

Sistemas operativos: Gestión de
archivos y dispositivos

TEMA 19

ABACUS NT

Oposiciones 2021

Índice

- 1. Introducción**
- 2. Estructura lógica del disco. Particionamiento.**
 - 2.1. MBR vs GPT**
- 3. Sistema de archivos**
 - 3.1. Archivos y directorios (o carpetas)**
 - 3.2. Asignación y gestión del espacio libre**
 - 3.3. Protección y seguridad**
 - 3.4. Rutas y nombre de archivos**
 - 3.5. Archivos compartidos**
- 4. Implementación de sistemas de archivos.**
 - 4.1. Sistemas de archivos en GNU/Linux**
 - 4.1.1. Estructura de EXT**
 - 4.1.2. Características de EXT4**
 - 4.2. Sistemas de archivos en Windows**
 - 4.2.1. FAT32**
 - 4.2.2. NTFS**
 - 4.3. RAID**
- 5. Sistemas operativos: Gestión de dispositivos**
 - 5.1. Planificación de disco**
 - 5.2. Planificación en otros dispositivos**
- 6. Gestión de dispositivos en sistemas operativos actuales**
 - 6.1. GNU/Linux**
 - 6.2. Windows**
- 7. Conclusión**
 - 7.1. Relación del tema con el currículo**
- 8. Bibliografía**

1. Introducción

La información en la sociedad actual es un aspecto relevante en múltiples ramas del conocimiento; incluyendo la biología genética, la física cuántica, la divulgación científica y por supuesto la informática, Internet y el tratamiento de la información.

Cada día, se intercambian unos 2,5 billones de bytes de datos. Se estima que el 90% de los datos existentes en el mundo actual se han generado en el curso de los últimos dos años. Este diluvio se conoce como **Big Data**.

Nos enfrentamos al reto de almacenar de forma fiable, duradera y fácilmente transmisible, **exabytes** de datos. Para ello necesitamos soportes que mantengan la información más allá del ciclo de trabajo de un ordenador, y una estructura de datos que nos garantice su acceso.

Los ficheros son la estructura de datos en los que se apoya el almacenamiento en memoria secundaria. Tanto programas como datos, incluyendo bases de datos completas y copias de seguridad, se almacenan en última instancia en ficheros sobre dispositivos auxiliares.

La gestión que lleve a cabo el sistema operativo sobre estos ficheros y los dispositivos en los que los almacene serán clave en el eficiente desempeño del sistema.

2. Estructura lógica del disco. Particionamiento.

Pistas, Cilindros y Sectores

El disco está organizado en platos, que se dividen en delgados círculos concéntricos llamadas pistas. Las pistas están formadas por sectores, que son un conjunto de segmentos concéntricos de cada una de las pistas. En un sistema con varios platos y cabeza móvil, aquellas pistas que se acceden en una misma posición constituyen un cilindro. Dado que las cabezas están alineadas unas con otras, la controladora puede escribir en todas las pistas del cilindro sin mover el cabezal, por ello los discos con más platos son más rápidos.

Cluster (unidades de asignación)

Un cluster, o unidad de asignación, está formado al menos por un sector lógico y puede tener un tamaño mayor. Este tamaño se define al formatear el disco. Si se utiliza un tamaño de cluster grande, se efectuarán menos lecturas/escrituras por archivo, pero se desperdiciará más espacio (fragmentación interna). Si se utiliza un tamaño de cluster demasiado pequeño, los archivos grandes se fragmentan mucho, requiriendo muchas operaciones de lectura o escritura.

El tamaño de cluster estará por tanto determinado por el tipo de información que vayamos a almacenar en el disco, es decir, por el uso que le vayamos a dar.

Registro de arranque

El registro de arranque informa sobre la estructura del disco, tabla de particiones, fallos en el mismo, etc.

Actualmente existe una transición entre el sistema MBR (Master Boot Record) apoyado por las antiguas BIOS, y el GPT apoyado por el estándar UEFI.

Direccionamiento

El direccionamiento clásico (CHS=Cylinder, Head, Sector) para leer o escribir un dato en el disco, se efectúa dando al periférico los siguientes parámetros:

- Número de unidad
- Número de cilindro (C)
- Número de pista (H)
- Número del sector (S)

Actualmente se tiende a reemplazar el direccionamiento CHS por GUID (Global Unique ID), un número que identifica cada sector por un sólo número empezando por el cero.

El brazo sitúa rápidamente el cabezal encima de la pista correspondiente y espera a que el sector se posicione bajo la cabeza. Podemos considerar, entonces, dos tiempos:

- Tiempo de búsqueda de la pista (tseek)
- Tiempo de espera del sector (latencia)

2.1. MBR vs GPT

A la hora de dar formato a un disco duro, el estilo de tablas de particiones más conocido es MBR. Este estilo de formato lleva más de 30 años funcionando en la mayor parte de sistemas operativos, sin embargo, con las últimas versiones de Windows, especialmente coincidiendo con el auge de los sistemas UEFI, MBR está siendo sustituido por un nuevo estilo de particiones, GPT, más fiable, moderno y listo para acabar con las principales limitaciones de la estructura MBR.

Con Windows 8, Microsoft empezó a configurar GPT como tabla de particiones por defecto al dar un nuevo formato al disco. Poco a poco, GPT irá reemplazando a MBR como estilo de particiones por defecto. Ambos son dos formas diferentes de crear y gestionar las tablas de particiones de un disco duro.

MBR, acrónimo de Master Boot Record, es el estándar que empezó a funcionar en 1983 y que a día de hoy sigue siendo totalmente funcional, sin embargo, los años empiezan a pesar en él tal como avanza la tecnología. Una de las principales limitaciones de este estilo de particiones es el tamaño máximo con el que puede trabajar: 2 TB (aunque por software sí es posible superarlo, aunque no recomendable). Otra limitación, aunque no muy importante, es que MBR solo puede trabajar con

4 particiones primarias, por lo que para crear más de 4 debemos recurrir a las particiones extendidas.

GPT, acrónimo de GUID Partition Table, es el nuevo estándar que está sustituyendo a MBR y que está asociado con los nuevos sistemas UEFI. Su nombre viene de que a cada partición se le asocia un único identificador global (GUID), un identificador aleatorio tan largo que cada partición en el mundo podría tener su ID único. A día de hoy, GPT no tiene ningún límite más allá que los que establezcan los propios sistemas operativos, tanto en tamaño como en número de particiones (por ejemplo, Windows tiene un límite de 128 particiones).

La fiabilidad de los discos GPT es mucho mayor que la de MBR. Mientras que en esta segunda la tabla de particiones se almacena solo en los primeros sectores del disco, estando en problemas en caso de que se pierda, corrompe o sobrescriba, GPT crea múltiples copias redundantes a lo largo de todo el disco de manera que, en caso de fallo, problema o error, la tabla de particiones se recupera automáticamente desde cualquiera de dichas copias.

3. Sistema de archivos

3.1. Archivos y directorios (o carpetas)

Un archivo es un **mecanismo de abstracción** que permite referenciar de forma unívoca a un conjunto de datos.

Existen principalmente 2 tipos de archivos:

- **Los archivos regulares:** contienen información del usuario.
- **Los directorios:** Son utilizados por el SO para mantener una estructura en el sistema de archivos. En sistemas operativos con GUI son también llamados “carpetas”.

Un sistema alternativo al uso de carpetas, es el etiquetado de archivos, que permite asignar varias “carpetas virtuales” de búsqueda a un solo archivo. Las tags o etiquetas son atributos que podemos aplicarle a cualquier archivo (por ejemplo, en Windows 10) y que nos facilitan notablemente la tarea de encontrar una serie de documentos con algo en común (a modo de palabra clave), sin necesidad de que se encuentren en la misma carpeta de nuestro disco duro.

3.2. Asignación y gestión del espacio libre

El SO va a determinar mediante el sistema de archivos de qué forma se distribuye la información a nivel lógico, la cual puede distar de la forma en la que se distribuye la información a nivel físico. Así un archivo se encuentra constituido con un conjunto de bloques lógicos de memoria, es por ello que una cuestión clave va a ser el tamaño de dichos bloques, ya que, si se consideran tamaños muy grandes, se puede obtener una fragmentación interna considerable, pero si se opta por un tamaño

muy pequeño, el tiempo de acceso se puede ver penalizado. Según el autor William Stallings, en su libro Sistemas operativos (2005), una opción es optar por asignar bloques de tamaño variable.

En cuanto a la distribución lógica de los bloques, van a existir diferentes estrategias para asignar los bloques de memoria que constituyen un archivo. Éstas son:

- **Asignación adyacente:** En este caso los bloques se almacenan de forma contigua.
- **Asignación en forma de lista ligada:** Se conforma una lista enlazada con los bloques.
- **Asignación indexada:** Los punteros a los bloques de disco se almacenan en un vector de índices.

A su vez, es necesario registrar los bloques que se encuentran libres, para ello se pueden aplicar los siguientes métodos:

- **Mapa de bits:** En el mapa cada bit representa a un bloque, y tiene un valor igual a uno si el bloque está libre, y cero si está ocupado.
- **Lista enlazada:** Se trataría de mantener una lista enlazada con los bloques libres.

Adicionalmente, el SO debe llevar un control de qué bloque se encuentra defectuosos, para ignorarlos.

3.3. Protección y seguridad

Para la protección de los archivos, es posible asignarle un **dominio de protección**. Siendo esto un conjunto de parejas (objeto, derechos), la cual indica para cada objeto, el subconjunto de las operaciones que se pueden llevar a cabo en él. Por ejemplo, en Unix, el dominio queda definido mediante la identificación del usuario (uid) y la de su grupo (gid), y las operaciones que se pueden realizar son r (lectura), w (escritura), y x (ejecutar).

El acceso seguro a sistemas de archivos básicos puede estar basado en los esquemas de lista de control de acceso (Access control List, ACL) o capacidades.

3.4. Rutas y nombre de archivos

Normalmente los archivos y carpetas se organizan jerárquicamente.

La estructura de directorios suele ser jerárquica, ramificada o "en árbol", aunque en algún caso podría ser plana.

En algunos sistemas de archivos los nombres de archivos son estructurados, con sintaxis especiales para extensiones de archivos y números de versión (por ejemplo, GIT o Google Drive). En otros, los nombres de archivos son simplemente cadenas de texto y los metadatos de cada archivo son alojados separadamente.

En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada ruta (path, en inglés). La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta

viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una barra diagonal / o barra diagonal invertida \ (según el sistema operativo) y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

3.5. Archivos compartidos

Permite que un archivo aparezca de forma **simultánea** en distintos directorios, de forma que se puede resolver de dos formas:

- Enlistando los bloques del directorio en una pequeña **estructura de datos** asociadas al propio archivo.
- Mediante un enlace **simbólico**.

Estos dos tipos de enlaces reciben el nombre de enlaces duros (primer caso) y enlaces simbólicos (o soft link) en la nomenclatura de Unix/Linux. En Windows los enlaces simbólicos se denominan “accesos directos”.

4. Implementación de sistemas de archivos.

El sistema de archivos determina de qué forma se van a llevar a cabo las anteriores tareas (asignación y gestión de espacio, protección y seguridad, etc.). Existen diferentes sistemas de archivos, con diferentes características. Vamos a ver a continuación lo más utilizados actualmente:

4.1. Sistemas de archivos en GNU/Linux

El sistema de archivos nativo es actualmente **ext4** (Fourth Extended File System). Éste tiene una estructura de árbol, en la cual, el nodo principal es el directorio raíz “/”.

Las partes principales que podemos encontrar en este sistema de archivos son:

- **El superbloque**
- **Los bloques.**
- **Tabla de i-nodos.**

4.1.1. Estructura de EXT

EXT Consta de:

Un Superbloque

Al principio del sistema de archivos, existe un bloque (inicialmente era de 1024KB en EXT2) que contiene información, metadatos del propio sistema de ficheros (el tamaño del bloque, el número total de bloques...etc).

Grupo de bloques

Tienen un Backup de Superbloque, un Block Group Descriptor Table (o tabla de estructura para cada grupo de bloques, dirección de los bitmaps de bloques de i-nodos), un bitmap de bloques del grupo y un bitmap de i-nodos (del grupo).

I-nodos

Un i-nodo, es un nodo del árbol de la jerarquía de ficheros que guarda información de un archivo (ruta, tamaño, ubicación física...etc). El sistema de ficheros tiene una tabla donde se almacenan los i-nodos, los cuales pueden a su vez apuntar a ficheros o a nuevas tablas de i-nodos según vaya creciendo la necesidad de almacenamiento del sistema.

4.1.2. Características de EXT4

El sistema de archivos ext4 es capaz de trabajar con volúmenes de gran tamaño, hasta 1 ExaByte y ficheros de tamaño de hasta 16 TB.

Extents

Los extents han sido introducidos para reemplazar al tradicional esquema de bloques usado por los sistemas de archivos ext2/3. Un extent es un conjunto de bloques físicos contiguos, mejorando el rendimiento al trabajar con ficheros de gran tamaño y reduciendo la fragmentación.

Compatibilidad

Cualquier sistema ext3 existente puede ser montado como ext4 sin necesidad de cambios en el formato del disco. También es posible actualizar un sistema de archivos ext3 para conseguir las ventajas del ext4 fácilmente

Journal checksumming

Ext4 usa checksums en el registro para mejorar la fiabilidad, puesto que el journal es uno de los ficheros más utilizados en el disco. El Journal de Ext es usado para hacer una comprobación de errores inicial al arrancar el sistema y repararlos.

Desfragmentación en línea

Ext4 dispone de una herramienta que permite desfragmentar ficheros individuales o sistemas de ficheros enteros sin desmontar el disco.

4.2. Sistemas de archivos en Windows

4.2.1. FAT32

(File Allocation Table) es un sistema de ficheros que se desarrolló allá por 1977 por la empresa Microsoft y como evolución del sistema de archivos FAT. Se comenzó a utilizar en los primeros ordenadores personales de IBM en 1981 y posteriormente sería utilizado por el sistema operativo MS-DOS. Todavía a día de hoy se utiliza este sistema de ficheros en unidades de almacenamiento portátil, principalmente debido a su fácil implementación, a que está muy extendido, lo que facilita su compatibilidad y a su permisiva licencia de uso por parte de Microsoft.

Su principal inconveniente es la limitación para soportar archivos mayores de 4GB, o unidades de más de 64GB. Esto se puede solventar mediante el uso del estándar extFAT (FAT extendido) pero sigue siendo un sistema lento, inseguro y poco fiable.

4.2.2. NTFS

New Technology File System, es el sistema de ficheros más reciente del que dispone la empresa Microsoft para utilizar en su sistema operativo Windows. Se introdujo en 1993 con el sistema operativo Windows NT, y posteriormente se implementaría en los sistemas operativos Windows. NTFS ha sufrido algunas actualizaciones en cuanto a capacidad direccionamiento de espacio y ficheros.

Las características principales de NTFS son:

- Capacidad de recuperarse a partir de algunos errores relacionados con el disco automáticamente, lo que FAT32 no puede hacer.
- Compatibilidad mejorada para discos duros más grandes.
- Mejor seguridad porque puede utilizar permisos y cifrado para restringir el acceso a archivos específicos para usuarios aprobados.
- Journalising: El concepto de journalising se refiere a que si se arranca el sistema sin haberlo cerrado correctamente no es necesario hacer un chequeo ya que la recuperación sucede de forma automática a partir de su último estado. NTFS es un sistema seguro ante fallas que puede auto corregirse en casi todas las situaciones.
- Compresión: Los archivos en un volumen NTFS tienen un atributo denominado "compressed", que permite que cualquier archivo se guarde de forma comprimida con el propósito de ahorrar espacio, esa compresión es transparente para las aplicaciones. La compresión se lleva a cabo por bloques de 16 clusters y se usan "clusters virtual".

4.3. RAID

RAID es el acrónimo de Redundant Array of Inexpensive/Independent Disks (Vector redundante de discos económicos/independientes) también y provee un mecanismo barato y fácil de implementar para aumentar el rendimiento de las operaciones de disco de dos maneras posibles:

- Aumentando la velocidad de almacenamiento y lectura de los datos en disco.

- Aumentando la fiabilidad de los datos y su recuperación en caso de fallo.

El primer objetivo se logra mediante la división de la información entre distintos discos. De esta forma, al guardar en cada disco una porción de los datos, ahorramos el tiempo que tardaríamos en almacenar o recuperar dicha información completa, puesto que se efectúan las operaciones de E/S en todos los discos que componen el vector RAID de manera simultánea.

Niveles Raid:

Existe varios niveles Raid que se han ido desarrollando con el tiempo (el nivel 0, por ejemplo, no existía en las especificaciones originales). Los niveles RAID fundamentales son:

RAID 0 : Data Stripping. División de la información en dos o más discos.

RAID 1: Data Mirroring. Duplicación de la información en dos o más discos.

RAID 5:Espejo con paridad Duplica información y añade detección de errores mediante el cálculo de la paridad, la cual distribuye por todos los discos.

RAID 1+0: División de espejos. Crea un espejo de discos distribuidos.

RAID 0+1: Distribución de espejos.

Existen muchísimos más niveles RAID, algunos de ellos propietarios, pero estos son los más comunes.

5. Sistemas operativos: Gestión de dispositivos

5.1. Planificación de disco

Adicionalmente, los SO deben resolver aspectos específicos de cada dispositivo, así, por ejemplo, para las peticiones de lectura/escritura de disco, se debe establecer un orden para optimizar el tiempo **de búsqueda del cilindro**. Las técnicas que actualmente se utilizan son:

- **FCFS (First come-first served)**: Se atienden las peticiones por tiempo de llegada.
- **SSTF (Shortest seek time first)**: Se atiende la que se encuentre más cercana a la posición de la cabeza de lectura-escritura.
- **SCAN**: El brazo se mueve en todo momento desde el exterior del disco hacia su centro y viceversa, y se atienden aquellas peticiones que se encuentran durante el trayecto.
- **C-SCAN**: Se trata de una variación del anterior. Al llegar a la parte central, el brazo regresa al exterior sin resolver ninguna petición.
- **LOOK y C-LOOK**: En estos algoritmos la cabeza se mueve hasta la última solicitud de disco en una dirección, no teniendo por qué llegar hasta el final.

5.2. Planificación en otros dispositivos

Los SSOO también disponen de una serie de planificadores para el resto de dispositivos. Así, por ejemplo, en Android para las unidades de estado sólido podemos encontrar:

- **Noop**: Es similar a FCFC.
- **Deadline**: Utiliza el algoritmo Round Robin.

6. Gestión de dispositivos en sistemas operativos actuales

6.1. GNU/Linux

En GNU/Linux casi todos los dispositivos de E/S se representan como **ficheros especiales**. Una ventaja adicional de esto es que las reglas usuales de **protección** de archivos se aplican de manera automática a los dispositivos de E/S. Estos ficheros especiales, a su vez, se van a dividir en **ficheros especiales de bloque**, y **ficheros especiales de carácter**.

Cada dispositivo de E/S tiene asociado un archivo especial en el directorio / dev gestionado por el sistema de archivos, que se lee y se escribe igual que el resto de archivos. Para acceder a un dispositivo, basta con realizar peticiones de lectura o escritura a su archivo especial asociado.

Los archivos especiales se dividen en:

- **Archivos de bloque**. Están formados por una secuencia de bloques numerados, y se caracterizan por que es posible acceder a cada bloque por su dirección.
- **Archivos de carácter**. Se utilizan para dispositivos que producen o reciben flujos de caracteres, como teclados, impresoras, ratones, tarjetas de red, etc.

Cada archivo especial tiene asociado un manejador de dispositivo, el cual tiene un número principal que sirve para identificarlo. Si un manejador de dispositivo puede manipular varios dispositivos, cada dispositivo tiene un número secundario que le identifica. Por tanto, cada pareja de números principal y secundario identifica a un determinado dispositivo.

La E/S se implementa mediante una colección de manejadores de dispositivo, uno por cada tipo de dispositivo. Su función consiste en aislar al resto del sistema de las peculiaridades del hardware. Cuando un proceso de usuario accede a un archivo especial, el sistema determina los números principal y secundario, que son utilizados como parámetros para acceder a las rutinas de manejo del dispositivo correspondiente.

Los manejadores de dispositivo pueden estar incluidos estáticamente dentro del núcleo o incluidos en módulos que pueden ser cargados dentro del núcleo en tiempo de ejecución.

La entrada/salida puede realizarse de dos maneras:

- Utilizando DMA. Es el método más rápido, pero no permite que los procesos suspendidos por la operación de E/S puedan ser intercambiados. Además, el dispositivo se paraliza junto al proceso mientras dure la transferencia, quedando inasequible para otros procesos.
- Utilizando los buffers del sistema. Se pueden usar dos clases de buffers:

Caché de disco. Adecuada para dispositivos de bloque. Se trata de un conjunto fijo de bloques recientemente leídos que se guardan en memoria para aumentar el rendimiento. Si un bloque no está en la caché de disco, se lee del disco hacia la caché y de ahí a la memoria principal, y viceversa en el caso de escritura. Para cargar un bloque en una caché totalmente ocupada se plantea una situación muy parecida a la paginación, y se resuelve con algoritmos similares. La transferencia de datos entre la caché de disco y el espacio de usuario del proceso se produce mediante DMA. Ante la posibilidad de un fallo del sistema, es necesario que los bloques críticos que hayan sido modificados permanezcan el menor tiempo posible en la caché y se transfieran al disco, ya sea a intervalos regulares o justo después de modificarse la caché.

Cola de caracteres. Adecuada para dispositivos de carácter. El dispositivo de E/S escribe en una cola de caracteres de la que lee el proceso, o también, el proceso escribe y el dispositivo lee de ella. De esta manera, las colas de caracteres sólo pueden ser leídas una vez. A medida que se lee cada carácter, éste es destruido. Este mecanismo es distinto al de la caché de buffers, donde se puede leer varias veces.

6.2. Windows

En el caso de Windows, la gestión de los dispositivos se realiza a través de la **manipulación de objetos**, así para cada dispositivo se carga un objeto driver, y al menos, un objeto dispositivo, los cuales pueden crearse y cargarse cuando se conecta el dispositivo, lo que permite que se puedan conectar **dispositivos en caliente**.

El administrador de E/S de Windows está diseñado para proporcionar **un marco de trabajo extensivo y flexible**, que permita manejar una amplia variedad de dispositivos, y la instalación de controladores (plug-and-play). Para facilitar la tarea de codificación de drivers, Microsoft ha definido el **modelo WDM**, Windows Driver Model.

El gestor plug and play permite que los dispositivos compatibles sean inmediatamente reconocidos una vez conectados. Para ello, el gestor pide al dispositivo que se identifique, y con esta información carga en memoria el manejador de dispositivo apropiado.

La entrada/salida de dispositivos se maneja a través de las APIs del subsistema Win32, el cual se relaciona con el hardware a través de la capa de abstracción de hardware (HAL). Esta capa es una interfaz entre el hardware y el resto del sistema operativo que se caracteriza por lo siguiente:

- Está implementada como una biblioteca de enlace dinámico (dll) y es responsable de proteger y aislar el resto del sistema de las especificaciones del hardware.
- Presenta los dispositivos al sistema operativo de manera homogénea a través de un conjunto de funciones bien definidas. Estas funciones son llamadas tanto desde el sistema operativo como

desde los propios manejadores de dispositivo, permitiendo a éstos adaptarse a distintas arquitecturas de E/S sin tener que ser modificados en gran medida.

- Si Windows es portado a una nueva arquitectura de procesador el HAL debe ser reescrito, pero el resto del sistema simplemente debe ser recompilado.
- También suministra la interfaz para el multiprocesamiento simétrico, transformando cada procesador físico en un procesador virtual idéntico al resto de procesadores.
- Al HAL sólo pueden acceder componentes del Executive de Windows y nunca los programas en modo usuario. El HAL también intenta ser el único software dentro del sistema que se comunica con el hardware, aunque existe un pequeño número de llamadas de los manejadores de dispositivo y del núcleo que interactúan directamente con el hardware.

7. Conclusión

Aunque la gestión del disco es prácticamente transparente e independiente del dispositivo, tanto para el usuario y para el programador, esto se logra gracias a la labor del sistema operativo, que es capaz de gestionar eficazmente los dispositivos y los ficheros contenidos en ellos.

Los sistemas de archivos siguen evolucionando, el sistema GPT por ejemplo aún está implantándose; se buscan soluciones distribuidas, de mayores capacidades, mayor seguridad, con capacidad de cifrado y garantizando permisos robustos, cuotas de disco por usuario y algoritmos más eficientes para los nuevos soportes de memoria auxiliar (NVMe).

7.1. Relación del tema con el currículo

Este tema es aplicado en el aula en los módulos profesionales siguientes, con las atribuciones docentes indicadas (PES/SAI):

Grado Medio

- Sistemas operativos monopuesto (SMR) (PES/SAI)

Grado Superior

- Sistemas informáticos (DAM / DAW) (PES/SAI)
- Implantación de sistemas operativos (ASIR) (PES/SAI)

8. Bibliografía

- De Anasagasti, Miguel. "Fundamentos de la Computadora" 9^aed 2004 Edt. Paraninfo
- Patterson D.A. y Hennessy JL. "Estructura y diseño de computadoras: la interfaz hardware/software" 4^a Ed. (2005) Edt McGraw-Hill
- Prieto A, Lloris A, Torres JC. "Introducción a la Informática" 4^aed. (2006) Edt. McGraw-Hill
- Stallings W. "Organización y Arquitectura de Computadoras" (2006) 5^a Ed. Edt. Prentice-Hall

- Ramos A, Ramos MJ y Viñas S “Montaje y Mantenimiento de Equipos” (2012). Edt McGraw-Hill
- Jiménez Cembreras, Isabel Mª “Sistemas Informáticos” 2^aEd (2018) Edt. Garceta
- Informáticos” 2^aEd (2018) Edit. Garceta