Trabajo Práctico Nº 1
Introducción a GNU/Linux

1.- Particiones:

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

Particionar el disco duro es una manera de dividir el disco físico en varios discos lógicos. O lo que es lo mismo, al particionar un disco, dividimos el disco en varias particiones independientes unas de otras, creando la ilusión de que tenemos diferentes discos, cuando en realidad lo que tenemos es un solo disco físico dividido en partes.
La tabla de particiones situada en el MBR restringe a 4 la cantidad posible de particiones primarias en un disco. Sin embargo, es posible crear más de 4 particiones utilizando particiones extendidas.
Las particiones extendidas son similares a un nuevo disco, ya que poseen sus propia tabla de particiones que apunta a una o más particiones (ahora llamadas particiones lógicas) contenidas íntegramente en la partición extendida.
A diferencia de las particiones primarias, las particiones extendidas deben ser contiguas. La razón de este hecho se debe a que cada partición extendida contiene un puntero a la siguiente partición extendida, lo que implicaría que el número de este tipo de particiones sería ilimitado. Sin embargo, Linux impone un límite en este número. Por ello se limita a 15 el número de particiones extendidas sobre discos SCSI y a 63 en discos IDE.
Cualquier disco que tengamos en nuestro ordenador tiene al menos una partición primaria, que en la mayoría de los casos tiene un tamaño equivalente al total del disco.
Desventajas de tener el disco dividido en diferentes particiones:

- Tener el disco muy particionado puede pasar a ser malo, porque quizás tenemos, un archivo o un programa muy grande que guardar, y de tan chicas que son las particiones, porque dijimos que teníamos muchas, entonces, capaz no podemos guardar ese archivo o programa.
- Agrega más complejidad para el usuario

Ventajas en tener el disco particionado en varias particiones:

- Si ocurre un error/problema en una de ellas, las demás no se verán afectadas.
- Poder tener diferentes sistemas operativos en una sola maquina, totalmente independientes unos de otros.
- Poder tener sus archivos de datos en particiones totalmente independientes.
- Poder borrar/cambiar el contenido de una partición, sin que esto afecte a las demás.
- Aumentar la seguridad del sistema

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

Al contrario que Ms-Dos, Windows, OS/2, las diferentes particiones en linux no se denominan C:,
D:,
E:, etc, existe una denominación propia.

DISCOS IDE:

/dev/hda: Disco duro IDE como master en el canal IDE 1.
/dev/hda1: Partición primaria 1 en
/dev/hda
/dev/hda2: Partición primaria 2 en
/dev/hda
/dev/hda3: Partición primaria 3 en
/dev/hda
/dev/hda4: Partición primaria 4 en
/dev/hda
/dev/hda5: Partición extendida 1 en
/dev/hda
/dev/hda6: Partición extendida 2 en
/dev/hda
.....
/dev/hda16: Partición extendida 16
en /dev/hda
/dev/hdb: Disco duro IDE como esclavo en el canal IDE 1.

/dev/hdb1: Partición primaria 1 en
/dev/hdb
.....
/dev/hdc: Disco duro IDE como master en el canal IDE 2.
/dev/hdc1: Partición primaria 1 en
/dev/hdd: Disco duro IDE como esclavo en el canal IDE 2.
/dev/hdd1: Partición primaria 1 en
/dev/hdd
.....
.....
DISCOS SCSI:
/dev/sda: Disco duro SCSI número1.
/dev/sda1: Partición primaria 1 en /dev/sda
.....
.....
/dev/sdb: Disco duro SCSI número2.
/dev/sdb1: Partición primaria 1 en /dev/sdb
.....

c) ¿Cuantas particiones son necesarias como mínimo para instalar GNU/Linux? ¿Cuáles?
Básicamente podemos decir que para instalar Linux necesitaremos 3 particiones, pero como mínimo con 2 ya estaría bien (/ y SWAP): una para el sistema/datos, otra para albergar el kernel y otra para Swap. Usualmente se suelen tener más, una para el sistema/programas (/), otra para el kernel (/boot), otra para swap, otra para los archivos de los usuarios (/home), otra para la información temporal (/tmp), otra para la información variable (/var), etc. Esta decisión dependerá del uso que le quiera dar a su sistema. Para sistemas que se utilicen de forma particular y por uno o pocos usuarios bastara con tres particiones antes mencionadas, esto evitara los problemas de saber que cantidad de espacio necesitan las diferentes particiones y el quedarnos sin espacio en alguna partición vital, mientras que nos sobra en otras. Para sistemas servidores, con gran cantidad de servicios y usuarios es muy recomendable tener varias particiones/discos.

d) Ejemplifique diversos casos de particionamiento dependiendo del tipo de tarea que se deba realizar en su sistema operativo
Para máquinas de usuarios novatos, equipos personales, sistemas para el hogar y en general, cuando será usado por un solo usuario, probablemente una partición en / (además de la de intercambio), es la forma más fácil de comenzar.
Para sistemas multiusuario, o bien, con una gran cantidad de espacio libre para usar, lo mejor es dejar /usr, /var, /tmp y /home separadas de la partición /. Si la máquina va a ser un servidor de correo, se necesitará crear una partición separada para /var/mail. A menudo, dejar una partición separada para /tmp es una muy buena idea. Es suficiente dejarla con 20 ó 50 MB.
Si se está configurando un servidor con una gran cantidad de cuentas de usuarios, es altamente recomendable dejar separada la partición /home. En general, cómo se debe particionar el disco depende del uso que se quiera dar al ordenador.
Con respecto al tamaño de la partición de intercambio, hay diversos puntos de vista. Uno de ellos, que generalmente funciona muy bien, es asignar tanto espacio a la partición de intercambio como memoria tenga el sistema. En la mayoría de los casos no es recomendable que sea más pequeña que 16 MB. Obviamente, hay excepciones a estas reglas. Si se está tratando resolver 10000 ecuaciones simultáneas en una máquina con 256 MB de memoria, necesitará un gigabyte (o más) de espacio de intercambio. En arquitecturas de 32 bits (i386, m68k, 32-bit SPARC y PowerPC), el tamaño máximo para una partición de intercambio es de 2 GB. Esto debería ser suficiente para cualquier tipo de instalación. Sin embargo, si los requisitos para el espacio de intercambio son realmente altos, probablemente se debería tratar de separar la carga del intercambio, a través de diversos discos (también llamados “spindles”) y, si es posible, en diferentes canales SCSI o IDE. El núcleo podrá balancear el uso de la zona de intercambio entre múltiples particiones de intercambio dando así un mejor rendimiento.

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

5.4. Software para Particionar

Existe una amplia variedad de software para realizar el particionamiento. Algunos programas realizan un particionamiento destructivo y otros no.

_ **Fdisk**: Este es el software más conocido y utilizado, debido a que es distribuido gratuitamente con cualquier versión de DOS/Windows y Linux. Se emplea de una forma muy sencilla desde la línea de comandos, y luego presenta una serie de menús para realizar las tareas de particionamiento. La desventaja de fdisk es que es un particionador destructivo, por lo cual no le permitirá cambiar el tamaño de una partición ya creada. Para utilizar este programa, la forma más sencilla es crear un disco de inicio en DOS/Windows, iniciar el sistema booteando con el disco de inicio y luego escribir fdisk. Esta puede ser una práctica interesante para que se familiarice con el software de particionamiento. No tenga miedo de utlizarlo, ya que antes de realizar cualquier cambio sobre su disco rígido el mismo sistema le consultará seguidamente.

_ **FIPS** (**F**irst nondestructive **I**nteractive **P**artition **S**plitting program): Este es un programa diseñado para dividir una partición del tipo Fat sin la necesidad de perder los datos. Antes de poder utilizar Fips, se deberá defragmentar la partición que se desea dividir, de manera tal que los datos queden contiguos en el inicio del disco. Para defragmentar el disco utilice la herramienta defrag que viene incluida con su distribución de Windows. Fips sólo dividirá la partición si es que existe suficiente espacio disponible al final de la misma. Además se deberá realizar un escaneo del disco antes de utilizar fips. La herramienta más adecuada para hacerlo es scandisk.

Básicamente la tarea de fips es cambiar los valores de inicio y fin de una partición en la tabla de particiones y en el sector de booteo, reduciendo una partición existente y creando una nueva partición primaria con el espacio que se le sacó a la primera partición. Para utilizar esta nueva partición, antes debe ser formateada. Lo primero que mostrará fips cuando lo ejecute es la información de la tabla de particiones.

_ **Software Comercial**: Este software le permitirá particionar su disco duro, e incluso dividir una partición ya existente. Existen distintos tipos de software comercial para particionar (destructivos y no destructivos). Dentro de los más conocidos podemos enumerar al Partition Magic, al System Commander, etc. Todos estos se instalan en entorno Windows y proveen de un menú gráfico para realizar la tarea. Es importante recalcar que muchos de ellos se rigen por licencias propietarias y no libres y que además la mayoría de ellos tienen un costo para su utilización.

1) Particiones

Una partición es una forma de dividir lógicamente el disco físico:

- DOS y W95 no pueden manejar filesystems mayores a 2 GB
- Cada sistema operativo es instalado en una partición separada
- Cada partición se formatea con un tipo de File System distinto (fat, ntfs, ext, etcétera)
- Es una buena práctica separar los datos del usuario de las aplicaciones y/o Sistema Operativo instalado
- Tener una partición de Restore de todo el sistema
- Poder ubicar el Kernel en una partición de solo lectura

Debido al tamaño acotado en el MBR para la tabla de particiones:

- Se restringe a 4 la cantidad de particiones primarias o 3 particiones primarias y una extendida con sus volúmenes lógicos
- Una de las 4 particiones puede ser extendida, la cual se subdivide en volúmenes lógicos
- Para Linux se necesitan al menos 2 (/ y SWAP)
- Para crearlas, se utiliza software denominado Particionador. Existen 2 tipos:
    - Destructivos: Permiten crear y eliminar particiones (fdisk)
    - No destructivos: Permiten crear, eliminar y MODIFICAR particiones (fips, gparted)

2.- Arranque de un Sistema Operativo

a) ¿Qué es el MBR? ¿Que es el MBC?

5.1. Sector de arranque (MBR)

El *Master Boot Record* es el primer sector del disco físico. Se ubica en el cilindro 0, cabeza 0, sector 1 del disco. Existe un MBR en todos los discos, pero solo es ejecutado si el disco contiene las particiones del sistema. Si existiese más de un disco rígido en la máquina, sólo uno es designado como *Primary Master Disk*. Es de éste disco de donde el sistema realiza el booteo.
El MBR es creado con algún utilitario que permita diseñar la estructura del disco rígido como por ejemplo *fdisk*, *Partition Magic*, *System Comander*, etc.
La tabla de particiones provee información respecto a las particiones realizadas sobre el disco, como por ejemplo donde comienzan y terminan. Si se pierde esta tabla, se perderá la clave para acceder al contenido en el disco rígido, y no existe una tabla de particiones de backup.
La última acción del BIOS (después de realizar los testeos POST) es leer el primer sector de la unidad primaria (ya sea el disco rígido, diskette o CDROM). El programa de booteo lleva el sector leído a memoria, y si los datos son correctos, lo ejecuta, es decir ejecuta el arranque del sistema operativo.
El proceso de inicio de una máquina y carga del sistema operativo se denomina *proceso de bootstrap (carga)* o booteo.
El encendido de la máquina y carga del sistema operativo presenta un interesante dilema. Por definición, una máquina no puede realizar nada hasta que el sistema operativo es cargado. Él es el responsable de correr programas almacenados en el disco. Por ello, la máquina no podría correr un programa sin haber cargado previamente el sistema operativo, y por otra parte los programas del sistema operativo se almacenan en disco. Entonces, ¿cómo se carga el sistema operativo?
En las arquitecturas x86, el *BIOS (Basic I/O System)* es el responsable de cargar el sistema operativo. Para ello, el BIOS analiza el MBR, asumiendo que él puede llevar a cabo el resto de las tareas involucradas en la carga del sistema operativo.
Si se mantiene un sistema operativo instalado, entonces el MBR estándar será suficiente. El MBR buscará la primera partición booteable en el disco, y luego correrá el código en esa partición para proceder con la carga del resto del sistema operativo.
En el caso de tener múltiples sistemas operativos instalados, entonces es posible instalar un MBR diferente, uno que pueda mostrar la lista de los diferentes sistemas operativos, y permitir la selección de cuál bootear.
El proceso de booteo puede verse como una serie de pequeños programas cuya ejecución se va encadenando. Esto se debe a que los PC estándar imponen límites en el tamaño de los programas que pueden correrse en las diferentes instancias del proceso de booteo.
Sus 512 bytes contienen tres bloques con información sobre la arquitectura física y lógica del disco: el Código maestro de carga; la Tabla de particiones, y la Firma
MCB es un pequeño código (primeros 446 bytes del MBR) que permite arrancar el SO. La última acción del BIOS es leer el MCB. Lo lleva a memoria y lo ejecuta.
Dicho código es ya capaz de cargar y ejecutar cualquier otro programa situado en cualquier partición del disco. Que a su vez inicializará directamente el SO, o tal vez una utilidad conocida como gestor de arranque, que permite elegir entre distintas alternativas.Como vemos, es un proceso en cadena: el bootstrap loader es cargado en memoria por un programa situado en la BIOS, y a su vez es capaz de continuar la carga del Sistema Operativo.
El MBR es el Master Boot Record.

- Sector reservado del disco físico (cilindro 0, cabeza 0, sector 1)
- Existe un MBR en todos los discos.
- Si existiese más de un disco rígido en la máquina, sólo uno es designado como Primary Master Disk
- El tamaño del MBR coincide con el tamaño estándar de sector: 512 bytes:
- Los primeros bytes corresponden al Master Boot Code (MBC)
- A partir del byte 446 está la tabla de particiones. Es de 64 bytes
- Al final existen 2 bytes libres
- Es creado con algún utilitario
- EL MBC es un pequeño código que permite arrancar el SO
- La última acción del BIOS es leer el MBC. Lo lleva a memoria y lo ejecuta.
- El proceso de inicio de una máquina y carga del sistema operativo se denomina proceso de bootstrap

- Si se tiene un sistema instalado →MBC típico. Sino→ uno diferente (booteadores)
- El proceso de booteo puede verse como una serie de pequeños programas cuya ejecución se va encadenando
- La finalidad del Bootloader es la de cargar una imagen de Kernel de alguna partición para su ejecución

b) ¿Cuál es la funcionalidad de un "Gestor de Arranque"? ¿Qué tipo existen? ¿Dónde se instalan? Cite gestores de arranque conocidos.

5.3. Gestores de arranque

Antes de iniciar el sistema, se deben indicar una serie de instrucciones especiales que se encuentran en un gestor de arranque, un código existente en el disco duro principal u otro dispositivo de soporte que tiene información sobre cómo arrancar el sistema operativo.
Existen dos posibilidades a la hora de instalar cualquier gestor de arranque. Uno debería ser consciente de cuál utilizar en cada caso:
_ **Instalarlo en el MBR**: En el MBR siempre existe un gestor de arranque básico. Cuando trabajamos con Windows ni siquiera nos damos cuenta de que allí está instalado. Cuando queremos instalar otro sistema operativo en nuestra PC y queremos que ambos puedan convivir, es necesario instalar un gestor de arranque que provea una mayor funcionalidad.
Este tipo de gestor de arranque nos permitirá seleccionar qué sistema operativo queremos arrancar. Entonces instalaremos el gestor de arranque en el MBR si no tenemos ningún otro gestor de arranque instalado, o bien si el que tenemos instalado ya no nos sirve, es decir no provee la funcionalidad de seleccionar qué sistema operativo arrancar. Existe mucha documentación disponible en Internet acerca de cómo realizar esta tarea, la cual no es difícil.
_ **Instalarlo en el primer sector del disco**: Esta opción es correcta si ya estamos usando algún gestor de arranque (obviamente instalado en el MBR) que permita seleccionar qué sistema operativo cargar.
Entre estos programas podemos nombrar 2 entre los más conocidos:
_ LILO (***Li****nux* ***Lo****ader):* Este es el programa más comúnmente utilizado, puesto que es el que más tiempo hace que está en el mercado. Este es provisto con todas las distribuciones de Linux y es de uso Libre. A la hora de instalar Lilo debemos ser conscientes de lo que hacemos, debido a que si sobrescribimos el MBR incorrectamente no podremos volver a arrancar nuestro sistema. Básicamente la interfaz del Lilo es en modo texto, aunque algunas distribuciones de Linux crearon sus propias interfaces semi gráficas.
_ GRUB **(GR**and **U**nified **B**ootloader): Este programa es el más nuevo de los gestores de arranque y el más utilizado en la actualidad. Se incorporó recién en la versión 7.2 del Red Hat Linux y es el que actualmente se provee con las distribuciones modernas de linux.
Poco a poco, se está utilizando más en lugar del lilo. Es similar al Lilo, salvo que presenta una interfaz gráfica más amigable y además permite proteger el arranque con contraseñas.
Al igual que Lilo es gratuito y se lo puede bajar desde la web. Con respecto a los métodos de instalación, sigue las mismas características que Lilo y se lo puede instalar tanto en el primer sector del disco como en el MBR.

Gestores de Arranque

- Permiten la carga del sistema Operativo.
- Se ejecuta luego del código de la BIOS
- Existen 2 modos de instalación
  - En el MBR
  - En el sector de arranque de la partición raíz o activa
- GRUB, LILO, NTLDR, GAG, YaST, etc...
- Se denomina bootstrap (carga)
- En las arquitecturas x86, el BIOS (Basic I/O System) es el responsable de iniciar la carga del SO a través del MBC
  - Está grabado en un chip (ROM, NVRAM)
  - En otras arquitecturas también existe, pero se lo conoce con otro nombre:
    - Power on Reset + IPL en mainframe
    - OBP (OpenBoot PROM): en SPARC
- Carga el programa de booteo (desde el MBR)
- El gestor de arranque lanzado desde el MBC carga el Kernel

    - Prueba e inicializa los dispositivos
    - Luego pasa el control al proceso init
- El proceso de arranque se ve como una serie de pequeños programas de ejecución encadenada

c) Cuales son los pasos se suceden desde que se prende una computadora hasta que el Sistema Operativo es cargado (bootstrap).

El proceso de inicio de una máquina y carga del sistema operativo se denomina *proceso de bootstrap (carga)* o booteo. El encendido de la máquina y carga del sistema operativo presenta un interesante dilema. Por definición, una máquina no puede realizar nada hasta que el sistema operativo es cargado. Él es el responsable de correr programas almacenados en el disco. Por ello, la máquina no podría correr un programa sin haber cargado previamente el sistema operativo, y por otra parte los programas del sistema operativo se almacenan en disco. Entonces, ¿cómo se carga el sistema operativo?

En las arquitecturas x86, el *BIOS (Basic I/O System)* es el responsable de cargar el sistema operativo. Para ello, el BIOS analiza el MBR, asumiendo que él puede llevar a cabo el resto de las tareas involucradas en la carga del sistema operativo.

Si se mantiene un sistema operativo instalado, entonces el MBR estándar será suficiente. El MBR buscará la primera partición booteable en el disco, y luego correrá el código en esa partición para proceder con la carga del resto del sistema operativo.

En el caso de tener múltiples sistemas operativos instalados, entonces es posible instalar un MBR diferente, uno que pueda mostrar la lista de los diferentes sistemas operativos, y permitir la selección de cuál bootear.

El proceso de booteo puede verse como una serie de pequeños programas cuya ejecución se va encadenando. Esto se debe a que los PC estándar imponen límites en el tamaño de los programas que pueden correrse en las diferentes instancias del proceso de booteo.

d) Analice el proceso de arranque en GNU/Linux.

Cuando inicia una computadora el BIOS realiza la auto prueba de encendido (POST) junto con el descubrimiento e inicialización de los dispositivos iníciales, ya que el SO puede requerirlos.

A continuación, se lee el primer sector del disco de inicio - el MBR-, se coloca en una ubicación de memoria fija y se ejecuta. Este sector contiene un pequeño programa que carga un programa independiente llamado *boot* del dispositivo de inicio, que por lo general es un disco IDE o SCSI. El programa *boot* primero se copia a sí mismo en una dirección libre de la parte superior de la memoria, para liberar la parte inferior de la memoria para el sistema operativo.

Una vez que se traslado, el boot lee el directorio raíz del dispositivo de inicio. Para ello, debe comprender el sistema de archivos u el formato de los directorios, lo cual es el caso de algunos cargadores de inicio como GRUB. Después el boot lee el kernel del SO, lo coloca en memoria y salta ahí. En este punto termina su trabajo y el Kernel está en ejecución.

El código de inicio del kernel está escrito en lenguaje ensamblador y es muy dependiente de la máquina. El trabajo típico consiste en establecer la pila del kernel, identificar el tipo de CPU, calcular la cantidad de RAM presente, deshabilitar interrupciones, habilitar el MMU y, por último, llamar al *main* del lenguaje C para que inicie la parte principal del SO.

Después se asignan las estructuras de datos del Kernel. En este punto, el sistema empieza la autoconfiguración. Mediante el uso de archivos de configuración que indican los tipos de dispositivos de E/S que podrían estar presentes, empieza a sondear los dispositivos para ver cuales están presentes en la realidad. Si al sondear un dispositivo, éste responde se agrega a una tabla de dispositivos conectados. Si no responde, se asume que está ausente y se ignora.

Una vez que se ha configurado todo el hardware, lo siguiente es crear en forma manual y con cuidado el proceso 0, establecer su pila y ejecutarlo. El proceso 0 continua la inicialización y realiza actividades como programar el reloj en tiempo real, montar el sistema de archivos raíz, crear *init* (proceso 1) y el demonio de paginación (proceso 2).

Init comprueba sus banderas para ver si debe usar el modo de un solo usuario o de multiusuario. En el primer caso, bifurca un proceso que ejecuta el Shell y espera a que este proceso termine. En el segundo caso, bifurca un proceso que ejecuta la secuencia de

comandos del Shell de inicialización del sistema. Después lee etc/ttys, que lista las terminales y algunas de sus propiedades. Para cada terminal habilitada bifurca una copia de si mismo, que realiza ciertas labores de mantenimiento y después ejecuta un programa llamado gotty. Este programa establece la velocidad de línea y otras propiedades para cada línea y luego escribe "login" en la pantalla de la terminal, donde trata de leer el nombre de usuario del teclado. Cuando alguien se sienta en la terminal y proporciona un nombre de usuario, getty termina al ejecutar /bin/login, el programa de inicio de sesión. Después, login pide una contraseña, la cifra y la compara con la contraseña cifrada que está almacenada en el archivo de contraseñas /etc/shadow. Si es correcta login se reemplaza a sí mismo con el Shell del usuario, que después espera el primer comando. Si es incorrecta, login pide otro nombre de usuario.

e) Cuales son los pasos que se suceden en el proceso de parada (shutdown) de GNU/Linux.
Shutdown va a detener la máquina de una forma ordenada, siguiendo unos pasos definidos. En primer lugar, notifica el hecho a todos los usuarios conectados (mediante wall) y bloquea el proceso de registro (login). Posteriormente invoca a INIT' en un runlevel 0 (para simplemente detener el sistema), 6 (para reinicializarlo) o incluso 1 (monousuario, para realizar tareas administrativas). Entonces INIT ejecuta el script correspondiente (leído de /etc/inittab) que suele encargarse de eliminar todos los procesos de la máquina, notificar el evento en el fichero de log correspondiente, desmontar los sistemas de ficheros que existan, desactivar el área de swap (intercambio) y, según se haya invocado la orden, detener el sistema o reinicializarlo.

f) ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique.
Linux puede convivir con otros sistemas operativos en la misma máquina, es decir, puede correr Windows o DOS juntamente con Linux. Y esto es posible, ya que tenemos las particiones, entonces particionamos el disco y podemos instalar varios sistemas operativos, mediante el gesto de arranque, vamos a elegir, con cuál SO queremos que opere la máquina en cada vez, que prendamos.

3.- Características de GNU/Linux
a) Mencione y explique las características más relevantes de GNU/Linux.
Linux es únicamente su kernel (núcleo), gracias al proyecto GNU existe lo que se llama GNU/Linux que es la unión de los programas realizados bajo el techo de GNU y el kernel Linux.
Las siguientes características están basadas en GNU/Linux.
En líneas generales podemos decir que se dispone de varios tipos de sistemas de archivos para poder acceder a archivos en otras plataformas. Incluye un entorno gráfico X window (Interfaz gráfico estándar para máquinas UNIX), que nada tiene que envidiar a los modernos y caros entornos comerciales. Está orientado al trabajo en red, con todo tipo de facilidades como correo electrónico, suites de oficina, juegos, trabajo de audio y video, etc.
Entre sus muchas características podemos mencionar:
La clave del éxito de Linux ha sido la disponibilidad de los paquetes de software libre bajo los auspicios de la Fundación de Software Libre (FSF). Esta fundación se centra en un software estable, independiente de plataforma, con alta calidad y soportado por la comunidad de usuarios. El proyecto GNU proporciona herramientas para desarrolladores de software y la licencia pública GNU (GPLU: GNU Public Licence) es el sello de aprobación de FSF.

Además, con el código fuente disponible, los distribuidores pueden adaptar las aplicaciones y facilidades para cumplir unos requisitos específicos.
Cuenta con un amplio y robusto soporte para comunicaciones y redes, lo cual hace que sea una opción atractiva tanto para empresas como para usuarios individuales.

Da soporte a una amplia variedad de hardware y se puede correr en una multitud de plataformas: PC's convencionales, computadoras Macintosh, etc., así como costosas estaciones de trabajo.

- Linux y sus Shells: Cada usuario de un sistema Linux tiene su propia interfaz de usuario o Shell. Los usuarios pueden personalizar sus shells adecuándolos a sus propias necesidades específicas. En este sentido, el Shell de un usuario funciona más como un entorno operativo que el usuario puede controlar.
- Linux es Multitarea: La multitarea no consiste en hacer que el procesador realice más de un trabajo al mismo tiempo (un solo procesador no tiene esa capacidad), lo único que realiza es presentar las tareas de forma intercalada para que se ejecuten varias simultáneamente. Por lo tanto en Linux es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.
- Linux es Multiusuario: Para que pueda desarrollar esta labor (de compartir los recursos de un ordenador) es necesario un sistema operativo que permita a varios usuarios acceder al mismo tiempo a través de terminales, y que distribuya los recursos disponibles entre todos. Asimismo, el sistema debería proporcionar la posibilidad de que más de un usuario pudiera trabajar con la misma versión de un mismo programa al mismo tiempo, y actualizar inmediatamente cualquier cambio que se produjese en la base de datos, quedando reflejado para todos. En conclusión, en el sistema multiusuario, varios usuarios pueden acceder a las aplicaciones y recursos del sistema Linux al mismo tiempo. Y, por supuesto, cada uno de ellos puede ejecutar varios programas a la vez (multitarea).
- Linux es Seguro: El concepto de seguridad en redes de ordenadores es siempre difícil de abordar. Un sistema puede ser seguro para un determinado tipo de actividades e inseguro para otras. Si se quiere que el sistema sea seguro, se debe administrar de tal forma que se tengan controlados a los usuarios en todo momento. Para la ardua tarea de seguridad surgen nuevas herramientas constantemente, tanto para detectar intrusos como para encontrar fallos en el sistema y evitar así ataques desde el exterior.
- Linux y las Redes de Ordenadores: Cuando se trabaja con Linux se está ante un sistema operativo orientado al trabajo de redes de ordenadores.
    - Linux dispone de varios protocolos como PPP, SLIP, TCP/IP, PLIP, etc., para la transferencia de archivos entre plataforma. Tiene a su disposición multitud de aplicaciones de libre distribución que permiten navegar a través de Internet y enviar y recibir correo electrónico, hacer una video-conferencia, transferir archivos, etc. Posee gran variedad de comandos para comunicación interna entre usuarios que se encuentren ubicados en plataformas distintas (gracias a utilidades como telnet).
    - Independencia de dispositivos Linux admite cualquier tipo de dispositivo (módems, impresoras) gracias a que, una vez instalado uno nuevo, se añade al Kernel el controlador o driver necesario con el dispositivo, haciendo que el Kernel y el driver se fusionen. Lo importante de esto es que el controlador funciona como un modulo completamente aislado del núcleo de Linux, dando así una mayor seguridad y estabilidad al sistema.

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

Aspecto	GNU/Linux	Windows
Filosofía	El sistema es libre, cualquiera lo puede usar, modificar y distribuir.	Pertenece a Microsoft, única compañía que lo puede modificar.
Precio	Gratis, tantas licencias como se desee.	Según las versiones, cientos de euros, cada licencia.
Desarrollo	Miles de voluntarios en todo el mundo, cualquiera puede participar, pertenece a la "comunidad".	Lo desarrolla Microsoft, que vende algunos datos técnicos relevantes y oculta otros.
Código fuente	Abierto a todos.	Secreto empresarial.
Estabilidad	Muy estable, es difícil que se quede colgado. Los servidores que lo usan pueden funcionar durante meses sin parar.	Poco estable, es común verse obligado a reiniciar el sistema. Los servidores no admiten más allá de un par de semanas sin reiniciar.
Seguridad	Extremadamente seguro, tiene varios sistemas de protección. No existen virus para Linux.	Muy poco seguro, existen miles de virus que atacan sistemas Windows.
Facilidad de uso	En muchas tareas, poca. Día a día mejora este aspecto.	Cuando funciona, es muy sencillo de manejar.
Controladores de hardware	Desarrollados por voluntarios; algunos dispositivos no funcionan en absoluto porque sus fabricantes ocultan los detalles técnicos.	Los fabricantes de dispositivos siempre los venden con controladores para Windows, todos deben funcionar en pocos momentos.
Difusión	Poco extendido en hogares y oficinas, muy extendido en servidores.	Copa todo el mercado, salvo el de servidores.
Disponibilidad de programas.	Existen programas para casi todas las facetas, pero no hay tanta variedad como los programas para Windows.	Miles y miles de programas de todo tipo que se instalan con facilidad.
Precio de los programas.	Existen programas de pago, pero lo más habitual es que sean libres.	La mayor parte de los programas son de pago.
Comunicación con otros sistemas operativos.	Lee y escribe en sistemas de archivos de Windows, Macintosh, etc. Por red, se comunica con cualquier otro sistema.	Sólo lee y escribe sus propios sistemas de archivos, y presenta incompatibilidades entre algunas de sus versiones.

c) ¿Qué es GNU? Indique una breve historia sobre la evolución del proyecto GNU.

El sistema operativo GNU es un sistema completo de software libre, compatible hacía el futuro con Unix. El término GNU proviene de «GNU No es Unix». Richard Stallman escribió el anuncio inicial del Proyecto GNU en setiembre de 1983. Una versión extendida, denominada el Manifesto de GNU se publicó en setiembre de 1985. Se ha traducido a diversos idiomas.

El nombre «GNU» se eligió porqué satisfacía unos cuantos requisitos. En primer lugar, era un acrónimo recursivo para «GNU No es Unix».

La palabra "libre" en "software libre" se refiere a libertad, no a precio. Puede o no pagar un precio por obtener software de GNU. De cualquier manera, una vez que obtiene el software, tiene cuatro libertades específicas para usarlo. La libertad de ejecutar el programa como desée, la libertad de copiar el programa y darlo a sus amigos o compañeros de trabajo. La libertad de cambiar el programa como desee, teniendo acceso completo al código fuente. La libertad de distribuir una versión mejorada ayudando así a construir la comunidad (si redistribuye software de GNU, puede cobrar una tarifa por el acto físico de efectuar la copia, o bien puede regalar copias.).

El proyecto para desarrollar el sistema GNU se denomina «Proyecto GNU». El Proyecto GNU se concibió en 1983 como una forma de devolver el espíritu cooperativo que prevalecía en la comunidad computacional en sus primeros días; hacer la cooperación posible al eliminar los obstáculos impuestos por los dueños de software privativo.

En 1971, cuando Richard Stallman comenzó su carrera en el MIT, trabajó en un grupo que usaba software libre exclusivamente. Incluso compañías informáticas frecuentemente distribuían software libre. Los programadores eran libres de cooperar unos con otros, y frecuentemente lo hacían.
En los 80, casi todo el software era privativo, lo cual significa que tenía dueños que prohibían e impedían la cooperación entre usuarios. Esto hizo necesario el Proyecto GNU.
Cada usuario de ordenadores necesita un sistema operativo; si no existe un sistema operativo libre, entonces no puedes ni siquiera comenzar a usar una computadora sin recurrir a un software privativo. Así que el primer elemento en la agenda del software libre es un sistema operativo libre.
Decidieron hacer el sistema operativo compatible con Unix porque el diseño en general ya estaba probado y era portable, y porque la compatibilidad hacía fácil para los usuarios de Unix cambiar de Unix a GNU.
Un sistema operativo incluye un núcleo, compiladores, editores, editores de texto, software de correo, interficies gráficas, bibliotecas, juegos y muchas otras cosas. Por todo esto, escribir un sistema operativo completo es un trabajo bastante grande. La Free Software Foundation se fundó en octubre de 1985 con el objetivo inicial de recaudar fondos para ayudar a programar GNU.
En 1990 ya habían encontrado o escrito los componentes principales, excepto uno, el núcleo. Entonces, Linux, un núcleo similar a Unix, fue programado por Linus Torvalds en 1991 y lo liberó como software libre el 1992. La combinación de Linux con el prácticamente completo sistema GNU formó un sistema operativo completo: el sistema GNU/Linux. Se estima que existen decenas de millones de personas que en la actualidad usan sistemas GNU/Linux, habitualmente mediante distribuciones.

d) Explique que es la multitarea preventiva, e indique si GNU/Linux hace uso de ella.
Multitarea preventiva es una característica de los sistemas operativos. Consiste en que en una multitarea (cuando varias aplicaciones se ejecutan al mismo tiempo) es el procesador el que asigna tiempos de CPU a las tareas que se están ejecutando. En sistemas antiguos, son los programas los que toman el control del procesador. Tal situación conlleva a que si la aplicación se cuelga el procesador queda inutilizado provocando un error del sistema. Linux hace uso de ella, OBVIAMENTE. http://es.wikipedia.org/wiki/Multitarea_preventiva

e) ¿Qué es POSIX?
POSIX es el acrónimo de **P**ortable **O**perating **S**ystem **I**nterface; la **X** viene de UNIX como seña de identidad de la API. Una traducción aproximada del acrónimo podría ser "Interfaz para Sistemas Operativos migrables basados en UNIX".
Estos son una familia de estándares de llamadas al sistema operativo definidos por el IEEE y especificados formalmente en el IEEE 1003. Persiguen generalizar las interfaces de los sistemas operativos para que una misma aplicación pueda ejecutarse en distintas plataformas. Estos estándares surgieron de un proyecto de normalización de las API y describen un conjunto de interfaces de aplicación adaptables a una gran variedad de implementaciones de sistemas operativos.
Especifica las interfaces de usuario y software al sistema operativo en 15 documentos diferentes. La línea de comandos estándar y las interfaces de *scripting* se basaron en Korn Shell. Otros programas a nivel de usuario (*user-level*), servicios y utilidades incluyen AWK, echo, ed y cientos de otras. Los servicios a nivel de programa requeridos incluyen definición de estándares básicos de I/O, (file, terminal, y servicios de red). También especifican una API para las bibliotecas de threading, que es muy utilizada en una gran variedad de sistemas operativos.

http://gbtcr.chileforge.cl/info_web/node42.html

Multitarea

Linux desde su concepción fue diseñado como un sistema operativo multitarea, lo que le permite ejecutar varios programas a la vez, de forma que no tiene que esperar a que termine

uno para empezar otro. La multitarea está controlada por el Sistema Operativo (S.O.) y no por las aplicaciones, por lo que es muy difícil que el fallo de un programa "cuelgue" el sistema por una mala utilización de los recursos del equipo.

32 bits reales

Linux permite aprovechar toda la potencia del procesador, corre a 32 bits reales en un procesador Intel o AMD, y a 64 bits en los nuevos procesadores que están llegando al mercado. Esto le confiere al sistema rapidez, eficacia, seguridad y fiabilidad.

Multiusuario

Linux es un sistema operativo capaz de responder, simultáneamente, a las solicitudes de varios usuarios que empleen el mismo ordenador, incluso con necesidades distintas. Además proporciona los elementos necesarios para garantizar la seguridad y privacidad de los datos entre los diferentes usuarios.

POSIX

POSIX es un estándar de la industria que asegura una calidad mínima en ciertas partes del S.O. y asegura la compatibilidad a nivel de código. De esta forma los programas POSIX que funcionan en un UNIX no tienen ningún problema para compilarse y ejecutarse en Linux.

Estabilidad

Linux es robusto, por lo que si un programa falla no interrumpirá el trabajo de los demás. Entraremos al sistema, desbloquearemos el programa y podremos seguir utilizando el sistema sin ningún problema. Esta característica permite que el sistema funcione durante periodos muy largos de tiempo sin necesidad de parar y volver a arrancar.

Es libre

Como disponemos del código fuente, podemos hacer cualquier modificación sin tener que esperar a que alguien nos envíe un "Service Pack" para solucionarlo. En el caso de que no sepamos arreglar el fallo podremos contratar a cualquier empresa para que lo arregle, aún cuando la empresa que nos vendió el programa haya cerrado o no le interese resolver nuestro problema, ya que se conoce el código fuente.

Soporte

Si compras una de las distribuciones de Linux dispondrás de soporte de las empresas que los distribuyen (Red Hat, Mandriva, SUSE, Ubuntu, etc.) o de otras muchas que se han especializado en Linux (desde gigantes como IBM o HP hasta empresas españolas como Activa Sistemas, Esware o Andago). Si aun así no lo ves claro aquí tienes una iniciativa que permite localizar empresas que dan soporte a aplicaciones de software libre **http://www.findopensourcesupport.com**

Adaptación

Linux es un S.O. que evoluciona rápidamente adaptándose a las novedades del mercado y solucionando rápidamente los problemas que puedan surgir, además se puede personalizar.

Sistema de archivos

Linux puede operar con una gran variedad de sistemas de archivos, pudiéndolos leer y operar con ellos. Por ejemplo: FAT, VFAT, OS2/FS, ISO9660, ReiserFS, etc.

Multiplataforma

Linux es soportado por los sistemas informáticos independientemente del microprocesador que lleven instalado (386, 486, Pentium, Pentium Pro, Pentium II, Pentium III, Pentium 4, AMD 64, Amiga, Atari, Alpha, PowerPC, SPARC, RISC, etc.).

Red

Linux fue desarrollado desde sus comienzos para trabajar en red. Su protocolo principal es TCP/IP, aunque soporta una gran variedad de protocolos como SLIP/PPP, PLIP, NFS, Telnet, TNP, SMTP, IPX, AppleTalk, etc. Además es capaz de mediar entre todo tipo de redes, permitiendo trabajar en red con equipos que utilicen sistemas operativos como Windows 98 o XP sin ningún problema.

Entorno Gráfico

Linux puede trabajar con o sin entorno gráfico. Por ejemplo para funcionar de manera óptima en equipos con poca memoria o en servidores donde el entorno gráfico consume recursos innecesariamente. Si por el contrario queremos usar un entorno de ventanas, existen un sinfín de gestores (ICEwin y otros) y de entornos de escritorio (KDE y GNOME son los más populares) que permiten al usuario doméstico trabajar de una forma intuitiva.
http://www.adrformacion.com/cursos/linux/leccion1/tutorial3.html

Explicación→http://www.scribd.com/doc/3897438/WINDOWS-VS-LINUX-VS-MAC

4.- Distribuciones de GNU/Linux
a) Investigue acerca de las principias distribuciones de GNU/Linux.
b) ¿En qué se diferencia una distribución de otra?

→http://www.adrformacion.com/cursos/linux/leccion1/tutorial3.html

Una distribución es un sistema operativo GNU/Linux unido a una serie de aplicaciones de configuración y de usuario "empaquetadas" juntas. Todas tienen en común el núcleo del sistema. Las diferencias entre unas y otras son las herramientas de configuración que utilizan y las diferentes aplicaciones que se incluyen junto al sistema operativo.
Si creamos una distribución para utilizar en ordenadores pequeños y en los que no usaremos monitor, no incluiremos entornos gráficos, ni herramientas para ver y editar vídeo. Sin embargo en una aplicación para usuarios domésticos será importante que sea fácil de configurar y que incluya programas de entretenimiento.

Knoppix

Es una distribución en un CD-live basada en Debian. Detecta automáticamente todo tipo de hardware y aporta el último escritorio de KDE y la suite OpenOffice.org. Muy útil para demostraciones y usuarios novatos en el sistema. Se puede encontrar más información en http://www.knoppix-es.org/ .

Red Hat

Es una de las empresas más importantes en el panorama Linux. Actualmente ofrece dos tipos de distribuciones, una paga, que incluye software comercial; y otra gratuita, que sustituye este software comercial por otro con licencia libre, que se llama Fedora Core.

Tiene el mérito de haber inventado el sistema de paquetes RPM (**R**ed Hat **P**ackage **M**anager) que facilita la instalación de nuevos programas.

¡Nota! Paquete es un fichero donde se ha incluido un programa, junto con la información necesaria para su correcto funcionamiento, avisándote si necesitas instalar algún paquete adicional, y permite desinstalar el programa de forma sencilla. Esto ahorra tener que compilar el código fuente del programa como se hacía antes.

La instalación de Red Hat es intuitiva. Permite elegir entre un cómodo interfaz gráfico u otro en formato texto, lo que facilita enormemente su instalación. Es una de las distribuciones que más se instala a nivel de producción en el mundo empresarial, sobre todo como en los proveedores de servicios Internet.

Podemos encontrar más información en http://fedora.redhat.com/ o http://www.redhat.es/fedora/.

Debian

Una de las primeras distribuciones de GNU/ Linux que aparecieron y aún siguen existiendo y evolucionado. El sistema de paquetes nos permite diferenciar claramente el software libre del que no lo es, permitiéndonos disponer de todo el sistema solamente con programas de licencia Free Software. Utiliza el sistema de paquetes .deb, aunque incorpora la herramienta Alien, que permite utilizar paquetes de cualquier otra distribución. Está desarrollada por un grupo de colaboradores distribuidos por todo el mundo y no cuenta con el respaldo de ninguna empresa.

Esta distribución es muy conocida por su fiabilidad, estabilidad. Podemos encontrar más información en http://www.gnu.org, http://www.debian.org y http://www.es.debian.org.

SUSE

Es una distribución basada en RPM. Es la segunda distribución implantada en el mundo empresarial. Incluye una gran cantidad de software, además de un sistema de administración y configuración muy cuidado por lo que es ideal para aquellos que vienen del mundo Windows. Recientemente ha sido adquirida por Novell.

Puedes encontrar más información en http://www.suse.com y http://www.novell.com/es-es/linux/suse/ .

Slackware

Una de las primeras distribuciones que aparecieron. Fue creada por Patrick Volkerding y tuvo un gran éxito en sus primeros años de existencia. No cumple la organización estándar de fichero de Linux, y la configuración del sistema se realiza a mano. Utiliza un sistema de paquetes TGZ, muy rudimentario. Posee un sistema de instalación semi-gráfico. Más información en http://www.eslack.org/ .

Gentoo

Es una distribución muy nueva. Su instalación se realiza desde el código fuente, recompilándolo, por lo que es lenta, pero asegura la máxima optimización. Requiere unos altos conocimientos de Linux. Puedes encontrar más información en http://www.gentoo-es.org .

Ubuntu

También es una distribución muy nueva, está impulsada por la empresa Canonical, propiedad del millonario Sudafricano Mark Shuttelworth (famoso entre otras cosas por ser el primer turista espacial de la historia). Su lema es "Linux para seres humanos" y a pesar de su juventud ha alcanzado una gran popularidad debido a su gran soporte para hardware. Además desde su página http:// www.ubuntu.com es posible solicitar el envío de los CDs de instalación de manera totalmente gratuita.

Mandriva (antes Mandrake)

Es una distribución creada a raíz de la fusión de dos empresas Mandrake (Francia) y Conectiva (Brasil). Destaca por su facilidad de uso, su sencillo proceso de instalación y por sus asistentes que permiten realizar la mayoría de las tareas de configuración de una forma intuitiva.

Puedes encontrar más información en http://www.mandriva.com/es o en http://www.mandrakefacil.org

5.- Estructura de GNU/Linux.

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

Los tres elementos fundamentales son:

- El Kernel (núcleo) es la parte fundamental del SO. Es, a grandes rasgos, el encargado de que el SW y el HW de una computadora puedan trabajar juntos. Es responsable de mantener todas las abstracciones importantes del SO, incluyendo elementos tales como la memoria virtual y los procesos.
- El Shell (intérprete de comandos) es el programa que recibe lo que se escribe en la Terminal y lo convierte en instrucciones para el SO. Un intérprete de comandos es un programa que lee las entradas del usuario y las traduce a instrucciones que el sistema es capaz de entender y utilizar.
- El Filesystem se traduce como "sistema de archivos" y es la forma en que dentro de un SO se organizan y administran los archivos.

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

De la misma manera que el Unix, el Linux se puede dividir generalmente en cuatro componentes principales: el núcleo(kernel), el shell, el sistema de archivos y las utilidades. El núcleo es el programa medular que ejecuta programas y gestiona dispositivos de hardware tales como los discos y lasimpresoras. El shell proporciona una interfaz para el usuario. Recibe órdenes del usuario y las envía al núcleo para ser ejecutadas. El sistema de archivos, organiza la forma en que se almacenan los archivos en dispositivos de almacenamiento tales como los discos. Los archivos están organizados en directorios. Cada directorio puede contener un número cualquiera de subdirectorios, cada uno de los cuales puede a su vez, contener otros archivos.
El núcleo, el shell y el sistema de archivos forman en conjunto la estructura básica del sistema operativo. Con estos tres elementos puede ejecutar programas, gestionar archivos e interactuar con el sistema. Además, Linux cuenta con unos programas de software llamados utilidades que han pasado a ser considerados como características estándar del sistema. Las utilidades son programas especializados, tales como editores, compiladores y programas de comunicaciones, que realizan operaciones de computación estándar. Incluso uno mismo puede crear sus propias utilidades.

6.- Kernel

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

El kernel es el núcleo del SO, en sí, ES el sistema operativo El kernel o núcleo es la parte esencial de un sistema operativo. El kernel proporciona todos los servicios básicos requeridos por otras partes del sistema operativo y aplicaciones, y en complemento, se encarga de la administración de la memoria, los procesos y los discos. El kernel es independiente de la distribución GNU/Linux utilizada, de tal forma que el mismo kernel debería servir en cualquier caso. La actualización del kernel se divide en 2 partes: compilación e instalación del nuevo kernel.

El primer Kernel de GNU/Linux presenteado al público fue la versión 0.01 en mayo de 1991. No disponía de capacidad de interconexión por red, sólo se ejecutaba sobre procesadores Intel compatibles con el 80386 y HW PC. La siguiente versión importante, Linux 1.0, fue lanzada en marzo de 1994. Esta versión culminó tres años de rápido desarrollo del kernel de Linux. Quizá la novedad más importante era la conexión por red: la versión 1.0 incluía soporte para los protocolos de red estándar TCP/IP de UNIX. El kernel 1.0 también incluía un nuevo sistema de archivos muy mejorado, sin las limitaciones del sistema de archivos Minix original, y soportaba diversos controladores SCSI para el acceso a discos de alta velocidad. Luego de dos años, de generar nuevas y nuevas versiones, que agregaban varias funcionalidades a la versión que las antecedia, se terminó por lanzar la versión 2.0. Con este lanzamiento se incrementó el número de versión principal para reflejar dos nuevas capacidades: el soporte para múltiples arquitecturas, y el soporte para arquitecturas multiprocesador. Y aún desde esas versiones, hasta las actuales se sigue trabajando constantemente para agregarle mejores al kernel.

b) ¿Cuales son sus funciones principales?

Sus funciones principales son:

- Ejecuta programas y gestiona dispositivos del hardware
- Es el encargado de que el software y el hardware puedan trabajar juntos
- Las funciones más importantes son las de administrar la memoria, comunicación y concurrencia, administración de procesos y gestión del HW.

c) ¿Cuál es la versión actual? ¿Que versiones existen? ¿Cómo se las diferencian? ¿Qué cambio en el versionado se impuso a partir de la versión 2.6?
La versión actual de kernel es la 3.3.4.
Antiguamente había dos tipos de versiones del núcleo:

- Producción (estable): Era la versión estable del momento y la que se debía de utilizar, ya que, esta versión era el resultado final de las versiones que estaban en desarrollo.
- Desarrollo: Era la versión que estaba en desarrollo y la que los programadores utilizaban para corregir bugs. Esta versión era muy inestable.

Se interpretan con X.Y.Z

- X: Indica serie principal. Cambia cuando su funcionalidad sufre un cambio muy importante
- Y: Indica producción o desarrollo (los números impares indicaban desarrollo, los pares producción)
- Z: Indica si el kernel tiene revisiones dentro de la versión (sólo se modificaban fallos de programación).

A partir de la versión 2.6 se introdujo un cambio en el versionado: A.B.C.[D]

- A: Denota Versión. Cambia con menor Frecuencia. En 1994 (versión 1.0) y en 1996 (versión 2.0)
- B: Denota Mayor revisión. Da igual verlos de forma par o impar, ya no tienen significado.
- C: Denota Menor revisión. Solo cambia cuando hay nuevos drivers o características
- D: Cambia cuando se corrige un grave error sin agregar nueva funcionalidad

d) Es posible tener más de un Kernel de GNU/Linux.
Es posible tener más de un Kernel gracias a los cargadores como GRUB o LILO que permiten elegirlos al iniciar el sistema. Podemos tener más de un Kernel en una misma partición con GNU/Linux, los distintos Kernel's formarían parte del núcleo, y GNU es lo que abraza a ese núcleo que sería uno solo, para los distintos Kernel que yo tenga.

e) ¿Donde se encuentran ubicados dentro del File System?
Las librerías esenciales y módulos del Kernel se encuentran en /boot/:

f) ¿El Kernel de GNU/Linux es monolítico? Justifique.
Hoy por hoy, Linux es un núcleo monolítico híbrido. La mayoría de los núcleos Linux son monolíticos. Un núcleo monolítico es aquel que incluye prácticamente toda la funcionalidad del sistema operativo en un gran bloque de código que ejecuta como un único proceso con un único espacio de direccionamiento.
Todos los componentes fundamentales del núcleo tienen acceso a todas las estructuras internas de datos y rutinas.
Si los cambios se hacen en cualquier porción de un sistema operativo monolítico, todos los módulos y rutinas deben volverse a enlazar y reinstalar, y el sistema debe de ser reiniciado para que los cambios tengan efecto. Como resultado, cualquier modificación, como por ejemplo añadir un nuevo controlador de dispositivo o función del sistema de archivos, es difícil. Aunque Linux no utiliza una técnica de micro núcleos, logran muchas de las ventajas potenciales de esta técnica por medio de su arquitectura modular particular. Linux está estructurado como una colección de módulos, alguno de los guales pueden cargarse y descargarse automáticamente bajo demanda. Por tanto, aunque Linux se puede tomar como monolítico, su estructura modular elimina algunas de las dificultades para desarrollar y evolucionar el núcleo.

7.- Intérprete de comandos (Shell)
a) ¿Qué es?
El Shell (intérprete de comandos) es el programa que recibe lo que se escribe en la terminal y lo convierte en instrucciones para el sistema operativo. Un intérprete de comandos es un

programa que lee las entradas del usuario y las traduce a instrucciones que el sistema es capaz de entender y utilizar.

b) ¿Cuales son sus funciones?
El shell es un programa que actúa como interfaz, para comunicar al usuario con el sistema operativo mediante una ventana que espera comandos textuales ingresados por el usuario en el teclado, los interpreta y los entrega al SO para su ejecución. La respuesta del SO es mostrada al usuario en la misma ventana. A continuación, la shell queda esperando más instrucciones. Se interactúa con la información de la manera más simple posible, sin gráficas, solo el texto.

c) Mencione al menos 3 tipos de intérpretes de comandos que posee GNU/Linux y compárelos entre ellos.
Es posible que un sistema operativo tenga varios intérpretes de comandos; dentro de GNU/Linux y Unix, existen tres grandes familias de Shells, estas son: Korn-Shell (ksh), Bourne-Shell (sh) y C-Shell (csh). Estas se diferencian entre sí básicamente en la sintaxis de sus comandos y en la interacción con el usuario. El mas usado hoy en día es bash (-su nombre es un acrónimo de bourne-again shell, haciendo un juego de palabras sobre el Bourne-shell-).

d) ¿Donde se ubican (ruta) los comandos propios y externos al Shell?
Los comandos propios y externos al Shell se ubican en /bin/

e) ¿Por qué considera que el Shell no es parte del Kernel de GNU/Linux?
La shell no forma parte del kernel básico del SO; sino que la misma "dialoga" con el kernel. La shell es iniciada por un proceso denominado "login", y dado que cada usuario tiene asignado una shell por defecto, la misma se inicia cada vez que un usuario comienza a trabajar en su estación de trabajo (es decir se "loguea" en una terminal). Dentro del contenido del archivo /etc/passwd, se puede ver cual es la shell que cada usuario tiene asignada por defecto.
El funcionamiento del shell consiste en que, en su forma mas básica, se muestra un prompt (-conjunto de caracteres que se muestran en una línea de comandos para indicarnos que está a la espera de ordenes. En el Bourne Shell y sus derivados, el prompt suele ser el carácter $ para los usuarios y # para el administrador-), en donde el usuario teclea una orden en el teclado y finaliza la orden (-normalmente con la tecla Intro/Enter-), y la computadora ejecuta la orden, proporcionando una salida de texto.

8.- Sistema de Archivos (File System)
a) ¿Qué es?
Filesystem se traduce como "sistema de archivos". Es la forma en que dentro de un sistema de cómputo se organizan, se administran, los archivos. Esa administración comprende:
• **Métodos de acceso:** cómo se acceden los datos contenidos en el archivo.
• **Manejo de archivos:** cómo actúan los mecanismos para almacenar, referenciar, compartir y proteger los archivos.
• **Manejo de la memoria secundaria:** Cómo se administra el espacio para los archivos en memoria secundaria.
• **Mecanismos de integridad:** con qué métodos se garantiza la incorruptibilidad del archivo.
La mayoría de los sistemas operativos poseen su propio sistema de archivos. Los sistemas de archivos estructuran la información que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos (por ejemplo Midnight Commander, Nautilus, Konqueror, etc.).
Los filesystem más comunes utilizan dispositivos de almacenamiento de datos que permiten el acceso a los datos como una cadena de bloques de un mismo tamaño, a veces llamados sectores (usualmente de 512 bytes de longitud). El software del sistema de archivos es responsable de la organización de estos sectores en archivos y directorios y mantiene un registro de qué sectores pertenecen a qué archivos y cuáles no han sido utilizados.
Generalmente un sistema de archivos (filesystem) tiene directorios que asocian nombres de archivos con archivos, usualmente conectando el nombre de archivo a un índice en una tabla de asignación de archivos de algún tipo, como FAT en sistemas de archivos MS-DOS o los

inodos de los sistemas Unix. La estructura de directorios puede ser plana o jerárquica (ramificada o "en árbol"). En algunos sistemas de archivos los nombres de archivos son estructurados, con sintaxis especiales para extensiones de archivos y números de versión. En otros, los nombres de archivos son simplemente cadenas de texto y los metadatos de cada archivo son alojados separadamente.
En sistemas de archivos jerárquicos, en lo usual, se declara la ubicación precisa de un archivo con una cadena de texto llamada "ruta" (path). 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. Los sistemas de archivos tradicionales proveen métodos para crear, mover y eliminar tanto archivos como directorios, pero carecen de métodos para crear, por ejemplo, enlaces adicionales a un directorio o archivo (enlace duro en Unix - los enlaces duros son una referencia, ó indicador a los datos físicos sobre un sistema de archivos-) ó renombrar enlaces padres (el ".." en Unix).

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

• **ext2** (second extended filesystem o "segundo sistema de archivos extendido") es un sistema de archivos para el sistema operativo GNU/Linux. La principal desventaja de ext2 es que no implementa el registro por diario o bitácora (Journaling) que sí implementa su sucesor ext3, el cual es totalmente compatible. ext2 fue el sistema de archivos por defecto de las distribuciones Red Hat Linux, Fedora Core y Debian hasta ser reemplazado recientemente por su sucesor ext3.

• **ext3** (third extended filesystem o "tercer sistema de archivos extendido") es un sistema de archivos con registro por diario. Es el sistema de archivo más usado en distribuciones GNU/Linux. La principal diferencia con ext2 es el registro por diario. Un sistema de archivos ext3 puede ser montado y usado como un sistema de archivos ext2. Otra diferencia importante es que ext3 utiliza un árbol binario balanceado (árbol AVL) e incorpora el asignador de bloques de disco Orlov.

• **ReiserFS** es un sistema de archivos de propósito general. Actualmente es soportado por Linux y existen planes de futuro para incluirlo en otros sistemas operativos. A partir de la versión 2.4.1 del núcleo de Linux, ReiserFS se convirtió en el primer sistema de ficheros con journal en ser incluido en el núcleo estándar. También es el sistema de archivos por defecto en varias distribuciones, como SuSE (excepto en openSuSE 10.2 que su formato por defecto es ext3), Xandros, Linspire y Knoppix

• **XFS** es un sistema de archivos de 64 bits con journaling de alto rendimiento. XFS se incorporó a Linux a partir de la versión 2.4.25. Los programas de instalación de las distribuciones de SuSE, Gentoo, Mandriva, Slackware, Fedora Core, Ubuntu y Debian ofrecen XFS como un sistema de archivos más. En FreeBSD el soporte para solo-lectura de XFS se añadió a partir de Diciembre de 2005 y en Junio de 2006 un soporte experimental de escritura fue incorporado a FreeBSD-7.0- CURRENT.

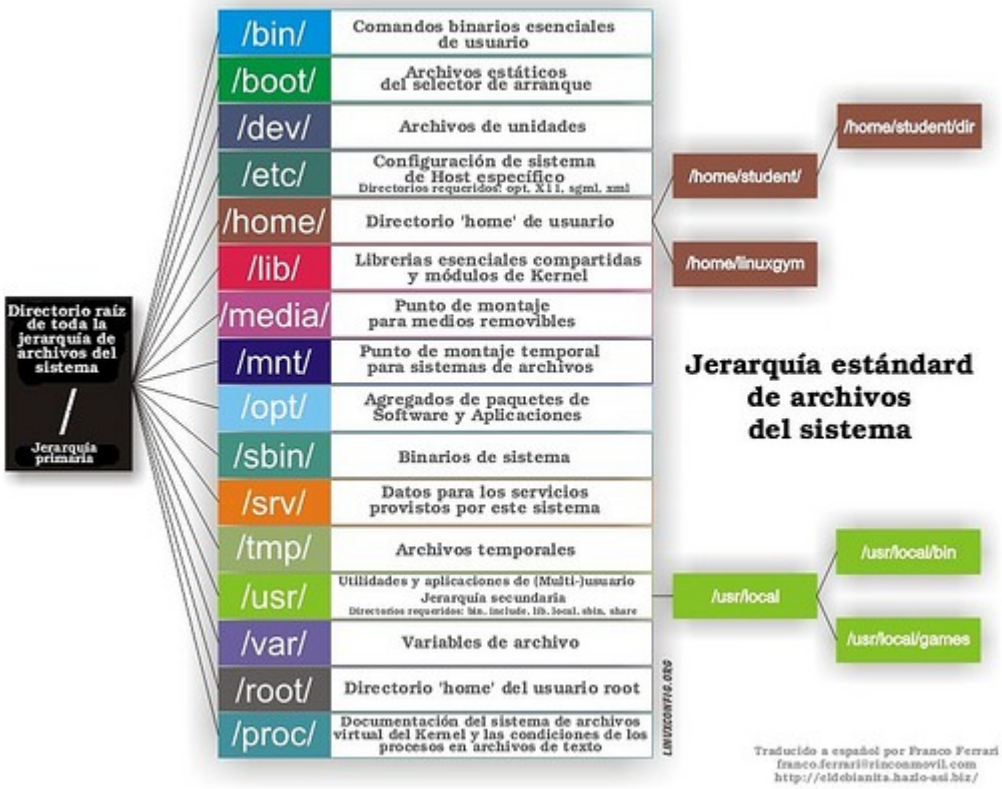
c) ¿Es posible visualizar particiones del tipo FAT y NTFS en GNU/Linux?

Si.

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

FHS define los directorios principales y sus contenidos en el sistema operativo 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. En 1995 se amplió el ámbito del estándar a cualquier Unix que se adhiriese voluntariamente. Todos los archivos y directorios aparecen bajo el directorio raíz /, aunque se encuentre en distintos dispositivos físicos.

En GNU/Linux no existen unidades, todo se “monta” como carpetas que empiezan en el directorio principal(también llamado raíz), además, aunque puedes crear las carpetas que quieras, existe una estructura básica:
/ **[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.
/lib Librerías esenciales compartidas y módulos del Kernel
/media Punto de montaje para dispositivos removibles.
/mnt Punto de montaje para montar filesystems temporarios
/home Aquí cada usuario posee un directorio donde se guarda toda su configuración, los programas instalados y documentos personales.
/opt paquetes de software de aplicación agregados.
/tmp archivos temporarios
/proc usado en lugar del /dev/kmen – soportar los procesos.
/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 bases de datos, directorio raíz de servidores HTTP y FTP, colas de correo, archivos temporales, etc.



9.- Comandos (Investigue su funcionamiento y parámetros más importantes):

Comando: cd
Descripción: Comando para cambiar de directorios
Uso: cd [directorio]
Ejemplos:

- cd Nombre (entra al directorio “nombre”

- cd .. (sale del directorio)

Comando: ls
Descripción: Lista los archivos y directorios del directorio donde estemos. También se puede listar otros directorios distintos de donde estamos.
Uso: ls [opciones] [directorio]
Ejemplos:
- ls (lista el contenido del directorio en donde nos encontremos.
- ls -l (muestra la vista en forma de lista)
- ls -h (cambia los valores en MB de los archivos)
- ls -a (muestra archivos ocultos)
- ls -R (muestra de forma recursiva el contenido del directorio

Comando: pwd
Descripción: Nos muestra en que directorio estamos parados

Comando: df
Descripción: Imprime la utilización del espacio del disco en sistema de ficheros.

Comando: shutdown
Descripción: Comando para apagar o reiniciar nuestro sistema
Uso: shutdown [opciones] [cuando] [mensaje]

Ejemplos:
- shutdown -h now (Apaga el equipo ahora)
- shutdown -r now (Reinicia el equipo ahora)
- shutdown -h 16:00 (Apagará el equipo a las 16 horas)
- shutdown -h +5 (Apagará el equipo 5 minutos después de ejecutado el comando)

Comando: reboot
Descripción: Comando para reiniciar la PC.

Comando: halt
Descripción: Comando que para la ejecución.

Comando: find
Descripción: Comando para buscar archivos dentro de nuestro filesystem
Uso: find [opciones] [path] [expresion] [acciones]
Ejemplos:
- find . -name archivo (busca en el directorio donde esto el archivo con nombre “archivo”)
- find / -name archivo -print (busca en todo el filesystem el archivo con nombre “archivo” mostrandome la ruta completa)

Comando: locate
Descripción: Busca archivos en una base de datos indexada.

Comando: uname
Descripción: Nos muestra información de nuestro sistema y de nuestra máquna.

Comando: dmesg
Descripción: Nos permite extraer información de nuestro kernel directamente desde el buffer de mensajes. El comando muestra información que se encuentra en el archivo **/var/log/dmesg**.

Ahora pasemos a los comandos, Para ver lo último accedido:
dmesg | tail

Creamos un archivo con el registro completo de dmesg:
dmesg > miregistro

Filtrando por palabras:
dmesg | grep -i busqueda

Para borrar el registro para empezar de nuevo:
dmesg –c

Comando: who
Descripción: Nos muestra una lista de los usuarios actualmente logueados en nuestro sistema.

Comando: lspci
Descripción: Lista todos los dispositivos PCI.

Comando: at
Descripción: Ejecuta comandos en un tiempo específico.

Comando: touch
Descripción: Cambia la fecha de un archivo o lo crea.

Comando: netstat
Descripción: Imprime información sobre el estado de la red.

Comando: tail
Descripción: Nos permite ver las ultimas lineas (por default son 10) de algun archivo.

Comando: head
Descripción: Imprime las 10 primeras lineas del archivo (10 por default).

Comando: mount
Descripción: Monta un dispositivo.

Comando: umount
Descripción: Desmonta un dispositivo.

Comando: losetup
Descripción: Activa y controla los dispositivos de bucle.

Comando: write
Descripción: Envía un msj a otro usuario.

Comando: mkfs
Descripción: Construye un sistema de ficheros Linux en un dispositivo, normalmente una partición del disco duro.

Comando: fdisk
Descripción: Manejador de la tabla de particiones.

10.- Indique en que directorios se almacenan los comandos mencionados en el ejercicio 9.

/bin/

11.- Archivos

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

En linux todo es un archivo. Para saber que tipo de archivo es existe el comando "file".

b) Investigue el funcionamiento del editor **vi**, **mcedit**, el comando **cat** y **more**.

VI

Al invocar este editor aparece en el monitor la pantalla de edición. En ella aparece la posición del cursor resaltada, las líneas en blanco señaladas con el carácter ~ y en la parte inferior de la pantalla aparece la línea de estado, que muestra el nombre del fichero y el número de caracteres que contiene.
Si se invoca el vi pasándole como parámetro el nombre de un fichero en la pantalla de edición aparecerá su contenido. Cuando se invoca este editor con el nombre de un fichero que no existe, se crea automáticamente.
Existen dos modos de operación en el vi:
• **Modo Edición:** Para añadir texto al fichero

• **Modo Comando**: Para introducir órdenes que realizan funciones específicas del vi.
Cuando se edita un fichero con el vi, los cambios no se hacen directamente sobre el fichero. En realidad, se aplican a una copia del fichero que el vi crea en un espacio de memoria temporal llamado buffer. La copia en disco del fichero se modifica sólo cuando se graban los contenidos del buffer.
Para terminar la sesión caben varias posibilidades, siempre en modo comando:
:q Salir cuando no se han hecho modificaciones
:q! Salir y descartar los cambios
:x Salir y guardar los cambios

MCEDIT
Midnight Commander (mc) es un gestor de ficheros ortodoxo para sistemas tipo Unix (también existe para la plataforma Windows) y es un clon del Comandante Norton. Midnight Commander es una aplicación que funciona en modo texto. La pantalla principal consiste en dos paneles en los cuales se muestra el sistema de ficheros. Se usa de un modo similar a otras aplicaciones que corren en el shell o interfaz de comandos de Unix. Las teclas de cursor permiten desplazarse a través de los ficheros, la tecla insertar se usa para seleccionar ficheros y las Teclas de función realizan tareas tales como borrar, renombrar, editar, copiar ficheros, etc. Las versiones más recientes de Midnight Commander incluyen soporte para el ratón para facilitar el manejo de la aplicación.
Midnight Commander posee características tales como la capacidad de explorar el contenido de los ficheros RPM, trabajar con formatos de archivos comunes como si de un simple directorio se tratasen. Incluye un gestor de transferencias FTP o cliente del protocolo FISH.
También incluye un editor llamado mcedit. Mcedit es un ejecutable independiente, el cual también puede ser usado de forma independiente a Midnight Commander. Esta aplicación permite visualizar el contenido de ficheros y disfrutar de características como la de resaltar la sintaxis para ficheros de código fuente de ciertos lenguajes de programación, y la capacidad de trabajar tanto en modo ASCII como en modo Hexadecimal. Los usuarios pueden reemplazar mcedit con el editor que prefieran.

CAT
Con este comando podemos ver archivos, también se pueden crear archivos o bien podemos concatenar archivos.
Opción de cat

cat Con este podemos ver el contenido de un archivo.
cat > Es posible crear un archivo y comenzar a tipearlo.
cat -b Muestra un archivo, indicando sus números de líneas sin numerar aquellas que se encuentran vacías y que son de espacios, esto selecciona automáticamente la opción "-n".
cat -e Muestra el archivo, marcando su fin de línea con el signo $, esto selecciona automáticamente la opción "-v".
cat -n Muestra el archivo, con todas sus líneas numeradas secuencialmente de 1 en 1, comienza desde el número 1.
cat -r Reemplaza varias líneas vacías consecutivas por una sola línea.
cat -s Con esta opción suprimimos todos los mensajes de error, sobre archivos no existentes, nombre idénticos tanto de entrada como salida.
cat -t Muestra un archivo, indicando el uso de los tabuladores mostrándolos con los signos ^I, esto selecciona automáticamente la opción "-v".
cat -u Deshabilita el uso del buffer y usamos el manejo de caracter por caracter.
cat -v Muestra el archivo, con los signos de $, ^I, ^?, es lo que se llama con sistema hablador (verbose), muestra todo los códigos que está usando el archivo, puede resultarnos inlegible o incomprensible.

Ejemplos de Uso del comando cat

.- Crear un archivo nuevo, (se termina con ctrl+Z).
cat > archivo-destino

.- Lee y abre el archivo eliminando todas las líneas de espacio redundantes.
cat -r nombre-archivo

.- Concatena uno o dos archivos en el primer archivo.
cat archivo1 archivo2 > archivo1

.- Concatena uno o dos archivos a un tercer archivo.
cat archivo1 archivo2 > archivo3

MORE

El comando **more** es un visor de archivos. Este comando permite revisar un documento, página por página, con relativa destreza. Tiene una serie de macros que permiten hacer búsquedas, navegación y ejecutar un editor de textos (vi).

Macros del **more**:

espacio o **f**	avanza una pantalla.
enter	avanza una línea.
q	sale de **more.**
b	atrasa una pantalla.
'	va al sitio donde se inicio la última búsqueda.
=	Muestra el número de la línea actual.
/cadena	busca la ``cadena'' de caracteres.
n	busca de nuevo la última ``cadena''.
! comando	ejecuta el *comando* en una concha.
v	Inicia el editor **vi** sobre el documento que se tiene abierto.
h	Ayuda.
^l	Refresca la pantalla.

Ejemplo:

yemanya%more video.tex
\documentstyle{article}

\begin{document}

\thispagestyle{empty}
\title{Video de CeCalCULA}

\section{Introducci\'on}

--More--(2%)

El comentario -More-(2%) indica que sólo se ha mostrado el 2% del archivo.

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

File se utiliza para saber de qué tipo es un archivo, bien porque carezca de extensión que lo identifique (txt, mp3, png…) o bien porque haya sido renombrado, este comando nos dirá de qué se trata, o si es un directorio.
Veamos cómo utilizarlo con algunos ejemplos:
file guia.pdf
guia.pdf: PDF document, version 1.4
O bien:
file notepad.png
notepad.png: PNG image, 128 x 128, 8-bit/color RGBA, non-interlaced
También podemos obtener un poco más de información con la opción -i:
file -i guia.pdf
guia.pdf: application/pdf; charset=binary
file -i notepad.png
notepad.png: image/png; charset=binary
Ahora vamos a cambiarle a uno de estos archivos su extensión a ver si podemos engañarle, renombraremos notepad.png a notepad.txt:
file notepad.txt
notepad9.txt: PNG image, 128 x 128, 8-bit/color RGBA, non-interlaced
Otra opción también muy interesante es saber que hay dentro de un archivo comprimido. Para este ejemplo escogí un fichero de mi disco duro y lo comprimí con la extensión bz2.
file nino.bz2
nino.bz2: bzip2 compressed data, block size = 900k
¿Y qué habrá dentro? Ahora es cuando vamos a utilizar la opción -z para obtener más información
file -z nino.bz2
nino.bz2: RIFF (little-endian) data, AVI, 720 x 384, 25.00 fps, video: XviD, audio: Dolby AC3 (6 channels, 48000 Hz) (bzip2 compressed data, block size = 900k)

Fabrizio Gelsi