Teoría Tema 3.

El arranque:

Lo realiza un pequeño programa que se llama BIOS. Es un programa de 16 bits o 32. Las BIOS tradicionales no pueden gestionar más de 2 GB de RAM, porque estaban hechas de 32 bits, por eso más RAM para un PC de 32 bits es una tontería.

La EFI BIOS no tiene ninguna limitación, porque es la nueva para sistemas informáticos de 64 bits. Con esta se puede gestionar toda la RAM que quieras, los límites están muy altos.

No confundir tres cosas: el chip donde está la memoria ROM, que después se convirtió en EPROM y ahora está alojada en el puente norte. Esto no es la BIOS, esto es la RAM CMOS. La BIOS, es el programa, es la instancia de ese programa. Y tampoco confundir con la RAM CMOS SETUP, que es el programa de modificación para cambiar los parámetros de la BIOS.

Un PC para funcionar necesita del software, siempre. La BIOS es ese programa principal y básico que nos va a cargar el SO. El SO necesita un programa para que se cargue, no se puede cargar el sólo. Es lógico que esté integrado en la placa. Lo primero que hace el procesador es buscar este programa.

El procesador, busca la BIOS a través de los buses, donde la placa le dice que está. Eso hace al hardware totalmente independiente del SO que vayamos a ejecutar.

Cuando se ejecuta la BIOS, se hace una serie de cosas. Lo primero que hace, es cargar un pequeño manejador primario de pantalla. En las placas antiguas ese manejador primario va a estar en la propia placa formando parte de la BIOS. Pero hoy en día, las tarjetas gráficas tienen su propia BIOS (sea externa o integrada). La BIOS ya lo que hará es acceder a la BIOS de la gráfica para cargar ese pequeño manejador primario de pantalla.

Una vez cargado el controlador gráfico mostrará cosas por pantalla y comenzará a hacer test. Los test más básicos. Hace test a los buses de entrada salida, memoria, procesador, etc. Esto es el POST → AUTOTEST DE ENCENDIDO. En este momento es cuando aparecen los pitidos (aunque se prevé que desaparecerá porque las EFI son muy grandes y se implementará una nueva manera de detectar errores). Si la memoria está mal colocada o está realmente mal, no se va a cargar nada. No se muestra nada por pantalla, y empieza a pitar como loca la pantalla.

- A veces es difícil detectar si un fallo es de placa o de memoria. Si tienes dual CHANNEL, desconectando un módulo y probando el otro en ese slot te das cuenta de si es la placa o la memoria. Es muy difícil que dos módulos de memoria fallen.

El POST tiene una segunda fase en la que empieza a lanzar mensajes. Muchas veces polladas. Son los mensajes de F1 para continuar. El error de OverClocking te dice que los valores se han vuelto por defecto.

Después se cargan en memoria unos controladores muy básicos para el funcionamiento más básico. Se accederán a los discos duros de la manera más básica, no como SATA; sólo pantalla, teclado, buses, unidades de almacenamiento para buscar algo que cargar, el USB y ya está.

Después aparece un cuadro, que reconoce e inicializa todos los dispositivos PnP (plug and play). Para el PC, todo lo que no es principal, mencionado anteriormente son cosas plug and play. Aunque estén conectados a la placa.

Al final te ofrece la opción de cargar la RAM CMOS SETUP. Que es el programita para cambiar parámetros de la BIOS.

Una vez hecho esto, buscará el código de arranque. Este código, será un script que nos va a dar la pista de donde está el SO. A la BIOS ya solo le queda buscar el SO y cargarlo en memoria. Cuando el SO se cargue irá sustituyendo los manejadores primarios por los suyos específicos. A partir de aquí ya tendremos sonido, por ejemplo.

Las BIOS antiguas tienen 4 colores, modificables con 2 bits.

Las EFI BIOS (Interfaz extendible del Firmware), es un sistema desarrollado para sustituir poco a poco a las BIOS, incompatibles con grandes memorias. Las BIOS usan instrucciones muy cortas, y no pueden gestionar grandes cantidades de memoria principal ni almacenamiento (Arquitecturas de 16 y 32 bits). Conforme los SO comenzaron a requerir más memoria y a sobrepasar el límite gestionable de 8 TB (particionado en 2 TB), se fueron sustituyendo.

El funcionamiento es similar a la BIOS. Ofrece un marco de emulación de sistemas operativos antiguos. Esto Windows 8 ya necesita la EFI para funcionar, no funcionaría en una BIOS. Windows 7 ya no funcionaría en una EFI, pero gracias al marco de compatibilidad de la EFI, funcionan adecuadamente. La EFI se capa, para hacer posible que se instale un SO antiguo.

EL EFI tiene funciones mucho más avanzada, la RAM CMOS es mucho más grande, con muchas más funciones. Las EFI pueden gestionar hasta 9 ZB (1 ZB = 1.000.000.000 TB). En Arquitecturas de 16 bits tenemos un marco de ejecución de 640 K y 2 TB para gestionar; las EFI suponen un gran avance.

La BIOS decimos que es tipo Scripts, nos referimos a que es 100% estructurada. Son todo instrucciones sin funciones, sin módulos. La EFI, es modular.

La EFI, buscará el código de arranque, que esta vez no tiene por qué estar en el disco duro. Podría estar en la misma EFI. Esto permite hacer cosas como que se permita usar el PC sin un SO siquiera. En las BIOS antiguas se necesitaba que el código de arranque estuviera en un disco duro, o alguna unidad de almacenamiento.

El firmware de la EFI va a estar en la RAM CMOS usualmente (esto se permite porque la EFI es modular, y podría tener los módulos en diferentes sitios). La EFI tendrá un selector que vaya cargando los módulos. Esos módulos pueden estar en unidades de memoria sólida, hdd, unidades auxiliares como un Pendrive, etc. En caso de usar un Pendrive, y de cargar un módulo en él, el Pendrive actuaría de llave. No arrancaría el PC sin el Pendrive.

Los SO nuevos, traen sus propias EFI en el instalador. Se tiende a que la placa base cada vez tenga menos manejadores. Cada vez, el firmware se está llevando más hacia el software. En este momento, el Firmware es prácticamente buscar una EFI (en eso consiste el firmware).

LA EFI INTEGRA LA PRIMERA ETAPA DEL GESTOR DE ARRANQUE. LAS BIOS ANTIGUAS SIEMPRE BUSCAN EN EL MBR Y DEPENDIENDO DE SI EL SO ES ANTIGUO O NO, IRA A BUSCAR EL CODIGO DE ARRANQUE EN EL PBR O BUSCARÁ LA 2DA ETAPA DEL GESTOR DE ARRANQUE. CUANDO EL SO SEA MEDIO MODERNO, EN EL MBR ESTARÁ LA PRIMERA ETAPA DEL GESTOR DE ARRANQUE Y EN EL PBR LA SEGUNDA.

Código de arranque:

También se llama bootstrap y va a ser una cosa muy pequeña, como un pequeño mapa del tesoro que localiza el SO y lo va a ejecutar. Este código puede ejecutar programas muy básicos independientes. Usa la BIOS como SO (pequeño). Lo que podemos hacer es que a través de la BIOS o bien ejecutamos el SO o bien ejecutamos alguna aplicación.

El maintest del grub de Ubuntu por ejemplo, es un pequeño programita que realiza un test a la memoria cargando esos pequeños programas independientes y muy primarios.

Otros códigos, cargan un pequeño SO, que en realidad es un pequeño módulo de Linux y desde ahí se cargaran los programitas primarios.

La parte de la BIOS o la EFI la parte que busca este código en el sistema se llama cargador de arranque o bootloader.

EL bootloader lo primero que hará es cargar el primer código de arranque almacenado en la RAM CMOS o en la EFI o en la BIOS. El bootloader depende del hardware totalmente y va a buscar el mapa. Y el código de arranque es el mapa.

Los mapas o códigos se hallaran en el primer sector de dispositivo de almacenamiento de unidades de almacenamiento. Estará en el primer sector lógico, si está, obviamente.

Si lo buscamos en una unidad de almacenamiento particionada, lo buscará en el primer sector lógico

Nota: en el caso de las BIOS, la BIOS siempre va a estar en el chip y el código de arranque siempre estará en un HDD. En las EFI, el código de arranque está en la propia EFI (RAM CMOS) y el programa, lo que es la EFI en sí, puede estar en varias partes.

El código de arranque, ese mapa que nos dirá dónde está el sistema operativo, lo vamos a encontrar en el primer sector (de una unidad o una partición, dependiendo de cómo este el disco duro). El bootloader lo que va a hacer es buscar ese código de arranque.

En el caso de los sistemas basados en Unix, incluido Linux, el código de arranque no existe y se usa un gestor de arranque. Esto quiere decir que el primer sector donde se encontraría ese código está vacío.

Tradicionalmente, el primer sector del disco duro va a ser reservado para el MBR (gestor de arranque maestro) que tendrá el código de arranque y la tabla de particiones. Este primer código de arranque nos va a decir dónde está el SO instalado. En la partición en cuestión estará el PBR que es el código de arranque del propio SO y ese ya lo que hace es cargar el SO. Eso es lo tradicional; pero no lo que ocurre ahora.

Ahora, lo que ocurre es que el PBR de la partición donde está el SO estará vacío, no habrá nada. Y en el primer código de arranque del MBR, estará la primera etapa del gestor de arranque; y esa llamará a la segunda tapa del gestor de arranque. La segunda etapa del gestor de arranque, si estamos en Windows lo tendremos en un archivo (dependiendo de la versión) o en una partición. El archivo "bootmgv", o una partición de 100 MB que se crea automáticamente.

En Linux, se va a guardar en /boot. /boot podría estar dentro de una partición o bien estar formando parte del sistema de archivos normal. El gestor de arranque más conocido de Linux es el Grub. Antiguamente se usaba otro, conocido como LILO.

El cambio de código por partición, va desapareciendo porque cada vez el arranque es más complicado.

En el caso de las EFI, ya se hace diferente. Aquí es la EFI la que busca el SO. La EFI está en un chip, no existe ya el MBR y ella misma se encarga de buscar al SO. La primera etapa está en la propia EFI. Esto es fundamentalmente el motivo de que no se pueda usar las EFI para SO antiguos y raros, porque la primera etapa en esos SO estará en el MBR y la EFI ahí no busca. Por eso en los casos en que tengamos una EFI y el SO sea antiguo, debemos desactivar el arranque por EFI. Aunque se arranque por EFI, los SO siguen teniendo su gestor de arranque (Windows su propia partición y Linux su carpeta /boot).

Arranque de red:

Para que se pueda hacer un arranque de red, primero tiene que haber un firmware en la placa capaz de arrancar por red. Antiguamente, las tarjetas de red traían ese firmware, hoy en día casi todas las placas lo llevan integrado en el puente sur. Antiguamente, había que usar una tarjeta de expansión con el conector de red, que incorporaba una interfaz poder efectuar el arranque. Hoy en día o forma parte del código de la BIOS, de la EFI, o estará embebido en el puente sur. Antiguamente había que dotar al equipo con una tarjeta de expansión.

El arranque por red normalmente tiene dos etapas. Primero se usa el protocolo de descubrimiento. Ese protocolo es el DHCP. El DHCP manda mensajes sin rumbo para ver quien escucha. Este protocolo usado para este fin es muy lento. El DHCP es un tipo de broadcast.

A través del software inicial, se descargan los ficheros de arranque y se arranca el SO. Para descargar el software inicial se usa el Trivial FTP (que es una versión muy simple de FTP) que es el protocolo que se usa para descargar ficheros de internet. Hoy en día se usa más el FTPS, FTP seguro.

El arranque de red, lo que pretende es que el SO no arranque de forma local, sino que arranque en red. Que haya un servidor que tenga los ficheros necesarios para arrancar los equipos.

Cuando vamos a instalar un SO, el SO es realmente el que se auto instala. Normalmente, tu metes el disco, se carga en memoria una versión del SO más ligera, y se auto instala. Del mismo modo, cuando damos a probar versión de prueba en el caso de Ubuntu, se carga en memoria y te deja trastear.

Según el arranque de red, se podría prescindir de un disco duro local. Sería parecido a probar Ubuntu, ese tipo de arranque.

El arranque de red se puede desactivar desde la BIOS.

Particionado

Registro de arranque maestro (MBR Master boot records)

Este sistema es muy tradicional y se está abandonando. El Arranque maestro se ubica en el primer sector físico. Este tipo de gestor se encontrará en unidades que se puedan particionar, es decir HDD (unidades magnéticas) y unidades flash.

Normalmente será el primer sector físico y el primer sector lógico.

En los primero 446 B vamos a tener el área de código. Ahí tendremos la primera etapa del gestor de arranque, en sistemas antiguos. Ahí estará el código. Evidentemente si ese disco duro lo usamos para almacenamiento de datos, ese sector estará vacío. Este primer sector estará ocupara 512 B.

Esto normalmente se da en antiguos sistemas Windows. En Unix, siempre se usó gestor. Podría tener un gestor de arranque muy simple, o bien el código de arranque. El gestor es múltiple, si tenemos más de un SO reconocibles entre sí, te aparecerá el correspondiente menú.

En los siguientes 60 B, vamos a tener la tabla de particiones. En esa tabla de particiones, solo caben 4 entradas para mapear 4 particiones. Cada entrada tendrá 16 B. En esos 16 B se indicara el tamaño y el principio de cada partición, además de unos B para indicar el tipo. Solo puede gestionar 4 particiones de 2 TB cada uno.

Por ultimo vamos a tener 2 B (16 bits) en los que estará la firma de arranque. Aquí se va a guardar en los antiguos Windows, la marca de arranque. Cuando un disco es de arranque, se le llama activo. Para que la BIOS reconozca un disco de arranque, hay que marcarlo como activo. Hoy en día no sirve para nada.

Gestor de Arranque (Second-stage bootloader)

Va a formar parte de la segunda etapa. El primer bootloader, va a estar en la BIOS. No confundir el Bootloader con la BIOS. El primero, buscará la primera pista. El primer bootloader, carga el MBR, y este después va y carga el Gestor de arranque del segundo bootloader.

PBR = Registro de arranque de partición.

En todos los discos la tabla de particiones del MBR tiene que estar. En un sistema antiguo, tradicional BIOS, este va a buscar en la TP el código de arranque. Esto es la primera etapa. El código de la BIOS que hace esta parte es el bootloader de la primera etapa.

La EFI, pasa del código de la TP en el MBR, y va directamente al bootloader de la segunda etapa. Es decir, la EFI, detecta el bootloader de la segunda etapa a través del bootloader de la primera etapa.

El bootloader de la primera etapa buscara el bootloader de la segunda etapa, que ya tiene como misión cargar el SO.

Los sistemas antiguos no tienen bootloader. Directamente la BIOS lee la TP del MBR (el código) y ése código, carga directamente el código que está en el PBR, que ya carga el SO.

Windows XP/Vista/7 carga el SO con la BIOS usando el MBR, que después busca el segundo bootloader. El W8 por ejemplo, ya no usa el MBR, el bootloader de la primera etapa se salta el paso de consulta a la Tabla de particiones y va directamente al segundo bootloader.

En los antiguos Windows el código de arranque está en el PBR, Ahora está vacío. En los sistemas modernos el PBR está vacío, y el ordenador arrancara con el gestor de arranque que o está en una partición o está en el sistema de archivos.

En los sistemas Windows, la segunda fase está en la partición de sistema o en una partición creada exclusivamente para ello durante la instalación. Si está en la partición de sistema, estaría en un ficherito. A día de hoy tiende a estar en la famosa partición de 100 MB. Esa unidad suele estar oculta, para que no tengamos accesos.

En sistemas basados en UNIX estará en /boot. En Macintosh parece que también es así.

El NTLDR es un gestor de arranque de Windows XP. Se lleva usando desde el Windows NT.

El BOOTMGR para Windows Vista/7/8/Server 2008/2012.

Si se diera un fallo, lo más probable es que el gestor de arranque esté mal. El gestor puede estar en dos sitios. Puede estar en el MBR o en el SO. Si el fallo está en el MBR, hay un comando cmd que se llama FIXMBR. Con eso arreglaríamos el problema. Si el fallo está en la segunda etapa, podríamos usar el FIXBOOT. Normalmente hay que usar los 2, porque si se hace un cambio en uno, después el otro se tiene que sincronizar. Puede que esos comandos solo estén disponibles en la consola de recuperación. Posiblemente el comando FIXMBR no esté disponible en Windows 8 porque usa EFI.

Si ejecuto esos comandos, para arreglar el arranque, y tenemos un sistema dual (dos SO, ejemplo Linux aparte de Windows), se sustituirá el Grub por el gestor de arranque de Windows, lo que haría inaccesible a Linux. ¿Solución? Vuelves a Instalar el Grub 2 después de arreglar el gestor de arranque de Windows.

Nota: Los comandos FIXMBR y FIXBOOT se usan para Windows XP. En Windows Vista/7/8 se usa el comando BOOTREC [/FIXMBR] [/FIXBOOT].

LILO es un gestor de arranque de Linux. El más antiguo es el LILO, es el tradicional. Ha sido sustituido por el GRUB, que va por la segunda versión.

Syslinux otro gestor de arranque de Linux.

Tipos de particiones MBR

En sistemas BIOS solo 4 particiones primarias de 2 TB cada una. La tabla de particiones solo tiene 4 entradas, una por cada posible partición.

Cada HDD tiene 1 partición como mínimo para ser aprovechable. El primer paso que hace el SO al formatear es crear una tabla de particiones y una partición como mínimo.

El primer sector de la partición estará reservado para el PBR. El primer sector de cada partición se reserva para el PBR. Y el primer sector lógico es el MBR.

Las particiones extendidas. Las particiones primarias, se redefinen y la cambiamos por una partición extendida. La extendida es otro tipo de partición. Las particiones extendidas se pueden particionar, hasta 23 veces. La partición primaria estará mapeada en la TP principal, y las particiones lógicas que se hacen dentro de las extendidas se mapean de diferente manera. En total podríamos tener 26 unidades lógicas. Un hdd solo puede tener 1 partición extendida. 26 unidades lógicas, y 3 primarias.

¿En total cuanta memoria se puede aprovechar? En total se puede gestionar 8 TB, porque aunque la extendida se pueda particionar hasta 23 veces, el tamaño real de la partición es 2TB.

A efectos prácticos, no hay diferencias entre particiones lógicas y primarias.

Para mapear la partición extendida, se usa un registro de arranque extendido (EBR) que va a estar en el primer sector lógico de la partición extendida. Va a ser muy parecido al MBR, pero se diferencian en que dentro de la extendida, vamos creando particiones lógicas. Como mínimo, se crea 1 dentro de la extendida, después se va cambiando. El primer sector es el sector de arranque extendido y cada partición lógica tiene su propio PBR.

Aunque la partición sea de datos, siempre va a tener el PBR, vacío, pero lo tendrá.

Registro de Arranque extendido (EBR)

Va a tener la misma estructura que la del MBR. La tabla de particiones que se usa, solo gestiona una partición. La primera entrada, mapea una partición, pero la siguiente entrada, va a decir donde hay un nuevo EBR; creándose una especie de lista enlazada. La primera entrada de la tabla te mapea la partición lógica, y la segunda entrada te envía al siguiente EBR. Se va creando un EBR por cada partición lógica.

En las particiones primarias, el mapeo es estático, mientras que las extendidas lleva un mapeo "dinámico".

Tabla de partición GUID o (GPT)

Fue implementado para ser usada por las nuevas BIOS, las EFI. Las tablas de particiones tenían que mapear grandes particiones. Las EFI van a tender a usar GPT, y las BIOS van a usar MBR. Se va a usar MBR a no ser que las particiones sean muy grande.

GUID es un identificador hexadecimal, que identifica a un componente lógico de hardware de una manera univoca. Está pensado para identificar cualquier elemento hardware.

Las GUID mantienen la compatibilidad con los antiguos MBR. Quiere decir, que si pongo una GUID en una BIOS antigua será compatible, o casi compatible (dependiendo de si me excedo del límite de 2 TB, normalmente el espacio de particionamiento capara las particiones a 2 TB). Si tengo un hdd de 3 TB gestionado con GUID y después lo usa una BIOS antigua, se capará los 3 TB a 2 TB y seguro que me quedan datos corruptos.

El GPT, se duplica. La tabla de particiones es algo muy delicado e importante y por ello se hace esta copia de seguridad (redundancia). Se ponen al principio y al final del disco, por protección. El GPT siempre usará sectores lógicos (LBA), nunca CHS.

El primer sector lógico (0) de un GPT, contendrá el MBR (una copia, por compatibilidad, en realidad es un MBR falso por si alguna vez uso una BIOS en lugar de una EFI). En el sector lógico 1, tendremos una cabecera, en la que se declaran que particiones van a haber (no se mapean, solo se declaran).

En los sistemas modernos los sectores son 4 KB. Como máximo, Windows permite hacer 128 particiones (todas primarias, porque aquí ya no existen las particiones extendidas).

En los sectores lógicos del 2 al 33, están las entradas a las particiones, ahí es donde está el mapeo de particiones.

Al final del disco lógico, esta estructura se duplica.

El GUID es único que no se repiten en la vida. Esto se consigue, elevando números primos entre ellos. Video del hotel infinito (lo explica).

Discos dinámicos

Un disco dinámico puede expandirse y reducirse, pero además puede usar espacio no contiguo. El particionamiento estático, si no lo realizas bien te puede dar problemas de fragmentación externa. Sin embargo, con el sistema de discos dinámicos, se resuelve. El mapeo es más complejo. Es un tema de Windows, y solo funciona bien en Windows (esto está hecho realmente para servidores, no para particulares).

En un dispositivo de almacenamiento que se pueda particionar, evidentemente, un disco dinámico va a permitir hacer virguerías.

Se cambia completamente la terminología. Cuando hablamos de discos dinámicos, hablaremos de "volúmenes" no de particiones. Ahora el PBR no se llamará así, se llamará el VBR (registro de arranque del volumen). Probablemente ahí se guarde la información del mapeo tan complejo que tienen (aunque no se sabe).

El particionamiento básico, ya sea MBR o GPT, sigue estando ahí.

Los discos dinámicos, se comenzaron a usar en versiones de 64 bits de XP. En otros sistemas operativos puede haber herramientas para crear discos dinámicos, pero no funciona realmente bien.

En los discos dinámicos, los volúmenes pueden ser:

- Simple: equivalente a una partición primaria.
- Distribuidos: capaces de expandirse en espacios no contiguos.
- RAID: acceso simétrico, tipo dual channel.

En los RAID la velocidad de acceso se duplica, triplica, etc. Hay RAID de diferentes tipos.

- RAID0: seccionados. Aumenta la velocidad de transferencia.
- RAID1: reflejado. Tiene redundancia, los sectores se copian en los discos. Se busca seguridad.
- RAID5: no hay redundancia, pero tiene un sistema de paridad y corrección de errores.

Discos dinámicos

Los volúmenes simples son los más sencillos. Cuando convierto un disco básico en un volumen simple de esta forma es la más simple. La diferencia que tiene con los discos básicos, el particionamiento básico, es que en un volumen simple no tengo por qué tener el espacio contiguo.

En Windows, a pesar de que el espacio no es contiguo, a todo ese espacio se le asignara una letra y se trabajara con el como una unidad lógica.

Al ser un disco dinámico, se podrá ampliar y reducir de manera dinámica, sencilla y fácil. Sin ser peligroso (a diferencia del particionamiento básico).

Los volúmenes se pueden crear directamente, o convertir particiones básicas en volúmenes. Los volúmenes simples, tienen que estar en el mismo disco (las particiones o espacio que ocupe, en el mismo disco).

El volumen distribuido es parecido al volumen simple, pero se puede extender a varios discos. La gestión es un poco diferente, ya que conlleva acceder a tablas de particiones de diferentes discos.

Para hacer esto posible, a la fuerza debe de haber una capa de abstracción por encima de los bloques LBA de cada disco. Tiene que haber unos bloques más lógicos todavía, el sistema tendrá que tener algún modo de hacer los sectores LBA en otro tipo de sectores lógicos que unifique las particiones de los diferentes discos.

Esto es más rápido en realidad, porque imaginemos que vamos a copiar un documento en el Volumen F que está en diferentes discos. Si tú copias un fichero dentro de la misma partición, el cabezal del disco debería de turnarse pero si el dato cayera en la partición de otro disco, sería mucho más rápido.

Los sistemas RAID combinan acceso simétrico y redundancia. Tipos de RAID hay un montón, pero los que usa Windows son los que nombramos anteriormente (los más comunes). A nivel de pequeñas o medianas empresas quizá lo encontremos.

El RAID se caracteriza por el acceso simétrico, aunque se diga que hay redundancia (depende del RAID tendremos redundancia). La ventaja sobre la velocidad de transmisión solo es útil en servidores, no tiene sentido en particulares. Lo que determina la eficiencia de un disco duro a nivel de particulares es el tiempo de acceso, la latencia, porque un cliente la mayoría de las operaciones que hará supondrá hacer pequeñas accesos; y la latencia no se mejora con estas técnicas. En servidores si hay que mover grandes flujos de datos, y tienen un ancho de banda, los clientes no tanto.

El RAID se implementa de dos formas:

- Por hardware: la placa base están preparadas (algunas), con un controlador RAID integrado (las modernas, antiguamente se tenía que hacer con un adaptador de host externo). De hecho el SATA tenía 3 modos de actuar, y entre ellos estaba el RAID. Los 3 controladores capaces de funcionar en modo RAID son los SCSI, los SATA y los SAS. Como el SAS es muy compatible con SATA, es de costumbre que los SAS usen el modo RAID con discos duros SATA.
  Los discos se van a usar enteros, y de las mismas características (tipo y velocidad de transmisión, a ser posible igual). Si se tuviera un disco más grande que otro, lo que le sobra al más grande se pierde, no se usa. Y si ese que es más grande, fuera más rápido, la velocidad se capa, por lo que también se desperdicia la velocidad.
  Para gestionar el RAID, cojo agrupaciones de sectores lógicos y se crean secciones, y ese será el tamaño del nuevo bloque. De esa manera se genera más fragmentación interna en realidad, y se podría decir que se desperdicia memoria, pero da igual porque esto solo lo montan los del taco y lo que sobra precisamente es memoria.
  Después de crear las secciones, ya se particionan. El SO, detectara los discos capados como un único disco y trabajará con él como un único disco. Incluso desde la partición de discos.
  Quien capa los discos y crea las secciones y particiona el RAID es la BIOS. Lo hace el hardware, por lo que es más eficiente que por software.
  Esto es universal, siempre que se tenga el manejador. Este modo de funcionamiento es universal.
  No es flexible, no se pueden particionar ni trastear mucho con ellos.

- Por software: se crea el RAID mediante discos dinámicos. En este caso se pueden usar volúmenes en lugar de dispositivos enteros, es mucho más flexibles, pero por el contrario son menos eficaces (al hacerlo por software). No es universal, puede ser incompatible entre diferentes SO.
  Para implementarlo, necesitamos pasar los discos a dinámicos. Aquí no se usa todo el disco, se pueden usar particiones, no se usa todo el disco necesariamente. Los volúmenes pueden estar en diferentes discos. Ahora son los bloques los que se transforman en secciones. Las secciones se crean a partir de un conjunto de bloques.
  Aquí tendremos una unidad lógica, mientras que por hardware el SO lo detectara como una unidad Física desde el primer momento.

  Es muy versátil y flexible, pero poco portable e incompatible con diferentes SO y es menos eficiente. Aquí el particionado se hace antes, mientras que en los RAID físicos después.

  Los bloques crean las secciones y después se formatea creando nuevos bloques.

Nota: Un disco puede ser básico o dinámico. Cuando es básico tendremos el MBR o GPT. En MBR tendremos particiones primarias, extendidas y lógicas. En GPT solo tendremos particiones primarias. Los discos dinámicos tienen MBR y GPT y pueden tener particiones simples… (Mirar los tipos).

Nota: Cuando tenemos un disco dinámico todas las particiones se convierten en Volúmenes. El disco o es dinámico o es básico.

Los RAID hardware se hacían desde la BIOS y los software desde el SO. Hay muchos tipos de RAID. Los RAID tienen sentido en grandes empresas, quizá para una pequeña o mediana empresa que se dedique al software podrían tener uno. Nosotros veremos 3 tipos de RAID (compatibles tanto por hardware o software con un pc corriente):

- RAID0 o volumen seccionado: el número del RAID tiene que ver con la paridad (redundancia). 0 significa que no tiene redundancia, los datos no están repetidos. Este RAID solo tendrá acceso simétrico. Puede utilizar cualquier número de unidades (discos duros), es flexible. En rendimiento aumentará proporcionalmente según el número de discos. El RAID 0 es como si tuviéramos un dual channel en discos duros. Los discos duros, no son tan robustos como una memoria RAM.
  En RAM cuando se hace un acceso a RAM, lo normal es que no de errores, esa memoria es muy fiable, mientras que en discos duros hay una probabilidad muy alta de que lo que leamos no tenga sentido (porque es una unidad óptica, da errores). Un disco, por muy bien que este, presenta fallos, entonces se relee y ya está. Si se intenta varias veces y no se puede leer, es que la superficie esta chunga. En redes pasa lo mismo. En redes estas pidiendo paquetes todo el rato, la perdida de paquetes conlleva perdida de velocidad.
  Si implementamos esta técnica, hay más probabilidad de que se pierda información. Se pierde facilidad de acceso. Si tuviéramos 3 discos, multiplicaríamos por 3 la velocidad de transmisión, pero también multiplicaríamos por 3 la probabilidad de fallos durante la lectura. Si monto dos discos duros de 1 TB, se verá como un único disco duro de 2 TB.

- RAID1 o volumen reflejado: la paridad es total, datos duplicados al 100%. Vamos a tener varios discos duros, y los datos estarán reflejados, duplicados. Se puede usar cualquier cantidad de discos, pero es tontería tener más de 2 discos duros o volúmenes. Eso da seguridad de sobra. Mientras que el RAID 0 aumentaba la velocidad de transmisión tanto en lectura como escritura, en el RAID1 solo aumentara la lectura, la escritura no va a aumentar (velocidad). Como hay redundancia total, si tengo un disco duro de 1 TB y otro de 1 TB, al final tendré un disco de 2TB del que aprovechare 1TB, porque se va copiando todo. Aquí también tenemos es problema de duplicar la probabilidad de fallos durante la lectura.

- RAID5 volumen de paridad distribuida: lo ideal es hacerlo con 5 discos. Hay una redundancia parcial. Propiamente dicho, no hay redundancia de datos, y como mínimo se requieren 3 discos. La lectura y la escritura son tan rápidos como un RAID0. Cada vez que se hace una lectura, se comparará con una sección de paridad (una operación lógica). No es tan gastoso en memoria como el RAID1. Si intento leer y una de las lecturas de la sección de datos no está bien, la sección de paridad me va a ayudar a corregirlo (los errores al leer se tratan de corregir al instante). El hardware de host tiene que ser especial. Los clientes normales no implementan en realidad esto, los servers sí podrían hacer un RAID 5; pero los clientes no pueden y tampoco es usual hacerlo por hardware. Esta es la mejor versión de entre las 3 que hemos visto. Aumenta velocidad y fiabilidad y la perdida de tamaño por tener paridad tampoco es tan grande (33%).

Actualización

Hay que actualizar sobretodo en Linux porque no es lo mismo el sistema Windows, en el que hay muchos intereses comerciales y que está compuesto por una serie de informáticos de élite asegurándose de que no haya muchos errores; que en Ubuntu donde hay muchísimos más informáticos actualizando Ubuntu. A la larga, los SO de Ubuntu por ejemplo, van corrigiendo los fallos dinámicamente y al ser muchos, tienen también mucha capacidad para descubrir fallos y arreglarlos.

Al actualizar se añaden nuevas funcionalidades. Se mejoran antiguas funcionalidades. Se corrigen errores de programación, los bugs. Un bug es fatal, es un pequeño error que da mucha lata. En ocasiones son tan pequeños que ni siquiera se manifiesta. Se solventan problemas de vulnerabilidad. La vulnerabilidad no tiene nada que ver con los errores, una cosa es que existan fallos y otra cosa es que sea vulnerable. El SO más invulnerable es el MS-DOS; porque no tiene conectividad a otros equipos. Cada vez que se crea un servicio para proveer conectividad, estas creando potencialmente un agujero de vulnerabilidad.

Hay dos tipos de bugs, los bugs críticos, que son los que afectan al funcionamiento gravemente. Por ejemplo algo que deje colgado el programa, pierda datos, etc. Después también están otro tipo de bugs, los glitchs, son sobretodo estético. Fallos superficiales. Los glitch se supone que son inocuos.

En el desarrollo de una aplicación, y en el periodo de prueba hay tres fases. La alfa, la beta y la gamma. La alfa es el periodo de prueba en el que tú y tu equipo probáis a buscar los fallos. Después, la beta es la fase en la que lo prueban personas ajenas al proyecto y se pide a los usuarios que se notifiquen los errores y la gamma ya es la versión que se comercializa. La

gamma, es la versión comercializada ya, la que sale a venta y está lista. Las actualizaciones actúan en esta fase ya.

Al actualizar, debemos hacerlo con actualizaciones oficiales. En Linux, están las actualizaciones de la comunidad, pero hay otros soportes que no son de la comunidad (a través de repositorios). Con los móviles pasa eso, más o menos, en Android esta todo diseñado para que las aplicaciones se descarguen desde el repositorio oficial; aunque sabemos que hay una opción que te habilita la descarga desde terceros.

Las actualizaciones hoy en día necesitan acceso a internet. Las actualizaciones estarán en repositorios. Los repositorios son servidores especiales donde están los paquetes.

Antiguamente, para actualizar algo necesitabas un disco de actualización.

Tipos de actualizaciones:

- Actualización crítica: las que corrigen errores graves.
- Actualización recomendable: aportan funciones nuevas u optimizan cosas que aunque no funcionaran mal, comenzarían a funcionar mejor.
- Opcionales: no se instalan automáticamente, ni te avisan ni nada. Añaden nuevas funcionalidades.

*** Cuando se actualiza, los archivos se cambian, pero hasta que no se reinicien los antiguos no dejaran de estar cargados en memoria. En el caso de actualizaciones críticas, por eso se requiere que reinicie el equipo.

Paquetes de distribución.

En el paquete de distribución estará todo lo necesario para instalar la aplicación o lo que sea. Puede tener más cosas, pero lo más importante que tiene es un script de las cosas que hay que hacer.

En algunos casos ese paquete tendrá dentro todos los archivos de instalación y la información. Es el caso en el que te descargas un ejecutable. Otras veces el paquete buscara los archivos necesario en una ubicación remota.

El paquete normalmente necesita una herramienta de instalación. En Windows, Windows installer. En Ubuntu, o debían, se usa el apt-get en entorno texto o el synaptic. En los móviles tenemos el google play.

La instalación en Linux es más sencilla que en Windows, que como es modular, tiene q instalar librerías y demás.

Actualmente se usa el Windows installer para instalar. Es simplemente la parte del sistema operativo que cuando se encuentra con un paquete de instalación realiza las acciones necesarias. Hay otras opciones, como el ccleaner que tiene una herramienta para leer los paquetes de distribución.

Los archivos pueden estar dentro del paquete de distribución o fuera del paquete de distribución dependiendo del tipo.

- Paquetes .msi: son los más normales. Son los que instalan cosas.
- Paquete .mst: para modificaciones. Serán archivos de actualización.
- Paquete .msp: también es para modificaciones. Son para actualización.
- Paquete .msu: para actualizar el SO, asociado a Windows update.

- Paquete .cab: son paquetes comprimidos con archivos dentro. Son famosos, pero solo se usan para tener archivos de información dentro; útiles para la instalación.
- Paquete .zap: solo tienen texto plano con ubicaciones de destino de otros ficheros en red.

El setup.exe corriente que conocemos, es el que llama a estos paquetes de distribución. Puede ser que el propio paquete de distribución este dentro del ejecutable. Eso es muy típico. Vas a una página y te descargas un .exe "instalador". Lo que hace el exe realmente es descomprimir todos los ficheros y hay dentro y después ejecuta el instalador.

El paquete de instalación normalmente incluye claves de configuración de registro de Windows. Son los archivos .reg. Cuando se ejecutan, intentan insertar en el registro de Windows una clave.

Con el comando REGEDIT ejecutamos el editor de registro de Windows. Si quiero cambiar manualmente cosas en el registro, teniendo en cuenta que es una gran bdd, podría exportar algo del registro y así hacer una pequeña copia de seguridad en un fichero .reg. Si haces doble clic, se restaura.

En los sistemas basados en debían (ej. Ubuntu). La herramienta de instalación, a bajo nivel va a ser el dpkg. Este comando es de bajo nivel. El comando init por ejemplo, lo que hace es cambiar los modos de "arranque, desconexión, explotación, etc.". Si pones init 0, es para apagarse, el 1 para resetearse. En Linux, para que un usuario no tenga que aprenderse los números, se crean comandos para que llamen a ese init 0. El reebot por ejemplo capaz y llama al init 0 (ejemplo ultimo no real, por poner).

El dpkg es otro ejemplo, como el init. Es una herramienta con las que se instalan los paquetes de debían. Para no acordarnos de la sintaxis, se desarrollaron otras herramientas como el apt, que es más fácil de recordar. El apt- llamaría al dpkg, pero ahora más por encima, a más alto nivel, tendremos el synaptic.

Los archivos de instalación están en .deb. En Linux, no hacen falta extensiones, estos .deb no sería realmente necesario, aunque los lleva.

Los paquetes deb se pueden descargar de los repositorios, habrá repositorios oficiales y repositorios no oficiales. Los oficiales serán de Ubuntu, y los no oficiales pueden atender a intereses comerciales o a cuatro enteradillos. Normalmente los oficiales son los buenos. Pero en el software libre, que sea oficial o no oficial es relativo.

- Sudo apt-get update actualiza los paquetes de distribución de la base de datos disponibles para instalar. Este actualiza la base de datos, es decir la lista con los paquetes disponibles.
- Apt-get download no necesita sudo porque descargar puede hacerlo cualquier usuario.
- Sudo apt-get upgrade actualiza todos los paquetes, no la lista, sino los paquetes disponibles.
- El apt-get es para conectar con repositorios remotos. Para instalar un paquete ya descargado, se usa "sudo dpkg –i ruta_paquete". Para desinstalar, el mismo comando pero con "-u" de uninstall. Este es a bajo nivel.
    - El apt-get es propio de debían y sus distribuciones descendientes. En red-had será diferente. A bajo nivel serán lo mismo, pero a alto nivel los comandos cambiarán.

En Android:

El sistema Android, utiliza una aplicación que se llama vending.apk. Los paquetes son .apk. Esta aplicación es la que vulgarmente se conoce como google play. A bajo nivel probablemente use el dpkg.

Adicionalmente existen plataformas no oficiales, alternativas a google play. Si actualizo la ROM desde esos repositorios, puedo joder el móvil y no tendré derecho a quejarme.

En el .apk, las aplicaciones no serán en código binario. Serán scripts, las app son interpretadas. Android prácticamente es un calco de java con algún añadido.

El formato .apk es una variante del formato jar, que a su vez es una variante de un zip, al que le he cambiado la extensión. Si quiero abrir un rar, simplemente le cambio la extensión a zip y lo abro. Con un .apk igual, le cambias la extensión y lo puedes abrir.

Un jar y un APK son ejecutables, se les cambia la extensión para que al hacer doble clic, no se descomprima y se ejecute.

Un .apk tiene dentro:

- Un XML.
- Un archivo de clases.
- Un archivo donde estará el material binario si hemos pre compilado cosas en binario. En java, podemos hacer scripts que se interpretan línea a línea, pero también puedo hacer que parte de esos scripts se compilen en binario. Cuando se compila en binario eso es chino, no lo entiende ni dios.
- Los recursos no compilados, están en res.
- Otra carpeta con certificados Meta-INF.
- Después vamos a tener una carpeta, con código compilado para la CPU. Como Android es muy portable, tendrá en esas librería el código para todo tipo de procesadores q se pueda encontrar.
    - Arm de última generación
    - X86 de 32 bits.
    - Mips basados en mips.

Pregunta: Por qué podemos instalar cosas sin privilegios de root en un móvil y por qué no en un SO Linux? En ambos casos tengo el núcleo de Linux, y por encima una serie de demonios o aplicaciones (en caso de Linux), En caso de Android tenemos por encima instancias de la máquina virtual de Android, y por cada instancia tenemos las aplicaciones. En un sistema Linux, si quiero instalar una cosa nueva, se instala directamente como aplicación Linux, y esto necesita privilegios de SuperUsuario. Sin embargo, en Android, a no ser que instale algo sobre Linux no necesitare permisos, porque se instalará sobre la maquina dalvik (sobre la que corre el propio Android).

Ej.: Las extensiones de Chrome, no se diferencian mucho de las aplicaciones de Android. Cuando me descargo una extensión no tengo que hacer nada con superusuario, porque se instala en el navegador y no en el SO. Eso pasa porque el navegador Chrome es como un miniSO, andando sobre una máquina virtual. Tiene framework.

En Android cuando instalo cosas, se instala en un SO que está en una máquina virtual que está sobre Linux.

El tema de que Linux siempre te pida permisos de administrador para instalar cosas, es por la filosofía. Linux desciende de Unix, y eran servidores en los que solo el administrador puede instalar o desinstalar cosas y los usuarios lo usan sin más. En Windows es diferente porque todo va por módulos y le quiere dar más independencia a los usuarios.

Características de Windows.

Es software propietario, y lo vamos a tener para diferentes ediciones:

- Ediciones para dispositivos móviles
- Ediciones home
- Ediciones server
- Ediciones para profesionales.

Normalmente salen todas el mismo año, y todas tienen el mismo núcleo. Para versiones móviles el nucleo esta optimizado para arm en general y el resto optimizados para pc.

Los servidores tendrán un controlador de dominio. El controlador de dominio será la diferencia esencial que un Windows para particulares. Los servers tienen un poco de cosillas mas y el controlador. Por todo lo demás, es el mismo.

Entre una versión home y profesional, la diferencia es que la home esta capada, o suele estar capada y suele traer menos funciones que la profesional. Significa que es mejor? No, porque las cosas que trae una edición profesional es una cosa que normalmente no usara un usuario normal (copias de seguridad programadas, raid 5). La home para quien no use este tipo de cosas es mejor. Ademas, la profesional tiene muchos mas servicios funcionando y supone mas carga para la ram. Ahora bien, dentro de las versiones de home, hay varias.

Dentro de la home, la mejor es la ultimate, pero depende también para que lo quieres. Todo esto después repercute en el precio.

Otra característica de lo Windows son los grupos de trabajo y grupos de hogar. Antes de que apareciera Windows, solo existía la opción cliente servidor. Esa es la buena, los grupo de trabajo y grupos de hogar, para aulas y uso domestico esta bien, pero fuera de ahí es una chapuza.

Esto consiste en crear una conexión cliente servidor temporal. Hacen una conexión directa. El defecto que tienen es que cuando empieza a aumentar el numero de hosts, va mal. Se supone que el máximo es 20, pero realmente con 10 ya va mal. La única diferencia entre el grupo de trabajo y el grupo hogar, es que el de trabajo se creo antes. Requeria que todos los pc tuvieran el mismo nombre en el grupo de trabajo y q todos los pc estuvieran en la misma subred. En el grupo hogar, permite reproducir contenido multimedia por streamming y solo requiere que estén en la misma subred. A nivel profesional no tienen interés. El grupo hogar tiene la misma función que el grupo de trabajo.

Para la seguridad tendremos el Windows defender, que es malilla para detectar hardware, y el firewall, que es bastante bueno.

La compatibilidad, Windows esta diseñado para 32, 64 bits, y ara arquitecturas circulantes IA-32. Estos últimos, solo son para servidores (solo los servidores tienen soporte para ellos).

La configuración básica mediante panel de control.

La configuración avanzada se realiza a través de la consola MMC. La mayoría de las cosas que aparecen en panel de control, forman parte de la consola MMC. Una aplicación por ejemplo es la de "administración de equipos". Esa ventana es una instancia del MMC.

En ejecutar, escribimos mmc, y nos aparece la consola. Ahí se carga la consola de configuración y hay que buscar los archivos que serán nuestra herramienta. La consola es la que realmente se ejecuta, cuando hacemos clic en equipo – administrar, se carga la consola con la herramienta para poder hacer cambios.

El visor de eventos, nos permite hacer auditorias. Las auditorias son importantes para ver que esta pasando en el sistema.

El discmanager, solo carga el gestor de discos.

Cuando va a herramientas administrativas, se encuentras los accesos directos a las herramientas que se ejecutan a través de la consola.

La configuración del sistema es el msconfig, gestiona por ejemplo que servicios activar o desactivar o que cosas se ejecutan al incio, arranque, etc.

Directivas de seguridad local. Los clientes tb tienen directivas, mas simples, pero funcionan de la misma forma.

En las herramientas administrativas:

- Administrador de equipos.
- Configuracion del sistema → msconfig. Este gestor esta desde los primeros Windows.
    - Se puede seleccionar el inicio. Muchas veces, cuando salen pantallazos azules es porque un driver esta mal. Cargando el inicio con diagnostico, se cargan las cosas mas básicas y así te da la oportunidad de ver lo q esta pasando, si inicia Windows, puedes ver que controlador falla.
    - Arranque. Ahí esta la configuración del gestor de arranque. Si solo hay 1 SO hay solo un gestor de arranque, si hubiera mas de un SO habría mas de un gestor de arranque. Ahí se puede modificar el tiempo de espera que dura el gestor.
    - En servicios, están todos los servicios del SO. Gestionar los servicios desde esta pestaña no es buena idea. Para ello, lo mejor es ir a panel de control>herramienta administrativa>servicios. Una manera de acelerar el ordenador, es deshabilitar servicios que no sirvan, asi no ocupan memoria tontamente.
    - Inicio de Windows, contiene la lista de aplicaciones que se inician al principio. Hay una carpeta que se llama inicio, todo lo que se ponga en esa carpeta, se va a iniciar cuando inicies el ordenador. Las cosas que quieras que se inicien a nivel de usuario, se meten en esa carpeta. A nivel de maquina local, tenemos el inicio de Windows del msconfig.
    - Por ultimo, hay una lista de herramientas. Es una manera mas de acceder a ciertas herramientas.
- Programador de tares: sirve para ejecutar cosas de manera programada.

Nota: Si hay un menú multiple en Windows, saldrá un menú doble al estilo Linux, pero de Windows. Al instalar Linux después, en el MBR se sobreescribe el grub por encima del gestor de arranque de Windows. Entonces, el grub cargara primero el de Windows en caso de que se seleccione y si el de Linux después es multiple pues saldrán las demás opciones.

Los servers son gratis y van perfecto, pero si conectas un equipo a ese servidor, ya es ilegal. Deberias de pagar por cada equipo conectado y por cada conexión.

Gestion de usuarios.

No confundir, que un equipo tenga capacidad de gestionar varios usuarios a que el equipo sea multiusuario. Hoy en dia casi todos los sistemas operativos tienen capacidad de soportar mas de una cuenta de usuario (hay algunos capados como Android, pero en realidad lo soporta).

Por cada usuario real que controle un equipo, cada usuario tiene q tener una cuenta. Las cuentas tienen que estar separadas. Cuenta por usuario. Existe un registro donde se quedan guardadas las cosas hechas por un usuario, un log por usuario de sus acciones.

Para que un usuario se pueda conectar a un dominio en red local, necesita una cuenta de usuario. Hay que crear el usuario en el servidor para poder conectarse. Hay usuarios locales, que se crean en el equipo, y usuarios en red, que se crean realmente en el servidor, siendo locales al servidor y en red para los equipos desde los que se conecte.

En los móviles, tipo Android por ej, solo tiene sentido que haya una única cuenta de usuario activa. En un móvil, como minimo estará la cuenta de root que estará desactivada y la cuenta de usuario. Cuando rooteamos un móvil, lo que hace es instalar una aplicación q lo q hace es una especie de sudo por acción que ejecutemos.

Una cuenta de usuario puede estar asociada a uno o mas grupos. Esto se hace automáticamente. Si nos las arreglamos para que un usuario no este asociado a un grupo, malo. Por grupo, los usuarios tendrán unos privilegios minimos. En Windows, cuando creas un usuario, se te pregunta por 2 grupos (por la interfaz grafica), te pregunta si quiere q la cuenta sea de administrador o como un usuario limitado.

Un usuario por defecto en Windows será un usuario limitado. En Linux, antiguamente, cuando creabas un usuario tenias que al mismo tiempo crearle un grupo o menterle un grupo. Se puede hacer tb por modo grafico o por comandos. En modo grafico se crea un grupo con el mismo nombre que el usuario.

En Ubuntu últimamente, se hace que un usuario pueda pertenecer a varios grupos. Hay grupo principal y grupos secundarios.

Tanto los usuarios individuales como los grupos tendrán una serio de privilegios sobre objetos (equipo, impresora, políticas de seg).

En Linux, todos esta virtualizado y todas las cosas son archivos, asi que los privilegios recaen sobre esos archivos.

Cuando nosotros queremos registrar los permisos, no decimos nada sobre lo q puede o no puede hacer. Solo se crea el nombre y poco mas, su id, y poco mas. Los permisos se le dan a los objetos o archivos.

Cuando tenemos un archivo o usuario, y queremos hacer q un usuario no tenga acceso a un archivo, hay q coger el archivo y decirle que el usario tal no puede tocarlo.

Ej de boli: indico en el boli que es de tal usuario, y no le pongo al usuario todo lo q sea suyo.

Es importante que las cuentas están protegidas con contraseñas fuertes. Sobretodo en red. Cualquiera podría acceder con una conexión vpn por ejemplo a una cuenta en red si no tienes una contra fuerte.

Gestion de usuarios en Windows

La cuenta administrador siempre se va a crear. En los sistemas modernos estará desactivada y no va a tener contraseña. Siendo admin, tendrá acceso a todo, de cualquier persona.

La cuenta esta deshabilitada, no se debe usar. Cuando instalamos Windows, nos pide que creemos una cuenta. Esa cuenta es del grupo administrador, pero no es el administrador.

Las cuentas de administradores serán un grupo que se va allamar administradores y todos los usuarios que queramos que sean administradores los tengo que meter en este grupo. Yo puedo crear grupos nuevos y con los privilegios que quiera, pero por defecto esta esta.

Las cuentas de usuario deben de ser pocas, porque estos son los que gestionan el sistema y pueden acceder a los otros usuarios. El administrador de sistema debe ser solo 1. Cuando queremos nosotros instalar algo, se lo tenemos que pedir al administrador. El admin como cuenta que pertenece al grupo administradores es diferente al usuario administrador que esta deshabilitado.

Las cuentas limitadas o de usuario tiene bajos privilegios con respecto a la gestión del sistema.

Cuenta de invitado, en verdad para una empresa no tiene sentido, esto esta pensado para el uso domestico. Esta desactivada pero la puedes activar. Cuando se cierra la cuenta de invitado, se guarda toda la configuración que la persona en cuestión haya hecho. En esta cuenta no se pueden hacer acciones permanentes. Esta es la que tiene menos privilegios de todas.

La cuenta de sistema. Muchas veces van a ser invisibles. Son cuentas que usa el sistema y normalmente ni las veremos ni las gestionaremos. Una es “system”. Cuando el SO crea algo, lo crea a nombre de system. Todo lo que cree, se crea a nombre de un usuario. Una cosa es la propiedad y otra cosa es tener permiso. El propietario tiene pleno derecho sobre el archivo. Y otro puede tener ciertos permisos.

El administrador puede coger un archivo y apropiárselo, solo las cuentas con alto privilegio pueden hacerlo.

La propiedad de todos los archivos que se crean durante una instalación, se crean a nombre de system.

En Linux, todo lo que crea el usuario del SO se crean a nombre de root.

Seguridad local en Windows.

En Windows tenemos 3 opciones osbre los objeto. Vamos a tener una lista de acciones, después vamos a tener la opción de que se permita, la opción de que se prohíba o que no se le diga nada.

En el caso en que se prohíba o en el caso en que no se diga nada, no pueden hacer la acción. Solo la pueden hacer en caso de que tengan permiso. Tengo la lista de acciones, y sobre ellas voy dando permisos a usuarios o no.

Cuando dos permisos se solapan, si hay un grupo por ejemplo el de jefes, tiene permisos sobre la impresora porq pertenece al grupo jefe y permiso a la maquina de café porq pertenece a al grupo plantilla que tiene permisos para la maquina. Si en la plantilla digo q prohíba la impresora, anula la impresora porq en el grupo jefe estaba ignorado.

El permiso que prevalece, es el prohibitivo. Aunque en un grupo tengas una acción permitida, si la prohíbes con otro grupo y se contradicen, prevalece la prohibición. Esto es para Windows, y siempre en local.

Algunos privilegios están englobados en otros. Control total, engloba permisos de lectura, escritura, etc. Evidentemente si pongo un permiso que es de modificación, esta implícito el de lectura.

Permisos:

- Control total
- Modificar
- Lectura y ejecución
- Mostrar contenido
- Lectura

La lectura es el permiso mas simple, junto con la escritura. Existen permisos específicos. Los anteriores, se pueden desglosar en mas permisos (permisos mas concretos). Este sistema es complicado pero es muy versátil. En Linux es mas simple. En red se usa un sistema tipo Linux.

Si metes un grupo de altos privilegios en un grupo de privilegios limitados lo estas capando. Hay que intentar evitar poner prohibiciones, para que al solapar grupos, no se capen. Es mejor jugar con eso de dejar la acción sin nada.

La herramienta de prohibir realmente se debe de usar en casos muy concretos.

El propietario del objeto siempre tendrá la ultima palabra. El administrador puede usurpar cualquier cosa.

La herencia casi siempre esta activada,pero se puede desactivar. La herencia cuando esta activada, si creamos una carpeta dentro de otra, la nueva hereda todo lo de la padre.

Cuando cambiamos los permisos de un directorio, nos va a preguntar si hacemos que se transmitan esos permisos a todos los elementos contenidos en el directorio. Esto se conoce como dependencia. Si digo dentro de una carpeta q el usuario x no puede leer, todos los objetos “heredaran” ese permiso.

Mientras que la herencia y la dependencia no estén desactivadas, no voy a poder tocar los permisos de dentro de una carpeta, tendría q cambiarlo desde el directorio, y asi hasta el raíz.

La dependencia hacia los padres, y la herencia hacia los hijos.

La carpeta de usuario, es la que no tiene herencia de la carpeta raíz. El usuario limitado, no puede meterse en “C” ni en los archivos que hereden de C. El usuario administrador, si podrá trastear por ahí, y lo q se cree si que sigue heredando.

Caso concreto: Usuarios limitado, maria jose, y jose. Si maria jose da permiso a jose para que pueda entrar a su carpeta personal y jose crea un fichero ahí, y después maria jose le quita el control o permiso sobre su carpeta; jose no podría ver su fichero. Aunque fuera propietario. Si te quitan los permisos, aunque tengas plenos derechos sobre carpetas, no podras acceder a tu carpeta.

Caracteristicas de Linux.

Es gratuito, es una consecuencia de la licencia GNU. Tenemos diferentes versiones según las distribuciones. El nucleo de Linux, se usa en muchos dispositivos. En robotica, consolas, etc. Se suele usar en plan middleware, como Android, que es el ejemplo mas claro.

- Para móviles. Se usa un sistema middleware, xk da estabilidad.
- Distribuciones de escritorio. Ubuntu por ejemplo.
- Distribuciones profesionales.
- Para servidores. La RedHad por ejemplo (versiones servidor y cliente).

Vamos a tener un monton de aplicaciones asociadas al proyecto gnu Linux. Con licencias GNU tb. La única condición es que no le puedas cambiar la licencia.

Se suelen reconocer los sistemas Linux por los entornos de escritorio que tiene, tienen muchos; unos potentes y otros muy malos. Los mas famosos son GENOME (NOME) y KDE.

Tradicionalmente la configuración siempre se ha hecho en Linux a mano.

Gestion de usuarios en Linux

Tendremos grupos, y tendremos usuarios, vamos a tener un id único de usuario (uid) y un identificador único de grupo (gid). En Windows, es igual. En Windows es más opaco, es algo q el usuario no lo verá, mientras que en Linux casi siempre se trabaja con los identificadores. El uid y el gid es básicamente como una clave primaria de bdd.

En principio, en una distribución Linux (pureta) tendremos solo un grupo para cada usuario. En Windows podemos estar en mas de un grupo, en Linux antiguamente, solo perteneces a un grupo. Actualmente, han posibilitado que un usuario tenga otros grupos de forma secundaria. Tu tienes un grupo principal, y otros secundarios. El grupo principal del usuario, se crea automáticamente con el mismo nombre de usuario. Tu serías el único integrante de ese grupo, a no ser que agregues a mas gente. En local siempre habrá un grupo principal, aunq pertenezcas a grupos secundarios.

La cuenta root, en sistemas actuales, esta deshabilitada. Equivale a la cuenta administrador de Windows que crea las carpetas de system. Los usuarios, siguen una administración parecida a Windows, de tal forma que hay usuarios qe pertenecen al grupo administrador, y puede gestionar instalaciones y usuarios limitados.

Seguridad local en Linux.

El chmod, es quien cambia los permisos. Lo podemos usar de una forma simbolica u octal.

El ls de Linux, sale todo limitado, si quieres que aparezcan mas cosas, hay q ponerle parámetros. Ls –o te muestra los permisos de los ficheros, y ls –l te muestra además el grupo. Aquí, se marcan en los archivos que usuarios van a poder hacer cosas y que usuarios no.

Cuando hacemos un ls-o aparece la tupla de permisos. La tupla de permisos son 10 caracteres, que aparecen al principio de la lista. El carácter 0 indica el tipo de archivo. Los 1, 2, 3 simboliza los permisos del usuario propietario. El 4, 5, 6 simboliza el grupo principal de quien haya creado el objeto, y los tres últimos el resto de usuarios.

Aquí, los permisos son solo o concedidos o denegados y no hay solapamientos posibles, a diferencia de Windows que se solapan y hay 3 tipos de permisos sobre objetos.

Hay tres tipos de permisos. El primera carácter de cada trio de caracteres, representa la lectura, el segundo la escritura, y el tercer carácter la ejecución. En la tupla, aparecen con una r, w, x. Cuando están denegados aparece una -. Cuando los permisos recaen sobre un directorio, el significado es diferente. El de lectura de un directorio, lo que hace es leer el contenido del directorio, que no es mas que leer las entradas (enlaces duros) de ficheros. Cuando creamos cosas en el directorio, lo que estamos haciendo es escritura, porq estamos añadiendo enlaces (o moverlo, o eliminarlo). Toda la gestión, será de escritura. Y el permiso de ejecución es para abrir ficheros. Los directorios lo necesitan para poder abrirse.

Ej: drwxr----- → El usuario puede hacer de todo, y el grupo solo podrá hacerle un ls.

Ej: ejecutable en /bin (ls) → -rwxr_xr_x → pertenece a root.

Ej: fichero del disco duro /dev/sda2 → brwx------ → pertenece a root, y es el único que lo usa, porq solo lo usa el sistema. Tu nunca escribes en un fichero de discoduro.

Ej: el fichero de la salida estándar stdout → lrw-rw-rw pertenece a root, y es un enlace simbolico a 1 (que es la salida estándar).

Existen permisos speciales, que son los suid/sgid, equivalentes a los caracteres 3 y 6. En lugar de aparecer la x en esos caracteres, aparece la s. Estos permisos tienen lógica si creo un ejecutable y quiero que muchos usuarios lo usen. NOTA: SI LO PONES DE ESTA FORMA, CUALQUIER USUARIO PODRIA EJECUTARLO COMO SI FUERA PROPIETARIO, Y SI LO PONES EN EL GRUPO, CUALQUIER GRUPO PODRIA EJECUTARLO IGUAL.

En directorios tiene un significado diferente. Hace que cualquier archivo que se cree en ese directorio, pase a propiedad de quien creo la carpeta.

El Sticky, cuando en vez de una x tenemos una “t” en el ultimo trio de caracteres; en caso de tratarse de un fichero regular no significa nada. Cuando está en un directorio, lo que hace es proteger al directorio. Lo que hay dentro, los ficheros si los puedes modificar, pero las entradas del directorio no. No puedes borrar las entradas ni crear nuevas entradas a ficheros. Solo puede borrar o hacer cambios de ese tipo el propietario o el root. (El propietario de la carpeta).

Esto podría funcionar en dev, sobretodo para la salida y entrada estándar. Es probable que lo tenga, puedo que no interesa que se cambien las entradas del directorio. Los ficheros de dentro, no tienen por que tener todos los permisos, tendrá los q corresponda. La t sirve para que no te cambien las entradas del directorio y/o borren.

CHMOD:

Chmod expresión ruta_archivo

La expresión puede ponerse en octal o de forma simbolica. El chmod, cuando no recibe parámetros, te dice los permisos que tiene.

- En el octal, cogemos los tripletes, cuando veo, que en la tupla que quiero dejar hay un permiso concedido pongo un 1. Cuando esta denegado, se pone 0. Entonces, para el ejemplo de –rwxrwx—x → **chmod 771 ruta.**
- Simbolica, para hacer lo mismo sería ugo+x (al usuario, al grupo, y a otros) o bien a+x; O bien con +x. Si hago ug = rwx lo que hago es sobreescribir. El usuario, y grupo tendrían la tupla de esa forma.
    - Los especiales, de sticky y de guid o suid se usa u+s, u-s, g+s, g-s, sticky +t, -t.

Tema 4. Explotación de una RED INFORMATICA.

Un sistema en red, es cuando un equipo conserva su identidad. Cuando se conecta a un dominio, todavía existe individualidad. Cuando estamos en un sistema distribuido no hay esa individualidad. Una red de cajeros automáticos entre los bancos, todos trabajan como si fueran una piña, la bdd es única, pero van a estar en diferentes puntos de España. En un sistema distribuido las operaciones son transparentes por completo, comunicación perfecta y terminales sin identidad. Las transacciones se tienen que propagar de forma atómica.

En un dominio ls datos tienen una ubicación física. Los sistemas distribuidos no se consideran redes.

En una bdd distribuida, si se cae un servidor ni te enteras, mientras que en red, si se cae el servidor sabes cual es.

Redes peer-to-peer (p2p). Se hacen conexiones directas, se trata de coger un cable cruzado y conectas una tarjeta de red con otra, trabajando como si fuera una red. Ninguno de ellos tiene estatus de cliente o servidor. Este tipo de conexiones se llama ad hoc. Esto del cable seria a microescala.

Tenemos dos tipos de filosofías en red (redes). P2P o cliente-servidor. La filosofía de los mainfriend era cliente servidor, aunque en realidad, no eran cliente servidor. Estas filosofías se pueden aplicar a macroescala o a microescala.

A microescala, cliente-servidor, el servidor media y los clientes comparten los recursos. Mientras que en una conexión p2p, no se necesita mediador. La p2p, a microescala tb es conocida como ad hoc, que significa algo así como “para eso”, “especifico”. No es muy intuitivo, pero a lo q va es que la conexión no necesita una conexión. Los protocolos son diferentes.

En macroescala. El ejemplo perfecto es el del emule, en el emule, tendremos los servidores remotos (en listas muy grandes). Para compartir un archivo con otro tipo, primero tienen que pasar los paquetes por el router de mi casa, que si sale a internet se considera un servidor (a microescala no es así). Despues salta por el nodo de telefónica por ejemplo, y después un monton de nodos por lo que pasan los paquetes hasta llegar a la nube, donde todavía hay un huevo de nodos. Lo que tu envíes, por fuerza tiene que pasar por la puerta de enlace de tu compañía telefónica (por el ISP). La nube realmente es internet. Detrás de la nube estará el servidor al que te conectas. Esto funciona así:

- Vas saltando hasta llegar a tu servidor, el servidor te dice que otro cliente tiene lo que buscas, y ya haces una conexión con el cliente directamente. El servidor te provee la ip del tipo, entonces la conexión primero es cliente-servidor y después ya se establece la conexión p2p con el otro cliente. Todo lo que hay entre medias de tu casa hasta el servidor que te provee la ip del tipejo, se consideran nodos, aunque tb puedan ser servidores a microescala.

- Por otro lado, tb se pueden ir haciendo conexiones p2p, e ir conectándose a los diferentes nodos hata encontrar al tipejo. Eso es en una red LCAD. Cuando te conectas a esta red, de esta forma, los diferentes nodos tienen una lista de todos los archivos que tienen y comparten. Aquí no hay servidores, solo que cada cliente tiene una lista con todo lo que comparte y cada uno le pasa una lista con lo que tienen, se la pasan entre

si, entre los diferentes clientes conectados. Cuando se encuentra quien tiene el fichero que buscas, se establece la conexión definitiva para mandar información. La primera parte seria de consultas, pero la segunda, una vez encontrado el nodo, es que se fija la ruta.

- La diferencia, es quien contiene el enlace de quien tiene lo que quieres. Si eso lo tiene el servidor, estamos en una conexión de cliente-servidor (de emule por ej) y si esa lista no la tienen, la conexión es LCAD. El tema de que el servidor tenga las direcciones ip de quien provee de cosas, es mas peligroso, en las transferencia LCAD no se puede empapelar a tanta gente, lo q hacen es empapelar a los servidores.
- La forma mas eficiente es la que tenga mas usuarios.

Tipos de conexiones.

- PAN: conexión p2p, directa entre 2 dispositivos que están muy cerca. Alcance pequeño.
- LAN: alcance de habitación, de área local.
- CAN: para campus universitarios, el alcance es de ese tipo.
- MAN: parecido al can, pero como privado. Supongamos una empresa muy grande. Para la misma ciudad, diferentes ubicaciones.
- WAN: de área extensa, es internet. Internet puede evolucionar, pero nada lo sustituirá.
- WPAN: es una pan, pero wifi. Casi todas lo son.
- VPN: tiene dos tipos de filosofías, lo publico y lo privado. Intenta tener los beneficios que se tienen en una red privada, como usar impresora, compartir archivos fácilmente, hacer streamming. Estos protocolos privados no se podrían hacer en internet, porque en internet hay muchos peligros. Esto es un nuevo nivel de encapsulamiento, en el que se mandan los paquetes de una red lan para mandarlo a una red wan. Es como ponerle una protección de mas a algo muy sensible. Como se mueve por internet, se puede hacer a distancia. Aquí por ejemplo, para imprimir desde la clase, tengo que conectarme a la impresora de departamento. Por internet seria imposible. Para eso se crean conexiones vpn, para poder mandar los mismos mensajes que mandarias en una red local, para internet. Para eso hay q configurar el router de tu casa, para que cuando lleguen los paquetes vpn los acepte. No todos los routers pueden hacerlo. Cuando llegan los paquetes vpn, se le quita la cascara de protección y se convierte en paquetes Ethernet. Entonces los paquetes vpn, tendrán un “corazón” Ethernet , con cabeceras de internet.

Topologias:

- Conexiones dedicadas. Son topologías en malla, conexiones solo para algo concreto. Cuantos menos intermediarios, mas eficiente. Conexiones directas. Normalmente, las conexiones en malla, se dan a alta velocidad, tipo fibra óptica. Es muy robusto sobre caídas. A microescala es muy caro y no se usa, pero a macroescala si se usa, sobretodo en bdd distribuidas.
- Conexiones compartidas. Son las de bus y anillo, se tiran cables y te conectas al cable. Cuando formaba un anillo, se llama tb token ring. Lo bueno que tiene, es que si algún host se cae, no afecta. No hay caída, solo si el servidor cae se jode.
- En telaraña (estrella y árbol). Hay un servidor principla, serie de nodos que derivan la señal hsta llegar a los host. Esto es sensible a las caídas, si un nodo se cae, hay q investigar quien es, y si se cae, se cae parte de la rama. Hoy en dia se usa este tipo de topologías porq es la mas barata y se le puede aplicar seguridad.

Tipos de comunicación:

- Simplex. La fibra óptica solo trabaja asi. Tiene una conexión para enviar, o para recibir. 1 solo sentido.
- Half dúplex. La transmisión se realiza en 2 sentidos, pero no simultáneamente.
- Dúplex. En los dos sentidos y a la vez.
    - Este tipo de comunicaciones son para redes. Aunque también se dan dentro de un PC. El simplex, es la forma en que funciona la fibra óptica. La fibra óptica no puede enviar paquetes en ambas direcciones, tendremos un cable para la ida y otro para la vuelta. El half-duplex y el dúplex es el que se usa en el resto de redes, normales. El dúplex, es del wifi por ejemplo, y no se turnan la emisión y recepción. El cable de Ethernet (cable de cobre) half dúplex, se turnan emisión y recepción a veces.

Transmision síncrona. No confundir con el término referido a los ciclos de reloj. Aquí se refiere a la conexión. Cuando hay una conexión, hay una negociación al principio y se envían los paquetes, luego se espera a que haya confirmación. La negociación es algo como “oye, te envio paquetes. Cuantos?, 20, los aceptas? Si” y se establece. La síncrona es propia del protocolo tcp.

El protocolo udp, es asíncrono, este envía los paquetes sin destino claro. En el síncrono, se envían a un destino claro.

Transmision en paralelo: esto es parecido a lo del hardware del 1er trimeste. El wifi por ejemplo, no transmite por un único canal, usa varias frecuencias y todo al mismo tiempo. El paralelismo consiste en aumentar los canales de transmisión, pero esto a veces da lugar a la interferencias. Por eso hay veces que se transmite en serie.

Transmision en serie: se usa en cables Ethernet.

Componentes básicos.

En una red, encontraremos host (ETD o DTE). Los host serán servidores o terminales tontos, o clientes. Los terminales tontos son aquellos que no pueden funcionar de manera autónoma, necesita al servidor para funcionar (puede ser un pc por ejemplo sin SO que necesite al servidor). El cliente o el terminal autónomo, si se cae el servidor es capaz de funcionar, teniendo autonomía local o al menos un poco de autonomía local.

Medios de transmisión:

- Cable coaxial. Es muy antiguo pero se sigue usando. Ahora se usa en el ultimo tramo de fibra óptica. Esto es asi, porque la fibra óptica es muy delicada, entonces en las partes de manipulación se usa este cable.
- Cable biaxial. Es muy caro, pero muy rápido, incluso mas que la fibra óptica.
- Hilo telefónico. Se usa para cubrir grandes distancias para el adsl. No esta trenzado, transmite en varios canales, en paralelo a través de los subcanales; pero no hay varios hilos, tan solo el cable positivo y negativo.
- Par trenzado. Se usa para los cables Ethernet. Cuando haya cables, se necesita uno con voltaje positivo y otro negativo, para hacer la diferencia de potencial. Uno es la tierra, el 0.
- Fibra óptica. No necesita polo positivo ni negrativo, solo un cable, porq la luz no necesita polaridad ni voltaje para moverse. Ahora bien, necesita uno para emisión y otro para recepción, porq es simplex.

- Radiofrecuenta (ondas de radio o microondas). Las microondas son las que se usan, son mas energéticas y tienen mas ventajas que las ondas de radio. Las ondas de radio solo se usan para la radio. Las microondas
- Infrarrojos: cada vez mas en desuso, dan muchos problemas. Medio aéreo, son mas energéticos pero el problema de los infrarrojos es que interaccionan mucho con la materia. El microondas rebota y se propaga por toda la habitación, no son direccionales, mientras que los infrarrojos no rebotan, la materia los absorbe, entonces para que no sean absorbidos hay que lanzarlos directamente. Es uno de los principales problemas.
- Induccion magnética (nfc): Se crea un campo magnético y por sus oscilaciones se pueden transmitir también información. El problema de esto es qu es de muy corto alcance. Se usa para pagar en el autobús, o tarjetas de crédito para pagar, los móviles casi todos vienen con tecnología nfc.

Dispositivos de interconexión, nodos (ECD o DCE).

Son nodos que repiten la información, la reenvían, a veces cambiando las cabeceras de los paquetes y cosas asi. Casi siempre en realidad.

- El adaptador de red (la tarjeta de red, el nic, net interface card).
- Modem.
- Concentrador, conmutador
- Punto de acceso (etc.)

Medios de transmisión (no hay que sabérselo, si un poco la velocidad relativa).

- Cable infiniband es el biaxial. Se usa en sonido.
- Cable sin trenzar (rj-11) para hilos telefónicos.
- El de pares trenzados (rj-45) es el cable Ethernet. El trenzado mejora la señal. El hilo telefónico no necesita trenzado porq son solo 2 hilos, y las velocidades no son muy grandes. Para algancar la máxima velocidad el rj-45 necesita usar varias técnicas. El que estar mejor es el s/stp.
- Fibra óptica. La velocidad no es muy superior a la del rj-45. La fibra óptica como es inmune a las interferencias, Porq a la luz no le afecta nada que no sea otra luz, es mejor. En la fibra submarina para comunicar pahises, se usa el monomodo en la fibra óptica.
- El hibrido, combina una de las tecnologías anteriores con el fibra óptica. Pudiendo alcanzar el tope de 100 Mbps. Con el uso de proxy se le puede sacar un 20% mas de información. Estos proxy son proxy cache.
    - Un proxy es un servidor que esta entre 2 redes, puede integrar un firewall para filtrar. Un cache para almacenar los archivos descargados y favorecer la velocidad, etc.

La ventaja principal del infiniband, es de muy corto alcance. Los hilos en las conexiones Ethernet son simples,uno para cada cosa, pero la conexión es dúplex.

La fibra óptica: un tubito con un cristal en medio y una envoltura totalmente refrectante. Se pone un led o un laser, entonces se van emitiendo pulsos de luz y la luz va rebotando. Si las paredes son totlmente refrectantes y un material totalmente transparente, no se debería perder potencia, pero no existen esos materiales. Lo que realmente se hace es que se coge un material con mucha densidad hacia la envoltura y muy poca densidad en la parte interna, y la luz no va rebotando sino q se va curvando.

El monomodo el fibra óptica es mandar un único haz de luz, multimodo mandar mas de 1. En teoría los haces no interaccionan, pero cuando pasan un tramo muy largo, si el recorrido es muy largo al final si que acaban interaccionando.

Medios no guiados (sin cable):

- Los infrarrojos se usan para conexiones p2p en wpan.
- Las microondas. (AP punto acceso, en la transparencia aparece).

El bluetooth y el wifi tb. Si las cosas están lejos, invierten mas energía en aumentar el alcance. Si tenemos una conexión y la señal es fuerte, apenas se gastara energía, si la conexión e sdebil se gastara mucha energía. El bluetooth en modo bajo consumo consume muy poco. El wifi, lo máximo que tiene es de 20 metros (espeficicación, puede que tenga más en realidad). Esos 20 metros se amplian con repetidores. Vamos a tener 3 tipos de redes configurables en wifi.

- Ad-hoc (p2p). Com con cable pero sin el. En el móvil se llama wifi direct.
- BSS. Infraestructura de servicio básico. Aquí se coloca un punto de acceso y todos se colocan a el. Un punto de acceso es aquel aparato con capacidad de crear una red en infrastructura (el router tiene esa capacidad).
- ESS infraestructura de servicio extendido. Con muchos puntos de acceso. Los puntos de acceso de pueden configurar en modo repetidor.

Tenemos también el municilap wi-fi que es para redes wifi un poco más extendidas, para municipios.

Redes de datos basadas en 3g. Para eso necesitamos un adaptador 3g. Tendremos tarjetas de red para wifi o para 3g, o las 2 cosas.

Las que están basadas en 4g son mas rapidas, pero son mas de lo mismo.

Las redes de inducción magnética (NFC). Es muy antiguo. Ha estado dando vueltas mucho tiempo hasta que Android lo ha rescatado. Android lo ha perfeccionado. La velocidad de transmisión no es muy alta. Con estas cosas se suele transferir datos de identificación, no archivos.

Dispositivos de interconexión.

El adaptador de red es un hardware que transforma los datos del ordenador en datos de red. En su etapa más baja (Se ve en el modelo tpc/ip). Las tarjetas de red suelen incluir su propio firmware pxw (para el arranque en red).

Cada adaptador de red tiene un identificador único al que llamamos MAC. También se le llama identificación física. Cada tarjeta tiene un identificador, que es la mac, único e unívoco. No se puede modificar, pero se puede enmascarar. Las MAC están siendo sustituidas por identificadores únicos de hardware (uids o guids, mirar anteriormente).

Cada vez que se envía un paquete, se manda información y en esa información refleja la mac que la manda. La mac no se puede cambiar, pero la información de esos paquetes sí. Eso es el enmascaramiento.

Aparte de la mac se envía un identificador lógico, que es la IP. IP es protocolo de internet. En una red, no puede repetirse, y la red mas grande es internet (que no se pueden). Con la v4 ya nos quedamos cortos, pero con la v6 nunca nos quedaríamos cortos. La ip 4 usa 4 octetos, y la v6 8 octetos (para la 4 32 bits y para la 6 son 128 bits).

Los módems. Los datos cuando se transmiten por el medio telefónico, normalmente la señal, no va a ser puramente digital, va a ser un poco analógica. Se va a transmitir como una onda. El modem tiene un componente analógico. El modem, va a ser el hardware que va a cambiar de una naturaleza puramente digital a "analógico" (no es puramente analógico). Lo qu hace realmente es, coger una señal analógica, y dentro se pone la señal digital (eso es modular). Hacer lo contrario, es desmodular (de hay viene el nombre módems.).

Tipos de modulación digital:

- Modulacion de amplitud (ask). Cuando se suma la señal portadora (analógica) y la señal moduladora te da la señal modulada ask. Se trata de, coger la onda portadora, y cuando en la señal moduladora haya un 1 capturarla y cuando no, no, porq la moduladora es digital y es 0 o 1. Es poco resistente al ruido y las interferencias. Es poco costoso, se usa sobretodo en cable coaxial o fibra óptica.
- Modulacion de frecuencia (FSK). Al sumar la señal portadora y la moduladora, no se corta realmente la señal, se mantiene la analógica solo que se superponen y en ciertos puntos se ha capurado y en otros no. Necesita un gran ancho de banda. No le afecta el ruido. Esto es lo qu se usa en el hilo telefónico.
    - En ambas se suman las ondas moduladora y portadora, pero en una se modifica la amplitud de onda y en otra se modifica la frecuencia.
- Modulacion de fase (PSK). Este es resistente a errores, pero es como un termino medio. Es fácil de hacer, poco costoso y este es el que se usa en las wifi y 3g.

*** En Ethernet, la señal está desmodulada.

ADSL2+: es la que se usa ahora. Es asimétrico, porque los subcanales que se usan son muchos para descarga y solo 2 o 3 para la subida. La velocidad siempre es la misma, pero se multiplican los subcanales.

El VDSL: llevan también televisión.

- RTC usaba solo una señal analógica, por eso cuando llamabas se caia el internet.
- ADSL uno para telefonía, unos pocos para subida y los demás para bajada.

Para fibra óptica, tenemos el DOCSIS, que es el cable modem, el de ono. Es el que lleva VoIP + datos + tv.

Un router, router, propiamente dicho no da internet, solo enruta. Para modularizar señales se necesita el modem, hoy en dia te dan un router con funciones de modem. Para tener internet realmente se necesita un modem.

Un concentrador (hub) es un nodo que lo encontraremos en topologías de estrella y árbol. No tiene sentido en topologías de maya o bus. No tiene capacidad de multiplexion (cualquier elemento eléctrico o electrónico capaz de separar flujos mezclados). La información se reenvia. Hay unos concentradores que son activos, que lo que hace es coger la señal y la amplifica. Pero aun así los hub no son muy eficientes. Los pasivos transmiten la señal sin amplificarlas.

El hub en principio no hace nada, solo empalmar cables, y de uno sacad 2, 3, 4 etc. Lo que haga el hub dependerá siempre del elemento mas lento, si hay un elemento muy lento, la red irá a la velocidad de ese elemnto.

Cada vez que conectamos un host a un concentrador la velocidad de transmisión decrece drásticamente. Si conectamos muchos hub anidados (los hub no suelen tener más de 6 derivaciones u 8 como mucho) la velocidad decrecerá una burrada. Este tipo de elementos no se usan, ni se venden. Hoy en día lo que se usa son conmutadores. Lo de amplificar la señal, no afecta a la velocidad, afecta a la distancia.

El pc que emite la señal, manda un paquete a diferentes host conectados a un hub. Lo que sucede es que el pc, manda todos los paquetes. Esos paquetes llegan al hub, y el hub los reenvia a todos los nodos, sin diferenciar al destinatario. Después el destinatario cogerá lo que quería. En una red local, problemas de seguridad no va a haber. Lo que va a pasar esq tantos paquetes dando vueltas no es eficiente.

Un conmutador, o switch, sigue otra filosofía. Es parecido a lo que hace el hub pero aquí si hay multiplexion, se reparten los paquetes. Por cada cable solo irán los paquetes que tienen que ir. Como se las ingenia el switch para hacer esto? Por la mac. Cuando conectas a un equipo, hay una negociación y se identifican a los dispositivos por la mac. Dependiendo de la mac que lleve el paquete, lo enviará por un sitio o por otro. El paquete llevará la mac de origen y de destino. Esto alivia muchísimo el tráfico. Es lo que normalmente se utiliza. Va a usar siempre un modo half-duplex o fullDuplex. Aquí si hay una negociación, tb es capaz de discriminar velocidades y el lento ya no jode la red (dependiendo del switch).

Existen switch capaces de hacer redes virtuales. Estos switch se les llama de capa 3. Una red virtual lo que hace es coger varias subredes y mediante unos protocolos monta una subred ficticia (montar una subred lógica, independiente a la subred física, con host muy diferentes).

Un punto de acceso es el equivalente al switch pero para redes inalámbricas. Hay equivalentes del hub en wifi? No hace falta, porq el aire ya lo transmite todo a la vez. Una diferencia que tiene con los switch, es que el ap tiene una ip para gestionarlo. Aunque hay switch que también tienen ip.

Los ap tienen 3 modos:

- Modo ap: modo por excelencia. Funciona como un switch, se le conecta un cable, y crea diferentes conexiones para diferentes host inalámbricos.
- Modo cliente (infraestructura). Lo usaremos para convertir una tarjeta de red, no inalámbrica en inalámbrica. Se trata de conectar una tarjeta de red no inalámbrica, a un ap en modo infraestrucura, que se conecta a otro nodo ap. El ap que funciona en modo infraestructura, no tiene tarjeta de red, se tiene que enganchar forzosamente a una tarjeta de red.
- Modo cliente ad-hoc. Es muy parecido al anterior, pero en vez de conectarse a otro ap en modo ap, se conecta a cualquier dispositivo electrónico en modo ad-hoc (otro dispositivo con bluetooth, a otro ap modo cliente, a otro nic wifi, etc (cualquier dispositivo capaz de hacer conexiones p2p)).

- Modo puente (bridge). Vamos a conectar inalámbricamente dos segmentos de una misma subred. Tendremos un ap principal, y conforme aumenta la distancia disminuye su potencia o conectividad, pues lo q se hace es poner otro ap en modo puente. La primera se llama red 1, y la segunda red2. Los segmentos solo se aplican a cosas inalámbricas. La red tiene un identificador, pues dependiendo del segmento tendrán un identificador u otro (normalmente mismo nombre con un subfijo), pero forman parte de la misma subred. Esto quiere decir que el rango de ip pertenecerá a la misma dirección de red. Entonces, te conectes a la que te conectes estaras en la misma subred. El que funciona en modo bridge es una extensión del primer punto de acceso, por tanto, sigue funcionando como punto de acceso.
- El modo repetidor es parecido al modo bridge pero no se diferencia la segunda red de la primera. Los host verán la misma señal.

DIFERENCIA ENTRE MODO REPETIDOR Y BRIDGE: EL MODO BRIDGE, SUSTITUYE LAS CABECERAS DE QUIEN SE LO MANDA POR LAS SUYAS PROPIAS. EN MODO REPETIDOR, NO SE SUSTITUYE LAS CABECERAS, SE VUELVE A ENCAPSULAR POR CADA SALTO EN EL REPETIDOR. A LO LARGO DE LA RED, EN LOS REPETIDORES LAS TRAMAS SE VAN HACIENDO DEMASIADO LARGAS. EL MODO BRIDGE VUELVE A IMPLEMENTAR LOS PROTOCOLOS WIFI PARA AÑADIR LAS CABECERAS DE ESE NUEVO PUNTO DE ACCESO.

La puerta de enlace (Gateway). Se diferencia del router, en que el router conecta dos redes diferentes de la misma naturaleza, mientras que un Gateway conecta dos redes de diferente naturaleza (una de protocolo Ethernet a internet por ejemplo). El Gateway también hace enrutamiento, convertirá direcciones locales en externas. El Gateway traduce lo que le llega, de protocolos externos a protocolos internos. Una vez hecho el cambio, enruta hacia donde tiene que ir.

Normalmente integran la función de modem. Un modem sin Gateway no tiene mucho sentido. Aunque en empresas, puede que este por separado. Pero en pymes y cositas modestas, todo esta integrado en el mismo aparato.

Cuando hacemos configuraciones en la red, nos pide la dirección de la puerta de enlace. La puerta de enlace, es la ip del cacharro que nos va a llevar a la wan. Cuando tengo que acceder a la del Gateway, suponiendo que tenga una red en cascada, deberé poner la dirección del router que me gestiona a mi (entendiendo router como router únicamente, sin las demás funciones).