No estás limitado a un solo directorio, indica más de uno a buscar antes de las expresiones:

**find /etc /usr /var -group admin**

(busca en tres directorios todos los archivos o subdirectorios que pertenezcan al grupo 'admin')

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

Al ejecutarse sin opciones, el comando muestra el espacio disponible en todos los sistemas de archivos montados al momento. El espacio en disco se expresa en bloques de 1Kb por defecto. Por ello, es aconsejable la opción -h, para que facilite la lectura en Gb, Mb o Kb. Si en algún momento nos interesa saber el espacio expresado específicamente en Mb utilizamos la opción -m.

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

who [options] [filename]

|  |  |  |
| --- | --- | --- |
| -a | --all | Todas las opciones más utilizadas a la vez: -b, -d, --login, -p, -r, -t, -T, -u. |
| -b | --boot | Fecha y hora del último inicio del sistema. |
| -d | --dead | Imprime los procesos muertos. |
| -H | --heading | Encabezados explicativos en cada columna. |
|  | --ips | Si es posible se imprimirá la ip y no el host. |
| -l | --login | Muestra los procesos del login del sistema. |
|  | --lookup | Si es posible canonicalizara los nombres de host a través de DNS. |
| -m |  | Imprime sólo el nombre del host y del usuario asociado con la entrada (terminal donde se ejecuto el comando). |
| -p | --process | Imprime los procesos activos generados por init. |
| -q | --count | Nos muestra el número y nombre de los usuarios conectados. |
| -r | --runlevel | Imprime el nivel de ejecución actual (runlevel). \*\*\*Ver NOTA al final del articulo\*\*\* |
| -s | --short | Imprime el nombre, tipo, y fecha con horario. |
| -t | --time | Si es posible imprimirá la ultima vez que se modificó el reloj del sistema. |
| -T -w | --mesg | Agrega un caracter que indica el estado de la terminal. Si se puede escribir (+), si no se permite (?). |
| -u | --users | Imprime el tiempo de cada usuario y la ID de proceso. |

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

se puede utilizar un editor de texto, o uno integrado a la consola, como puede ser vi, vim, emacs, nano… (nano es el mas sencillo)

se abre el archivo con el nombre del editor seguido del archivo.

Ejemplo:

nano redictadoiso-1

vi redictadoiso-1

etc.

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

tail [opciones] [archivo]

**tail "archivo.txt" // muestra las ultimas 10 lineas del archivo.**

**tail -n <NÚMERO> archivoejemplo.txt // muestra las ultimas "NÚMERO" lineas del archivo.**

**tail -c <NÚMERO> archivoejemplo.txt // muestra los ultimo "NÚMERO" bytes del archivo.**

Si se ponen 2 archivos muestra las ultimas 10 lineas de ambos archivos, o las especificadas por los argumentos.

**El argumento -f monitorea cambios en el archivo.**

11. Investigue su funcionamiento y parámetros más importantes:

(a) shutdown :

shutdown [OPTIONS] [TIME] [MESSAGE]

Apaga la computadora, si no se especifican paraemtros, se apagara tras un minuto.

shutdown now (apaga la pc inmediatamente)

shutdown hh:mm (apaga la pc a una hora especifica, formato 24hs)

shutdown +m (apaga la pc despues de m cantidad de minutos)

shutdown +m "mensaje" (apaga la pc despues de m cantidad de minutos, mostrando un mensaje especificado antes de apagarse)

el argumento de mensaje se puede pasar con las otras opciones tambien.

**Argumento "-r" o "-reboot" reinicia la pc, ejemplo:**

**shutdown -r +30 "mensaje" (reinicia la pc en 30 minutos mostrando un mensaje antes)**

**argumento "-p" o "-poweroff" apaga el sistema ( apagando tambien la unidad )**

**argumento "-h" o "-halt" detiene la maquina despues de terminar los procesos.**

**Argumento -c**

sirve para cancelar un apagado o renicio programado con anterioridad. Ejemplo:

**shutdown -c**

(b) reboot:

igual que el shutdown -r

se puede usar reboot de la misma forma. Necesita privilegios de root.

(c) halt:

Halt toma nota de que el sistema se va a venir abajo en el archivo / var / log / wtmp , y luego o bien le dice al núcleo que apague el sistema. Si halt es llamado cuando el sistema no está en el nivel de ejecución 0 o 6 , o sea cuando está funcionando normalmente , el apagado se invocará en su lugar ( con la opción- h o -r flag) .

**halt [OPTION]...**

**-n No sincronizar antes de detener. Tenga en cuenta que los controladores del núcleo y de almacenamiento pueden todavía sincronizar.**

**-w en realidad no para o reinicia, pero sólo escriben el registro wtmp (en el directorio / var / log / wtmp ) .**

**-d No escriba el registro wtmp . El indicador-n implica - d.**

**-f Fuerza halt, no llama a** [**shutdown**](https://www.ecured.cu/Shutdown)**.**

**-i Apaga todas las interfaces de red justo antes de detener o reiniciar el sistema.**

**-h Ponga todos los discos duros en el sistema en modo de espera antes de detenerse o de apagarse .**

**-p Cuando apagamos el sistema, desconecte la alimentación. Este es el valor por defecto cuando halt es llamado como** [**poweroff**](https://www.ecured.cu/Poweroff) **.**

(d) locate

El comando locate es la forma más práctica de buscar un archivo en Linux, este comando locate se encarga de realizar las búsquedas haciendo uso de una base de datos donde se encuentra toda la información y locate la consulta a ella directamente.

**Uso: locate "palabra clave"**

**busca todos los archivos que contengan palabra clave o que coincidan**

**argumento -c hace un conteo de cuantos contienen la palabra pasada por argumento**

**argumento -i ignora case (es casesensitive por default)**

**argumento -e comprueba que el archivo aun exista en el sistema (ademas de la base de datos, que puede estar desactualizada)**

(e) uname

Si utilizamos el comando sin argumentos nos entregara la palabra Linux extraída de la información del Kernel.

Literalmente imprime "Linux"

**Si deseamos** extraer la información de la versión del kernel **utilizamos el parámetro** -r**.**

**Si deseamos** extraer la fecha de cuando la versión del kernel **fue liberada utilizamos el parámetro** -v**.**

**Para** conocer la arquitectura de nuestro sistema operativo**, ya sea 36 o 64 bytes, utilizamos el parametro** -m

Para mostrar la información del procesador del equipo en el cual se ejecuta nuestro sistema operativo linux, utilizamos el parámetro -p.

**Para** obtener la información del nombre de la distribución de Linux **que estamos utilizando, utilizamos el parámetro** -o**.**

Para obtener la información completa en un solo parámetro, usamos el parámetro -a.

(f) gmesg

(g) lspci

(h) at

(i) netstat

(j) mount

(k) umount

(l) head

(m) losetup

(n) write

(ñ) mkfs

(o) fdisk (con cuidado)

(ñ) mkfs Comando Linux MKFS, se  utiliza para dar formato a un dispositivo de almacenamiento de bloque con un determinado sistema de archivos.  
Un Sistema de archivos es la estructura básica de toda los datos que se guarda, edita, borra o copia etc…en el ordenador, siendo toda esta información accesada a través de gestores de archivos en sus respectivos SO.  
ext – Extended file system (sistema de archivos extendido)  
ext2 – Second extended file system (segundo sistema de archivo extendido)  
ext3 – Third extended file system (tercer sistema de archivos extendido), una forma de ext2 con registro en diario. ext4 – Fourth extended file system (cuarto sistema de archivos extendido), una mejora de ext3 y también un sistema de archivos con registro en diario con soporte para extensiones. -v  
Producir una salida detallada, incluyendo todas las órdenes específicas del sistema de archivos que se ejecutan. Especificar esta opción más de una vez inhibe la ejecución de los comandos específicos del sistema de archivos. Esto sólo es realmente útil para la prueba. -t fstype  
Especifica el tipo de sistema de archivos que se construirá. Si no se especifica, se utiliza el tipo de sistema de archivos por defecto (actualmente ext4). fs-opciones  
Opciones del sistema de archivos específicas que se pasan al constructor del sistema de archivos real. Aunque no se garantiza, las siguientes opciones se admiten en la mayoría de los fabricantes de sistemas de archivos. -c  
Compruebe el dispositivo de bloques malos antes de construir el sistema de archivos. -l nombre de archivo  
Lea la lista de bloques malos desde nombre de archivo -h, – help  
Pantalla de ayuda y sale.