



# **44. CONCEPTO, EVOLUCIÓN Y TENDENCIAS DE LOS SISTEMAS OPERATIVOS. SISTEMA OPERATIVO UNIX- LINUX. SISTEMA OPERATIVO WINDOWS.**

## **Tema 44. Concepto, evolución y tendencias de los sistemas operativos. Sistema operativo UNIX-LINUX. Sistema operativo Windows**

---

### **ÍNDICE**

<b>44.1.- Sistema Operativo.....</b>	<b>1</b>
44.1.1 Concepto.....	1
44.1.2 Evolución de los Sistemas Operativos.....	2
44.1.3 Clasificaciones de los Sistemas Operativos.....	4
44.1.4 Estructura de los Sistemas Operativos.....	5
44.1.5 Funciones de un Sistema Operativo.....	5
44.1.6 Tendencias.....	6
<b>44.2.- Sistema Operativo UNIX-LINUX.....</b>	<b>7</b>
44.2.1 Características.....	7
44.2.2 Arquitectura de Unix-Linux.....	8
44.2.3 Gestión de procesos.....	8
44.2.4 Gestión de memoria.....	9
44.2.5 Gestión de E/S.....	9
44.2.6 Gestión de archivos.....	10
<b>44.3.- Sistema Operativo Windows.....</b>	<b>11</b>
44.3.1 Características.....	12
44.3.2 Arquitectura de Windows.....	12
44.3.3 Gestión de procesos.....	13
44.3.4 Gestión de memoria.....	14
44.3.5 Gestión de E/S.....	14
44.3.6 Gestión de archivos.....	15
44.3.7 Seguridad.....	16
<b>44.4.- Bibliografía.....</b>	<b>16</b>

#### **44.1.- SISTEMA OPERATIVO.**

##### **44.1.1 Concepto**

Se puede definir un sistema operativo (S.O.) como un conjunto de programas que controlan directamente los recursos hardware (HW) o físicos de un ordenador (CPU, memoria principal y periféricos) proporcionando una máquina virtual que oculta los detalles físicos de la máquina y ofrece al hombre un entorno más amigable. El sistema operativo es la capa de software más baja de un ordenador. Cada capa oculta a las capas superiores ciertos detalles de las capas inferiores. De esta forma se construye el software, basándonos en lo que ya existe

El sistema operativo tiene una serie de funciones, que se pueden agrupar en 3:

1. Inicializar la máquina. Prepara la máquina para el funcionamiento. Hay 2 formas de inicialización:

- a) TOTAL: Inicialización de todas las funciones y servicios que la máquina puede ofrecer. Por ejemplo, MS-DOS tiene inicialización total.
- b) PARCIALMENTE: Se va a ser selectivo con los tipos de servicios que se inician. Por ejemplo, Linux y Windows.

La principal ventaja/utilidad de la inicialización parcial es la recuperación de la máquina ante fallos: consiste en que si falla un servicio de la máquina no hace falta apagar toda la máquina, sólo hay que lanzar de nuevo el servicio.

2. Servir de máquina virtual. Se ocultan detalles de hardware proporcionando un entorno más amigable. Esto tiene 2 objetivos:

- a) La SEGURIDAD: En lugar de que el usuario acceda directamente a un recurso HW, lo hace el S.O. para que no se produzcan operaciones no

deseadas, también llamadas operaciones peligrosas: operaciones de entrada/salida (E/S), operaciones de acceso a memoria.

El HW tiene dos formas de actuar: modo supervisor y modo usuario.

Todos los programas actuarán en modo usuario hasta el momento en que haya que acceder al HW; será entonces cuando cambia a modo supervisor para evitar las operaciones que puedan causar problemas. Se generará una interrupción al realizarse una de estas operaciones susceptibles de fallo. Esa interrupción la coge el sistema operativo y actúa en consecuencia. El sistema operativo tomará el control del hardware y realizará la operación que se le indica.

Una interrupción es una señal física que generan los dispositivos del sistema y que la trata el sistema operativo.

Al conjunto de interrupciones se le llama **interface interna** del sistema operativo.

- b) **ABSTRACCIÓN**: Se abstraen las características físicas y reales de la máquina ofreciendo una serie de servicios incluso mayores de los que puede ofrecer la propia máquina. Por ejemplo para trabajar con ficheros se utilizan nombres, pero el ordenador no utiliza esos nombres para referirse a ellos; emplea una dirección.

La E/S: cuando tecleamos un dato lo vemos en pantalla tal como lo hemos imaginado, aunque realmente para la máquina son unos y ceros.

Esto constituye la **interface externa** del S.O., el lenguaje con el que nos vamos a comunicar con él. Se denomina SHELL.

Por tanto, tenemos 2 tipos de interface:

- La externa: forma de comunicación entre nosotros y el ordenador; a través de comandos/órdenes (abstracción).

- La interna: forma de comunicación del sistema operativo con el hardware (modo supervisor).

3. Administrar los recursos para su funcionamiento. Esta administración tiene que cumplir 3 características:

- tiene que ser CORRECTA: si hay 2 procesos que quieren acceder a un recurso, hay que darle acceso primero a uno y luego a otro, pero no mezclarlos.
- tiene que ser JUSTA: si tenemos 2 procesos hay que darle salida a los dos; un proceso no puede monopolizar.
- tiene que ser EFICIENTE: para mejorar el rendimiento del sistema.

Por último decir que un S.O. tiene que tener estas 2 características:

- DETERMINISMO: si se repite la misma operación con los mismos datos de entrada debemos obtener los mismos resultados.
- INDETERMINISMO: en el sentido de que tiene que responder oportunamente a las interrupciones, es decir, no sabe qué interrupción va a llegar primero, no conoce el orden pero debe saber tratarlas.

#### ***44.1.2 Evolución de los Sistemas Operativos***

Los sistemas operativos han evolucionado en paralelo al desarrollo del HW. Conforme el HW iba incorporando nuevas capacidades, los S.O. debían adaptarse para permitir gestionar eficientemente esas nuevas capacidades. La evolución de S.O. puede organizarse en generaciones con algunas características comunes:

##### **Primera Generación**

Abarca desde 1945 a 1955. Se caracteriza porque no existía un sistema operativo. Eran los propios diseñadores de las máquinas los que las programaban a través de cableado, y las manejaban.

## **Segunda Generación**

Desde 1955 a 1965. En el campo del hardware aparece el transistor. Se empiezan a utilizar las tarjetas perforadas. Se distinguen 2 tipos de sistema operativo:

- Monitor Residente: el S.O. se limitaba a cargar los programas a memoria, leyéndolos de tarjetas perforadas, y ejecutarlos. El problema era encontrar una forma de optimizar el tiempo entre la retirada de un trabajo y el montaje del siguiente.
- Trabajo por lotes: se utilizan las cintas magnéticas. Como solución para optimizar el tiempo de montaje surgió la idea de agrupar los trabajos en lotes, en una misma cinta o conjunto de tarjetas, de forma que se ejecutaran uno a continuación de otro sin perder apenas tiempo en la transición. En la memoria del ordenador existen dos ítems: a) el monitor de lotes: indicando qué trabajo se está ejecutando y b) el trabajo actual.

Estos dos tipos de sistema operativo se caracterizan por:

- No existe ningún planificador (el que decide qué trabajo se va a realizar): la razón de su inexistencia es que no es necesario, ya que sólo hay un trabajo en memoria principal
- Tampoco existe un reloj (mide el tiempo que un trabajo está ocupando la CPU): no es necesario porque sólo hay un proceso en memoria ejecutándose.

## **Tercera Generación**

Abarca desde 1965 a 1980. En el campo del hardware tenemos los circuitos integrados con tecnología LSI y VLSI. Antes de ver los distintos S.O., merece la pena dejar claros dos conceptos: Programa: código (algo estático) y Proceso: programa en ejecución (algo dinámico).



- S.O. Multiprogramación. En memoria principal va a haber más de un programa. La CPU ejecuta instrucciones de un programa, cuando el que se encuentra en ejecución realiza una operación de E/S, en lugar de esperar a que termine la operación de E/S, se pasa a ejecutar otro programa. De esta forma es posible, teniendo almacenado un conjunto adecuado de tareas en cada momento, utilizar de manera óptima los recursos disponibles.

Se va a definir el grado de multiprogramación como el número de programas que hay actualmente en memoria

OverHead es un parámetro que mide la diferencia de tiempo que hay entre que el sistema operativo está dedicado a hacer sus tareas y el tiempo dedicado al cambio de contexto entre procesos.

- S.O. Multiproceso. Tiene varios procesos en memoria principal. Hay que distinguir que:
  - o *Multiprogramación IMPLICA Multiproceso.* Multiprogramación: varios programas en memoria principal. Al estar en memoria principal se están ejecutando. Programas en ejecución = procesos.
  - o *Multiproceso NO IMPLICA Multiprogramación* Ahora bien, podemos tener únicamente un programa en memoria y este programa lanzar varios procesos. Un programa puede querer imprimir, leer algo por teclado, ... lanzar procesos. Varios procesos = multiproceso
- S.O. Multiprocesador. Se utilizan donde hay 2 CPU's o más.
- S.O. Interactivos. Sistemas que de alguna forma mantienen un diálogo con el usuario; mediante el SHELL (lenguaje de comandos).

Es muy importante el tiempo de respuesta: tiempo que transcurre entre que el usuario manda hacer algo al sistema hasta que obtiene la respuesta. Siempre se intenta minimizar el tiempo de respuesta.

- S.O. Multiusuario. Aquellos sistemas operativos en los cuales varios usuarios pueden acceder al mismo ordenador simultáneamente.

Por ejemplo: cualquier sistema UNIX o LINUX. Se puede tener una máquina y generar usuarios accediendo a esa máquina (a través de red local, Internet, ...).

- S.O. de Tiempo Compartido. Pretenden dotar a cada persona de una parte de la CPU. El usuario ve el ordenador como suyo propio, aunque no lo es.
- S.O. de Tiempo real. Estos sistemas se usan en entornos donde se deben aceptar y procesar en tiempos muy definidos un gran número de sucesos, en su mayoría externos al ordenador. Si el sistema no respeta las restricciones de tiempo en las que las operaciones deben entregar su resultado se dice que ha fallado.

### **Cuarta Generación**

Abarca desde 1980 hasta nuestros días, está marcada por los ordenadores personales.

Sistema Operativo de Red. El usuario es consciente de que existen otras máquinas. El usuario ya no quiere trabajar solo; quiere trabajar con otros usuarios. Tiene que acceder de forma explícita a esas máquinas.

Sistema Operativo Distribuido. El parámetro clave es la transparencia: el usuario no es consciente de que existen otras máquinas; no sabe en qué máquina está.

#### **44.1.3 Clasificaciones de los Sistemas Operativos**

- Según el número de usuarios:
  - o S.O. monousuarios: sólo aceptan un usuario en un momento determinado.
  - o S.O. multiusuarios: aceptan simultáneamente a más de un usuario.



- Según el hardware:
  - o Según el número de CPU'S:
    - ▢ S.O. monoprocesador: sólo controla una CPU.
    - ▢ S.O. multiprocesador: varios procesadores (más complicado de diseñar).
  - o Según la organización de la memoria principal:
    - ▢ S.O. centralizados: una memoria principal y los procesadores van a estar intentando acceder a esta memoria principal. Los procesos se comunican a través de la memoria.
    - ▢ S.O. distribuidos: cada procesador tiene su propia memoria principal. Los procesos tienen otros mecanismos de comunicación.
- Según el modo de trabajo con los sistemas operativos:
  - o Interactivo (on-line): el usuario dialoga con la máquina.
  - o Batch (off-line): no hay comunicación con la máquina cuando está realizando el trabajo.
- Según el objetivo para el que fueron diseñados:
  - o S.O. de propósito general: capaces de realizar cualquier tarea.
  - o S.O. de propósito específico: sólo pueden realizar una tarea específica, se instalan en microprocesadores que controlan el funcionamiento de electrodomésticos, vehículos, equipos de electrónica consumo, etc.
  - o S.O. de Tiempo Real: ofrecen una respuesta en un intervalo de tiempo bien definido.
  - o S.O. Virtuales: corren sobre HW de un ordenador ofreciendo a los niveles superiores copias exactas de la máquina real, de forma que en cada copia se puede ejecutar un sistema operativo distinto.

- o S.O. de dispositivos móviles: deben adaptarse a las limitaciones que estos dispositivos presentan: procesadores lentos, memoria limitada, pantallas pequeñas y consumo de energía limitado. Ejemplos típicos para estos dispositivos son iOS de Apple, Windows Mobile, Android, etc.

#### **44.1.4 Estructura de los Sistemas Operativos**

##### Sistemas operativos monolíticos

Estos S.O. no tienen una estructura definida. El S.O. se escribe como una colección de procedimientos entrelazados de tal forma que cada uno puede llamar a cualquier otro. Las características de este tipo de estructura son:

- Construcción del S.O. basado en procedimientos compilados separadamente que se unen en un solo fichero objeto a través del enlazador (linker).
- Buena definición de parámetros de enlace entre los distintos procedimientos existentes, lo que genera acoplamiento.
- Carece de protección al entrar a procedimientos que gestionan diferentes aspectos de los recursos del ordenador, como almacenamiento, E/S, etc.
- Son hechos a medida lo que tiene como ventaja que son eficientes y rápidos y como desventaja que carecen de flexibilidad para crecer.

##### Sistemas operativos con capas

El SO se organiza en una jerarquía de estratos, estando construido cada uno de ellos sobre el otro que tiene menor jerarquía que él. Ejemplos: THE, MULTICS.

##### Sistemas operativos Cliente-Servidor.

Minimizar el kernel (núcleo) del S.O., desplazando el código de todos sus servicios a estratos lo más superiores posibles. Para ello, la mayoría de sus funciones se implementan como procesos de usuario, denominados

procesos servidores, de forma que cuando un proceso de usuario llamado proceso cliente, necesita un servicio del S.O. lo que hace es enviar un mensaje al proceso servidor correspondiente, que realiza el trabajo y devuelve la respuesta.

#### ***44.1.5 Funciones de un Sistema Operativo***

Las principales funciones que tienen los S.O. son la gestión de procesos, la gestión de la memoria principal, la gestión del almacenamiento secundario y la gestión de los dispositivos de entrada/salida.

##### **Gestión de procesos**

La CPU es el recurso principal del ordenador de modo que es necesaria una gestión eficiente de la misma para garantizar su aprovechamiento.

El S.O. ha de cargar los distintos procesos, iniciarlos, supervisar su ejecución llevando a cabo los cambios de contexto necesarios y detectar su terminación normal o anormal. En los entornos multiusuario es fundamental la activación de mecanismos de protección que limiten las posibilidades de acceso de cada proceso a una serie de recursos para los que cuente con la debida autorización.

##### **Gestión de memoria principal**

En un sistema multiproceso los procesos tienen que compartir la CPU teniendo que encontrarse en memoria principal para poder pasar a ejecutarse inmediatamente; así, varios procesos tienen que compartir la memoria principal sin que unos puedan acceder a los recursos de otros.

Para ello hay que dividir la memoria en bloques y éstos se les asignaran a distintos procesos. Para hacer la división se utiliza la segmentación, la paginación o la segmentación paginada.

##### **Gestión de los sistemas de archivos**

En un sistema de archivos el S.O. tiene que hacerse cargo de: la gestión del espacio libre/ocupado, de los cachés de lectura y escritura, del vínculo entre nombres y archivos, de las asociaciones entre los bloques físicos de

los dispositivos y los bloques lógicos, de los permisos para el acceso y modificación a los distintos elementos.

### **Gestión de entrada/salida (E/S)**

La velocidad con que se comunican el procesador y la memoria principal contrasta con la velocidad cuando los programas deben interactuar con algún tipo de dispositivo de E/S; durante este proceso la ejecución del programa se ve interrumpida ya que la comunicación con dichos dispositivos es significativamente más lenta que con la memoria.

Conjuntamente con la multiprogramación surgen dos conceptos: el acceso directo a memoria (DMA) y las interrupciones. El procesador cede el control de la E/S a un módulo que se encarga de ejecutar este tipo de operaciones (el controlador de DMA) y esperar hasta que éstas se completen y cuando esto suceda avisar al procesador (que se encuentra mientras tanto ejecutando otras instrucciones -bien sea del mismo proceso o de algún otro-) que puede continuar con las operaciones subsiguientes que quedaron pendientes cuando se realizó la petición de E/S mediante una interrupción.

#### ***44.1.6 Tendencias***

Actualmente, la movilidad es lo primordial en nuestra sociedad, y asociado a esta movilidad está la seguridad. Se empieza a hablar de sistemas operativos en la nube (cloud computing) y se consolidan los sistemas operativos de los dispositivos móviles.

Otro aspecto a destacar derivado del momento económico es el ahorro de costes. Fruto de esto se puede ver una tendencia más que clara en la virtualización.

### **S.O. en dispositivos móviles.**

Si existe una carrera hoy en día en el desarrollo de S.O., ésta está en los S.O. para dispositivos móviles. Los nuevos sistemas operativos convierten al teléfono en un completo aparato multimedia. Hasta hace muy poco

tiempo la elección de un móvil venía determinada por sus características físicas: recepción de la señal, cámara... pero con la llegada de los smartphone, la elección del S.O. se ha convertido en algo muy importante.

A diferencia del mundo del ordenador personal, y debido quizás a su juventud en el mercado, no existe un dominador claro de S.O. móviles. Hay fabricantes hardware que son los fabricantes de sus propios sistemas operativos como por ejemplo Apple que sólo distribuye el S.O. iOS para iPhones e iPad, lo mismo que RIM que distribuye su S.O. BlackBerry OS en dispositivos BlackBerry, etc. En el otro extremo están los fabricantes que utilizan S.O. de otras compañías como Android, Symbian, Windows Mobile...

Las compañías que fabrican y distribuyen su propio S.O. tienen a favor que las actualizaciones de los dispositivos son muy controladas.

Otra tendencia vendrá de la mano del despegue de los dispositivos tablets (tabletas), teniendo en cuenta los anuncios de lanzamientos de tabletas: Apple (iPad 2), RIM (PlayBook), Samsung (Galaxy Tab 2), etc. Esto dará aun más auge a los S.O. móviles.

### **S.O. en la nube o en red.**

Estos S.O. surgen del concepto de Computación en la Nube (Cloud Computing) que es un nuevo paradigma que básicamente permite tener servicios computacionales a través de Internet.

Una de las grandes ventajas que se pueden encontrar a este nuevo paradigma es la baja inversión a realizar en HW ya que toda la infraestructura de la computación en la nube se encuentra en los grandes proveedores de servicios de Internet. Bastaría con un hardware mínimo, un navegador y una buena conexión a Internet.

Otra de las ventajas sería que las aplicaciones no se instalan en el pc, son aplicaciones Web lo que hace que sea compatible con la mayoría de los formatos conocidos.

Permite tener una única copia de un fichero disponible en cualquier lugar y momento.

Sus puntos débiles son la seguridad y la necesidad de una conexión a Internet.

Estos S.O. son una buena opción para los notebook que tienen poco hardware, e incluso podrían hacer que los fabricantes apostasen por modelos más baratos que permitiría difundir mucho más la informática.

Entre los S.O. más importantes destacan: eyeOS, ChromeOS de Google, oOS, iCloud, etc.

#### **44.2.- SISTEMA OPERATIVO UNIX-LINUX**

UNIX es un sistema operativo creado en 1969 por un grupo de investigadores de los laboratorios Bell de AT&T, entre ellos Ken Thompson, Dennis Ritchie y Douglas McIlroy, como una versión reducida del proyecto MULTICS; primero fue escrito en ensamblador, pero ello impedía la portabilidad a diferentes ordenadores. Después de que Dennis Ritchie en el 1973 crease el lenguaje C, se reescribe UNIX totalmente en este lenguaje de alto nivel, haciendo por tanto el código casi totalmente independiente del tipo de máquina, permitiendo la instalación de UNIX en diferentes plataformas.

Inicialmente los laboratorios AT&T Bell, consideran que UNIX era más bien un proyecto de investigación y lo llegó a distribuir de forma gratuita entre departamentos informáticos de las universidades, los cuales lo podían modificar y adaptar a sus necesidades. Pero, la gran demanda del sistema operativo hace que los laboratorios Bell iniciara su venta a través de distribuciones oficiales concediendo a los usuarios que lo requerían licencias de uso.

Debido a las múltiples versiones en el mercado de UNIX, el IEEE especifica una familia de estándares para definir una interfaz de programación de aplicaciones (API) para que todas las versiones fuesen 'compatibles'. A esta

familia se le conoce como POSIX (Portable Operating System Interface; la X viene de UNIX como seña de identidad de la API)

Linux, se creó en 1991 por Linus Torvalds basándose en otros dos sistemas operativos:

- El sistema abierto UNIX.
- El sistema educativo Minix creado en 1987 por Andrew S. Tanenbaum.

Torvalds crea sólo el Kernel, el núcleo del sistema sin la capa de servicios, gestores, aplicaciones graficas, etc. que serán creadas posteriormente por otros autores. El código del núcleo lo podemos encontrar en la dirección ([www.kernel.org](http://www.kernel.org)).

En la comunidad de programadores se crea el proyecto GNU (Gnu's Not Unix), proyecto para generar software libre, donde se generan editores, compiladores, etc. bajo la licencia pública general GPL (General Public License): usar, copiar, distribuir y modificar siempre que se conserve la firma del autor, pudiendo cobrar por ello.

Linux se crea con esta filosofía de libre distribución y el sistema operativo completo que se construye con este núcleo también. A todo el sistema se le da el nombre de GNU/Linux (distribución completa del sistema operativo con Linux), que contiene el núcleo más las otras capas del sistema operativo y utilidades. Si bien muchas veces se denomina a todo el sistema simplemente LINUX.

#### **44.2.1 Características**

Las características más relevantes del sistema UNIX son:

- UNIX ha sido diseñado como un sistema multiusuario en tiempo compartido; ofreciendo protección de los datos privados sobre ficheros y protección del entorno de ejecución.
- Portabilidad: UNIX fue escrito en el lenguaje C, un lenguaje de alto nivel, lo cual hace que sea relativamente fácil de leer, entender,

modificar y transportar a otras máquinas con una arquitectura física diferente.

- Código y funcionamiento escrito bajo la familia de estándares POSIX (Portable Operating System Interface).
- Interfaz de usuario simple e interactiva: el intérprete de órdenes (shell) es un programa independiente que el usuario puede sustituir. La sintaxis de utilización es idéntica para todas las órdenes.
- Modularidad: Proporciona primitivas que permiten construir grandes programas a partir de otros más sencillos, así como librerías para linkaje.
- Posee bibliotecas compartidas para facilitar el enlace dinámico.
- Protecciones de memoria.
- Soporta diferentes sistemas de archivos, incluidos los de Microsoft Windows
- Sistema de archivos con estructura de árbol invertido (de múltiples niveles que permite un fácil mantenimiento) y jerárquico (permite la unión de diversos sistemas de ficheros con el sistema principal, y una separación de directorios).
- Todos los archivos de usuario son simples secuencias de bytes (8 bits ), no tienen ningún formato predeterminado.
- Independencia de dispositivos: Los discos y los dispositivos de entrada y salida (E/S) se tratan todos de la misma manera: como meros archivos. Las peculiaridades de los dispositivos se mantienen en el núcleo (kernel).
- La arquitectura de la máquina es completamente transparente para el usuario, lo que permite que los programas sean fáciles de escribir y transportables a otras máquinas con hardware diferente



- UNIX no incorpora diseños sofisticados; de hecho, la mayoría de los algoritmos han sido seleccionados por su sencillez y no por su rapidez o complejidad.
- Incorpora todos los servicios de red, TCP/IP, DNS, sendmail, etc.
- Proporciona un completo entorno de programación: los filtros son utilidades simples que se concentran en realizar bien una sola función. Pueden combinarse de forma muy flexible utilizando las pipes (tuberías) y las redirecciones de E/S según las necesidades y preferencias de cada usuario.
- Mantenimiento fácil: consecuencia directa de la modularidad. El sistema sigue evolucionando y se perfecciona y enriquece con nuevas funcionalidades.
- Carácter abierto: permite ampliar fácilmente la funcionalidad con nuevos componentes sin tener que depender de un único fabricante.

#### **44.2.2 Arquitectura de Unix-Linux**

La arquitectura está basada en capas o niveles de forma que cada capa únicamente puede comunicarse con las capas que se hallan en los niveles inmediatamente inferior y superior.

En la capa inferior tenemos toda la parte del Hardware que el sistema operativo debe gestionar. Por encima de éste se sitúa el kernel de Unix que es el encargado de la administración de procesos, gestión del sistema de archivos, entradas / salidas, etc. A los procesos que trabajan a ese nivel se les llama procesos en modo kernel.

La biblioteca estándar se ubica por encima del kernel, se encarga, por ejemplo de las operaciones de apertura, cierre, etc. A este nivel se trabaja en modo usuario. La interface entre las dos capas, o el acceso de la capa

de biblioteca estándar a la del kernel se realiza a través de la interfaz de llamadas al sistema.

A un nivel superior tenemos los programas y utilidades como el shell, compiladores, etc., que sirven de ayuda a desarrolladores y usuarios que interactúan con el sistema operativo. La interface entre esta capa y la inmediatamente inferior es a través de la interfaz de biblioteca.

Por último, se ubicarían los usuarios que por medio de la interfaz de usuario se comunican con el shell, u otras utilidades del sistema Unix.

El núcleo de UNIX (kernel) es de tipo monolítico, diferenciándose dos partes principales: el núcleo dependiente de la máquina y el núcleo independiente. El núcleo dependiente se encarga de las interrupciones, los dispositivos de bajo nivel y parte de la administración de la memoria. El núcleo independiente es igual en todas las plataformas e incluye la gestión de llamadas del sistema, la planificación de procesos, la paginación e intercambio, la gestión de discos y del sistema de archivos

#### **44.2.3 Gestión de procesos.**

La gestión de procesos en UNIX es por prioridad y round robin. En algunas versiones se gestiona también un ajuste dinámico de la prioridad de acuerdo al tiempo que los procesos han esperado y al tiempo que ya han usado la CPU. El sistema provee facilidades para contabilizar el uso de CPU por proceso y una pila común para todos los procesos cuando necesitan estarse ejecutando en modo privilegiado (cuando hicieron una llamada al sistema).

Los procesos trabajan en modo usuario y en modo kernel. El paso de modo usuario a kernel o viceversa se realiza a través de traps que crean una interrupción para acceder a la interfase de llamadas al sistema y al resto de los componentes de nivel kernel. El paso del modo kernel a usuario es un retorno tras la realización de la petición que motivó el paso al modo kernel.

UNIX permite que un proceso haga una copia de sí mismo por medio de la llamada «fork», lo cual es muy útil cuando se realizan trabajos paralelos o concurrentes; también se proveen facilidades para el envío de mensajes entre procesos (pipes, signals).

Los procesos no interactivos se denominan daemons o procesos background. Cuando se inicia un proceso, se le asigna un identificador PID, se guarda el proceso que lo lanzó PPID, el propietario que lo lanzó UID y el grupo de pertenencia GID, lo que definirá el perfil de permisos de acceso a los que tendrá derecho. Existe la posibilidad de alterar el usuario o grupo efectivo de permisos, durante la ejecución del proceso, mediante la llamada a **setuid** o **setgid**, siempre que se disponga de los permisos apropiados. También existe una bandera de permisos **setuid** asociada al archivo del programa que permite ejecutar éste, con los permisos del propietario del archivo en lugar de los del usuario que lo ejecuta.

LINUX combina multiprogramación y tiempo compartido.

El gestor de procesos en el kernel del sistema UNIX, se encarga de la asignación de CPU, la programación de procesos, y las solicitudes de los procesos. Para realizar estas tareas, el kernel mantiene varias tablas importantes para coordinar la ejecución de estos procesos y la asignación de los dispositivos.

Utilizando una política predefinida, el programador de procesos selecciona un proceso de la cola de procesos listos y comienza su ejecución durante un pedazo de tiempo ya dado.

El algoritmo de programación de procesos selecciona el proceso con la mayor prioridad para ser ejecutado primero. Si varios procesos tienen la misma prioridad, se aplica el algoritmo round-robin.

#### **44.2.4 Gestión de memoria.**

Los sistemas UNIX utilizan el manejo de memoria virtual siendo el esquema más usado la paginación por demanda y combinación de segmentos paginados, en ambos casos con páginas de tamaño fijo.

En todos los sistemas UNIX se usa una partición de disco duro para el área de intercambio (swap). Esa área se reserva durante la instalación del sistema operativo.

Una regla muy difundida entre administradores de sistemas es asignar una partición de disco duro que sea al menos el doble de la cantidad de memoria real del ordenador. Con esta regla se permite que se puedan intercambiar flexiblemente todos los procesos que estén en memoria RAM en un momento dado por otros que estén en el disco.

Si no caben todos los programas en memoria principal se hace uso de la partición de intercambio (swapping)

- **Swap out** Cuando no caben en memoria procesos activos, se “expulsa” un proceso de memoria principal, copiando su imagen a swap, aunque no es necesario copiar todo el mapa. Existen diversos criterios de selección del proceso a intercambiar: dependiendo de la prioridad del proceso; preferencia a los procesos bloqueados; no intercambiar si está activo DMA sobre mapa del proceso.
- **Swap in.** Cuando haya espacio en memoria principal, se intercambia el proceso a memoria copiando la imagen desde swap.

Todos los procesos que forman parte del kernel no pueden ser intercambiados a disco.

Cada proceso dispone de su propio espacio de direcciones, organizado en segmentos según:

- **Text Segment:** que almacena el código

- **Data Segment:** que almacena los datos o variables que utilizan los procesos, éste segmento tiene dos partes: **Initialized Data** (datos inicializados) y **Uninitialized Data** (datos no inicializados).
- **Stack Segment:** que almacena la información referente a llamadas a otras funciones.

Es posible compartir código entre procesos mediante el empleo de **Shared Text Segments**. Dos procesos nunca comparten los segmentos de datos y de pila (salvo los thread), la forma de compartir información se lleva a cabo mediante el empleo de segmentos especiales de memoria compartida **Shared Segments**.

Linux comparte muchas de las características de los esquemas de gestión de memoria de otras implementaciones UNIX, pero tiene sus características propias.

En lo que respecta a memoria virtual, el direccionamiento de memoria virtual de Linux, hace uso de una estructura de tabla de páginas con tres niveles, formada por los siguientes tipos de tablas (cada tabla individual es del tamaño de una página): **Directorio de páginas** un proceso activo tiene un solo directorio de páginas que es del tamaño de una página. Cada entrada en el directorio de páginas apunta a una página del directorio intermedio de páginas. Para un proceso activo, el directorio de páginas tiene que estar en la memoria principal; **Directorio intermedio de páginas** este directorio puede ocupar varias páginas y cada entrada de este directorio apunta a una página de la tabla de páginas; **Tabla de páginas** esta tabla de páginas también puede ocupar varias páginas, y cada entrada de la tabla de página hace referencia a una tabla virtual del proceso.

Para utilizar esta estructura de la tabla de páginas a tres niveles, una dirección virtual en Linux se ve como un conjunto de cuatro campos. El campo más a la izquierda (más significativo) se utiliza como índice en el directorio de páginas. El siguiente campo sirve como índice en el directorio intermedio de páginas. El tercer campo sirve como índice en la tabla de

páginas. Y el cuarto y último campo, indica el desplazamiento dentro de la página seleccionada de la memoria

#### **44.2.5 Gestión de E/S**

Los dispositivos de entrada y salida son considerados ficheros especiales: Toda entrada/salida está basada en el principio de que todos los dispositivos se pueden tratar como ficheros simples que se acceden mediante descriptores de archivos cuyos nombres se encuentran generalmente en el directorio «/dev».

Cada proceso en UNIX mantiene una tabla de archivos abiertos (donde el archivo puede ser cualquier dispositivo de entrada/salida). Esa tabla tiene entradas que corresponden a los descriptores, los cuales son números enteros obtenidos por medio de la llamada del sistema.

Las llamadas al gestor de entrada/salida se hacen de dos formas: síncrona y asíncrona. El modo síncrono es el modo normal de trabajo y consiste en hacer peticiones de lectura o escritura y el proceso espera a que el sistema le responda.

El gestor de entrada/salida utiliza como elementos principales el buffer de cache; el código general de gestión de dispositivos, y drivers de dispositivos de hardware. Existen dos tipos de dispositivos:

Dispositivos de bloques:

- Usan secuencias de bytes (bloques)
- Utilizan buffer-cache
- Están estructurados en bloques de tamaño fijo (512 bytes)
- Permiten optimizar el rendimiento

Dispositivos de carácter:

- Son dispositivos sin estructura (terminales, impresoras, etc)
- No usan buffer

- La operaciones se realizan carácter a carácter

### **Interrupciones y excepciones**

UNIX permite interrumpir la CPU asíncronamente. Al recibir la interrupción, el kernel almacena el contexto actual, determina la causa y responde a la interrupción. Tras responder a ésta, devuelve el contexto interrumpido y sigue ejecutando. El HW asigna las prioridades a los dispositivos de acuerdo con el orden de actuación en las interrupciones.

Así como las interrupciones están causadas por factores externos a un proceso, las excepciones son sucesos inesperados producidos por procesos, tales como la ejecución de instrucciones reservadas, de forma que el sistema, al encontrarse con una, tiende a reiniciar la instrucción, en lugar de pasar a la siguiente.

No obstante, el kernel debe tener la posibilidad de impedir la aparición de interrupciones en momentos críticos, para evitar la degradación de los datos. El sistema que se utiliza es el de disponer de un conjunto de instrucciones restringidas que colocan el nivel de ejecución del procesador en el estado de palabra (status word). Al asignar un nivel de ejecución del procesado, todas las interrupciones de ese nivel e inferiores quedan suprimidas, permitiendo sólo las superiores.

#### **44.2.6 Gestión de archivos**

Un sistema de archivos permite realizar una abstracción de los dispositivos físicos de almacenamiento de la información para que sean tratados a nivel lógico, como una estructura de más alto nivel y más sencilla que la estructura de su arquitectura hardware particular.

El sistema de archivos UNIX se caracteriza porque posee una estructura jerárquica, realiza un tratamiento consistente de los datos de los archivos,

protege los datos de los archivos y trata a los dispositivos y periféricos (terminales, unidades de disco, cinta, etc.) como si fuesen archivos.

El sistema de archivos está organizado, a nivel lógico, en forma de árbol invertido, con un nodo principal conocido como nodo raíz ("/"). Cada nodo dentro del árbol es un directorio y puede contener a su vez otros nodos (subdirectorios), archivos normales o archivos de dispositivo.

Los nombres de los archivos (pathname) se especifican mediante la ruta (path), que describe cómo localizar un archivo dentro de la jerarquía del sistema. La ruta de un archivo puede ser absoluta (referida al nodo raíz) o relativa (referida al directorio de trabajo actual).

Todos los sistemas UNIX pueden manejar múltiples particiones de disco, cada una con un sistema de archivos distinto.

Una partición de disco clásica en UNIX contiene una estructura como la de la figura.

- Bloque de arranque: contiene el código para arrancar el ordenador con esa partición.
- Superbloque: Contiene información crucial acerca de la organización del sistema de archivos, incluido el número de nodos-i (i-nodes), el número de bloques de disco y el principio de la lista de bloques de disco libres. La destrucción del superbloque hace que el sistema de archivos ya no pueda leerse.
- Nodos-i: contiene la representación interna de un fichero que permite entre otras cosas localizar todos los bloques de disco que contienen los datos del archivo
- Bloques de Datos: almacenan la información. Lo normal es que un archivo ocupe más de un bloque de datos no siendo necesario que estén contiguos.



Dentro de la estructura de directorios de UNIX – Linux existen una serie de directorios comunes a todas las instalaciones que es preciso conocer:

- `/`: Directorio raíz, inicio del sistema de archivos.
- `/tmp`: Directorio de archivos temporales.
- `/dev` : Directorio de dispositivos. En él se encuentran todos los dispositivos de E/S, que se tratan como archivos especiales.
- `/etc`: Directorio para archivos de sistema diversos.
- `/bin`: Directorio para programas binarios (ejecutables).
- `/lib`: Directorio de bibliotecas del sistema.
- `/usr`: Directorio de usuarios.
- `/home`: Directorio base a partir del cual se ubican los directorios por defecto de las cuentas de usuario.

LINUX comenzó empleando el sistema de archivos de MINIX, pero estaba limitado a nombres de 14 caracteres y a archivos de 64 MB de tamaño. La primera mejora vino de la mano de un sistema de archivos denominado Ext que permitía nombres de archivo de 255 caracteres y 2 GB de tamaño por archivo, pero era muy lento. La evolución vino de la mano del sistema de archivos Ext2, con nombres de archivo largos, archivos grandes y mejor rendimiento. Ext2 evoluciona a Ext3 que trajo principalmente las transacciones (journaling) a Ext2. En ext3 se almacena la información necesaria para restablecer los datos afectados por la transacción en caso de que ésta falle. La evolución de ext3 es **ext4** que soporta un tamaño máximo de sistema de archivos de 1 ExaByte (1 ExaByte = 1024 PetaBytes = 1048576 TeraBytes) y un tamaño máximo por archivo de 16 TB para los archivos y por otro lado, modifica estructuras de datos importantes, como la destinada a almacenar los datos del archivo utilizando **“extent”** que es un conjunto de bloques físicos contiguos,

mejorando el rendimiento al trabajar con ficheros de gran tamaño y reduciendo la fragmentación.

### **44.3.- SISTEMA OPERATIVO WINDOWS**

#### **44.3.1 Características**

La familia de S.O. Windows es propiedad de Microsoft. Se podría decir que básicamente existen dos versiones distintas de S.O. Windows: aquellos enfocados al mundo empresarial y aquellos enfocados al consumidor final o usuario doméstico.

Dentro de esta simple clasificación podríamos hacer otras clasificaciones. Orientado al usuario domestico existen S.O. para equipos de sobremesa, portátiles, tablets, y dispositivos móviles. Y orientado al mundo empresarial podemos distinguir S.O. para servidores y S.O. para estaciones de trabajo. Aún podríamos dividir más los S.O. para servidores (Standard, Enterprise, Datacenter, Web, Storage, Small Business Server)...

Los últimos S.O. que han visto a luz son Windows 7 con las versiones Starter (sólo para 32 bits), Home Basic, Home Premium, Professional, Ultimate, Enterprise. Y a nivel de S.O. para servidores es Windows Server 2008 R2.

Es destacable el interés de mejora adquirido por Microsoft con los S.O. de servidor, comprometiéndose a un cambio cada 5 años (antes del 2008 estaba el 2003) y a una revisión cada 2 años aprox. (por eso existe Windows Server 2008 R2).

Windows Server 2008 es el último S.O. de servidor que incluye estas mejoras:

- Nuevo proceso de reparación de sistemas NTFS: proceso en segundo plano que repara los archivos dañados.

- Creación de sesiones de usuario en paralelo: reduce tiempos de espera en los Terminal Services y en la creación de sesiones de usuario a gran escala.
- Cierre limpio de Servicios.
- Sistema de archivos SMB2: de 30 a 40 veces más rápido el acceso a los servidores multimedia.
- Address Space Load Randomization (ASLR): protección contra malware en la carga de controladores en memoria.
- Windows Hardware Error Architecture (WHEA): protocolo mejorado y estandarizado de informe de errores.
- Virtualización de Windows Server: mejoras en el rendimiento de la virtualización.
- PowerShell: inclusión de una consola mejorada con soporte GUI para administración.
- Server Core: el núcleo del sistema se ha renovado con muchas y nuevas mejoras.

En los siguientes apartados se verán características de Windows, y si bien la mayoría son comunes para todas las versiones, las explicaciones estarán más centradas en Windows 2008 Server R2.

#### **44.3.2 Arquitectura de Windows**

La estructura modular de Windows 2008 aporta una gran flexibilidad. Su diseño le permite ejecutarse en una gran variedad de plataformas hardware. En esta estructura modular se distinguen dos capas principales:

- **Modo usuario**: Cuyos programas y subsistemas están limitados a los recursos del sistema a los que tienen acceso. Está formado por

subsistemas que pueden pasar peticiones de E/S a los controladores apropiados del modo núcleo a través del gestor de E/S.

- **Modo núcleo o kernel**: Tiene acceso total al hardware de la máquina, impidiendo a los servicios del modo usuario y a las aplicaciones acceder a al hardware que queda totalmente protegido por el sistema operativo. La arquitectura dentro del modo núcleo se compone de lo siguiente:
  - o El micronúcleo: situado entre la capa de abstracción de hardware y el Executive, proporciona la gestión *multiprocesador*: gestión de procesos, hilos y tratamiento de interrupciones, y de excepciones.
  - o Una capa de abstracción de hardware (en inglés Hardware Abstraction Layer o HAL), se encarga de ocultar las diferencias de hardware y por tanto proporciona una plataforma única donde pueda ejecutarse el S.O. independientemente del HW.
  - o Controladores o también llamados drivers: utilizados para interactuar con los dispositivos hardware.
  - o Executive: sobre el cual son implementados todos los servicios de alto nivel. Se relaciona con todos los subsistemas del modo usuario. Se ocupa de la entrada/salida, la gestión de memoria, Plug&Play, la seguridad y la gestión de procesos.

Dado que el enlace estático de los programas de usuario con las bibliotecas de la API Win32 conllevaría un tamaño enorme en los programas y un desperdicio de memoria, pues cada programa en ejecución tendría su copia de estas bibliotecas, todas las versiones de Windows manejan bibliotecas compartidas, llamadas bibliotecas de vínculos dinámicos (DLLs; Dinamic Link Libraries).

#### **44.3.3 Gestión de procesos.**

Los procesos se crean como objetos, y un proceso puede tener varios hilos. Dado que el proceso es un objeto, su composición será un conjunto de

datos sólo accesibles a través de un conjunto de funciones que los ocultan del resto de aplicaciones o funciones. Estas funciones o servicios se activan por medio de mensajes. El proceso tendrá al menos un hilo, que a su vez puede ejecutar otros hilos, pudiendo hacerlo en paralelo en un sistema multiprocesador.

Windows mantiene dos listas diferentes con la información de todos los procesos e hilos. Cada proceso tiene asociado un bloque o estructura de datos EPROCESS, que apunta al EPROCESS siguiente y al anterior (doble lista enlazada), y la otra estructura es ETHREAD para recoger la información de los hilos.

El algoritmo de planificación está basado en colas de retroalimentación de múltiples niveles con prioridades. Cada cola se gestiona con una política Round Robin.

La planificación se aplica sobre los hilos, no a los procesos (sin tener en cuenta a qué proceso pertenecen los distintos hilos que se ejecutan), y está basada en prioridades, es decir, siempre se ejecutará el hilo de mayor prioridad de la cola de hilos preparados.

Cuando se selecciona un hilo para su ejecución, se le concede un quantum, o intervalo de tiempo durante el cual se permite al hilo ejecutarse antes de que lo haga otro hilo del mismo nivel de prioridad. Los valores de quantum pueden variar.

Aunque se conceda un quantum a un hilo, éste podría no consumirlo completamente porque aparezca en el sistema un nuevo proceso de mayor prioridad, que obligaría al que se está ejecutando a abandonar el procesador.

El sistema trata igual a todos los hilos que tengan la misma prioridad, asignando a cada hilo de mayor prioridad un intervalo de tiempo de procesador con un Round Robin. Si ninguno de estos hilos estuviera preparado para ejecutarse, pasarían a ejecutarse, con la misma política, los de la prioridad inmediatamente inferior.

En cada uno de estos casos, Windows debe determinar qué hilo debe ejecutarse a continuación, y esta decisión es lo que se conoce como dispatcher.

Cada proceso recibe una prioridad base para todos sus hilos. El sistema se basa en 32 prioridades, del 0 (menor prioridad) al 31 (mayor prioridad):

- La prioridad 0 está reservada para el hilo de sistema responsable de poner a cero las páginas libres cuando no las necesiten ningún hilo.
- Las prioridades 1 a 15 son las reservadas para los procesos de usuario (prioridades variables).
- Las prioridades 16 a 31 están reservadas al sistema operativo (prioridades en tiempo real).

El planificador funciona accediendo a la tabla por el proceso de prioridad 31 y viendo si tiene hilos listos para ejecutar. Si los hay toma el primero de la lista y lo ejecuta durante un quantum. Mientras existan procesos preparados de una prioridad superior, el sistema les concederá todo el tiempo que precisen. Este comportamiento se repite para cada una de las entradas de la tabla de prioridades.

En ciertas condiciones un subproceso puede ver incrementada su prioridad base, pero nunca por encima de la prioridad 15 y nunca para subprocesos de prioridad mayor de 15. Si una operación de E/S libera un subproceso, éste ve incrementada su prioridad base de modo que pueda ejecutarse pronto.

También se produce aumento de prioridad si el subproceso estaba esperando por un semáforo, mutex u otro suceso. Estas elevaciones de prioridad van disminuyendo a medida que un subproceso beneficiado va consumiendo por completo su quantum, hasta volver a situarse en su prioridad base.

#### **44.3.4 Gestión de memoria.**

La gestión de memoria en Windows es de memoria virtual con paginación.

Las aplicaciones de 32 bits tienen un espacio de dirección del proceso de 4 GB de memoria. Los sistemas operativos de Microsoft Windows proporcionan a las aplicaciones acceso a 2 GB de espacio de dirección del proceso, específicamente conocido como espacio de dirección virtuales del modo de usuario. Todos los subprocesos pertenecientes a una aplicación comparten el mismo espacio de direcciones virtuales del modo de usuario. Los 2 GB restantes se reservan para el sistema operativo (también conocido como espacio de dirección del modo de kernel).

El espacio de direcciones virtual se pagina por demanda con tamaño fijo de páginas (mínimo 4KB para arquitecturas x86 y x64bits y 8KB para arquitecturas IA64 y máximo de 4MB para arquitectura x86, 2MB para x64 y 16MB para IA64).

Windows utiliza un algoritmo de **paginación por demanda anticipada**, es decir, cada vez que se produce un fallo de página, el sistema copiará en memoria la página correspondiente a la referencia a memoria que ha causado el fallo de página y además un conjunto de páginas próximas a ella, tanto anteriores como posteriores, al suponer que, debido a la localidad de las referencias, es casi seguro que en un futuro próximo también se hará referencia a estas páginas, que cuando se quieran utilizar ya estarán en memoria y, por lo tanto, no producirán fallos de páginas adicionales.

El mecanismo de paginación se apoya mucho en el concepto de **Conjunto de Trabajo (Working Set)** que asegura una cierta cantidad de memoria física para cada proceso.

Windows presta especial atención al momento de arranque de los procesos ya que, como no tienen ninguna página cargada en la memoria, hasta que carguen todas las páginas necesarias se producirán muchos fallos de

página. Para optimizar la carga de los procesos, Windows cuenta con lo que se conoce como “Prefetcher” cuya misión es acelerar el proceso de carga.

Si se produce un fallo de página y es necesario sustituir algún marco de página que está en memoria, Windows emplea el algoritmo LRU (aunque algunas versiones utilizan también FIFO).

Permite compartir páginas, al poder proteger contra lectura o escritura las mismas. Igualmente admite que se pueda bloquear una página en memoria que sea crítica, impidiendo que pueda sustituirse ante una falta de página, facilitando así la implementación de aplicaciones en tiempo real.

#### **44.3.5 Gestión de E/S**

El sistema de entrada/salida (E/S) de Windows es el que permite utilizar los dispositivos facilitando el acceso a los mismos e independizando a los programas de los dispositivos, ofreciendo además la seguridad en su uso y la escalabilidad del sistema.

Las entradas y salidas en Windows pueden ser síncronas (el proceso esperará hasta que se haya completado la operación en el dispositivo hardware) o asíncronas (el proceso lanza la operación y sigue con su ejecución y cuando la operación E/S finaliza el S.O. lo avisa).

En Windows se cargan y descargan los drivers en cualquier momento, evitando que consuman recursos si no se van a utilizar.

Esto se hace gracias al Plug and Play (PnP) que permite detectar cualquier dispositivo que se conecte al sistema y cargar el driver correspondiente.

El sistema de E/S se compone de los siguientes módulos:

- El gestor de E/S: define la infraestructura que soporta los drivers de dispositivos. Forma parte del sistema operativo.



- El driver de dispositivo: proporciona un interface de E/S para un determinado tipo de dispositivo. Los drivers reciben peticiones canalizadas a través del gestor de E/S y las dirigen al dispositivo concreto, e informan al gestor de que se ha completado la operación de E/S. Estos módulos los desarrollan los fabricantes.
- El gestor de PnP: detecta los dispositivos hardware al conectarse o desconectarse.
- El gestor de energía: facilita al sistema, así como a los drivers de dispositivo, los cambios de estado de consumo de energía eléctrica de acuerdo con la actividad del dispositivo.

#### **44.3.6 Gestión de archivos**

En Windows, la asignación del espacio la realiza el subsistema de ficheros en unidades “cluster”, cuyo tamaño depende de la capacidad del disco, normalmente oscila desde 512 bytes hasta 4 Kbytes. Utiliza 64 bits para direccionar los cluster y permite definir ficheros de 264TB (16.384 petabytes), aunque, lógicamente, el tamaño máximo de los ficheros está limitado por la capacidad de los discos.

Windows gestiona los discos y la información que contienen en base a particiones y volúmenes.

- **Particiones**. Cada disco se puede dividir en particiones primarias y extendidas. Las particiones primarias serán aquellas que puedan contener un S.O. y, por lo tanto, permitan el arranque del S.O. desde las mismas. De aquí podemos deducir, que el sistema requiere como mínimo una partición primaria en algún disco.

Una partición nunca podrá sobrepasar un disco. Sólo puede haber una partición extendida por disco, y como máximo sólo podrá contener 4 particiones en total.

Una vez creada la partición, es necesario darle formato para que pueda contener datos. Un volumen es sinónimo de partición formateada.

En Windows, las particiones las gestiona el gestor de particiones. Este gestor utiliza el gestor de E/S para identificar las particiones y crear los dispositivos que las representen, es decir, las unidades lógicas correspondientes.

Este gestor envía un comando al gestor de volúmenes (descrito más adelante) para saber si la partición tiene un volumen asociado, y si es así, a partir de ese momento cualquier acción sobre la partición se la notificará al gestor de volúmenes.

- **Volumen.** Un volumen desde el punto de vista del usuario es una partición formateada. Las particiones primarias sólo podrán contener un volumen, mientras que las extendidas podrán albergar varios, teniendo en cuenta que un sistema sólo podrá tener como máximo 24 volúmenes, ya que se identifican por medio de las letras del abecedario, y en inglés sólo tiene 24 letras.

En Windows Server podemos trabajar con dos tipos de discos:

- Discos básicos. Son los que se basan exclusivamente en tablas de particiones MBR (Master Boot Record) o tablas GPT (GUID Partition Table).
- Discos dinámicos. Se basan en *volúmenes dinámicos*, que permiten la creación de volúmenes de particiones múltiples tales como simples, distribuidos, espejos, stripes y RAID-5. Los discos dinámicos se particionan con el Administrador de discos lógicos (LDM – Logical Disk Manager).

Windows trabaja con los siguientes tipos de volúmenes dinámicos:

- Volúmenes distribuidos (spanned). Es un único volumen lógico compuesto por un máximo de 32 particiones libres en uno o más discos. Es una forma de juntar el espacio no asignado en un sistema con varios discos en una única unidad lógica.

- Volúmenes Espejo. En este tipo de volumen, el contenido de la partición se duplica en una partición idéntica en otro disco, aunque se ven como un único volumen y no como dos. Los volúmenes espejo se conocen como RAID de nivel 1 (RAID1) y son tolerantes a fallos.
- Volúmenes divididos (Striped). Similar al volumen distribuido, utiliza el espacio de varios discos y los convierte en una única unidad lógica. Utiliza un tipo especial de formato para escribir en el disco y tiene más rendimiento que el volumen distribuido. Los fallos de escritura suelen ser mayores que en el caso del volumen distribuido. Se conocen como RAID de nivel 0 (volúmenes RAID-0).
- Volúmenes RAID-5. Como los volúmenes striped, pero con tolerancia a fallos ya que distribuyen la información de paridad entre todos los discos miembros del volumen.

### **Sistemas de ficheros**

Windows soporta los siguientes formatos de sistemas de ficheros:

A) CDFS (sistema de ficheros de CD-ROM): sólo permite la lectura, y soporta los formatos de disco ISO-9660 y Joliet.

B) UDF: es un subconjunto del formato ISO-13346 con extensiones para formatos como CD-R y DVD-R/RW. UDF está incluido en la especificación DVD y es más flexible que el CDFS.

C) FAT12, FAT16, y FAT32: Windows es compatible con el sistema de ficheros FAT por compatibilidad con MS-DOS y otras versiones de Microsoft Windows. El formato FAT (File Allocation Table) incluye un mapa de bits que se utilizan para identificar clusters o bloques en el disco.

FAT32 tiene una capacidad teórica para direccionar volúmenes de 8 terabytes (TB), no obstante limita los volúmenes a un máximo de 32 GB.

D) NTFS (New Technology File System): es el formato nativo de Windows Server para los sistemas de ficheros. NTFS utiliza direcciones de disco de 64 bits con lo que podría gestionar volúmenes de hasta 16 exabytes, sin

embargo, Windows limita el tamaño de un volumen NTFS a lo que se pueda direccionar con 32 bits, que es un poco menos de 256 TB (con clústeres de 64 KB). NTFS admite ficheros de máximo 16 TB.

NTFS añade características de seguridad de ficheros y directorios, cuotas de disco, compresión de ficheros, enlaces simbólicos basados en directorios, y cifrado. Una de sus características más significativas es la recuperabilidad: registra los cambios que se realizan en los metadatos como si fueran transacciones con la finalidad de que se puedan recuperar en el caso de la pérdida de ficheros o de sus datos.

La estructura central de NTFS es la tabla maestra de archivos MFT (Master File Table), que es una sucesión lineal de registros de tamaño fijo (1 KB). Cada registro de MFT describe un archivo o un directorio, contiene los atributos del archivo, como su nombre y marcas de tiempo, y la lista de direcciones de disco donde están sus bloques. Si un archivo es demasiado grande puede ser necesario emplear más de un registro MFT para contener la lista de todos los bloques. En este caso al primer registro se le denomina registro base, y apunta a los demás registros MFT.

#### **44.3.7 Seguridad**

El administrador de seguridad, componente del Executive, hace que se respete el complejo mecanismo de seguridad de Windows 2008, que satisface los requisitos C-2 del Libro Naranja del Departamento de Defensa de Estados Unidos.

Cada usuario y grupo de Windows 2008 se identifica con un SID (Security ID) único a nivel mundial. Cada proceso lleva asociado una ficha de acceso que especifica su SID y otras propiedades.

Cada objeto tiene asociado un descriptor de seguridad que indica quién puede realizar qué operaciones con él. Un descriptor de seguridad está

formado por un encabezado, seguido de una DACL (discretionary Access Control List) con uno o más elementos de control de acceso (ACE). Los más importantes son Allow y Deny. Además del DACL, el descriptor tiene una SACL (System Access Control List) que no especifica quién puede usar el objeto sino qué operaciones con el objeto se asientan en el registro de sucesos de seguridad del sistema (función de auditoría).

En un sistema autónomo la validación corre por cuenta del proceso winlogon y la configuración de seguridad almacenada en la propia máquina en las claves del registro: SECURITY y SAM. Donde la primera establece las políticas globales de seguridad y la segunda la seguridad específica de cada usuario.

En un sistema en red, la autenticación de los usuarios está centralizada en ciertos servidores denominados controladores del dominio. Los equipos se organizan dentro de Dominios, pudiendo éstos estar gestionados mediante el empleo del Active Directory.

Windows 2008 dispone de administración centralizada de certificados. En las versiones anteriores se confiaba que cada aplicación mantenía su propia lista de claves o CA confiables.

El protocolo KERBEROS (RFC 1510), que es un estándar de Internet para autenticación, es el método nativo que emplean los sistemas Windows 2008. Cualquier servidor del Directorio Activo, automáticamente, tiene el servicio del Centro de distribución de claves de Kerberos (KDC- Kerberos Key Distribution Center).

#### **44.4.- BIBLIOGRAFÍA**

- Sistemas Operativos Modernos. Tanenbaum, Andrew. Prentice Hall, 2005
- Linux Bible 2008 Edition. Christopher, N. Wiley Publishing, Inc 2008.

- Windows internals 5<sup>th</sup> Edition. Mark E. Russinovich, David A. Solomon, Alex Ionescu. Microsoft Press, 2009

**Autor:** Francisco Javier Rodríguez Martínez

Subdirector de Sistemas Escola Superior Enxeñaría Informática Ourense

Colegiado del CPEIG