

Explotación y administración de sistemas operativos monousuario y multiusuario.

## TEMA 20

---

ABACUS NT

**1. Introducción**

**2. Clasificación de los Sistemas Operativos**

- 2.1. Segundo el número de usuarios
- 2.2. Segundo el número de procesos
- 2.3. Segundo el número de procesadores del sistema informático
- 2.4. Segundo el tiempo de respuesta

**3. Sistema operativo monousuario: Android**

- 3.1. Máquina virtual java
- 3.2. Gestión de Memoria en Android.
  - 3.2.1. Jerarquía de Memoria
  - 3.2.2. Paginación de la memoria.
  - 3.2.3. Algoritmo de Reemplazo de página.
- 3.3. Gestión de Procesos en Android.
  - 3.3.1. Procesos:
  - 3.3.2. Tipos de procesos en Android.
- 3.4. Planificación de procesos
- 3.5. Gestión de Usuarios en Android

**4. Sistema Operativo Windows Server**

- 4.1. Arquitectura de Windows Server
- 4.2. Executive
- 4.3. Capa de abstracción de hardware – HAL
- 4.1. Núcleo
- 4.2. Gestión de procesos
  - 4.2.1. Procesos e hilos
  - 4.2.2. Algoritmo de planificación
- 4.3. Gestión de Memoria
- 4.4. Sistema de archivos
- 4.5. Active Directory y seguridad del sistema
- 4.6. Hipervisor
- 4.7. Características Avanzadas

**5. Conclusión.**

- 5.1. Sistema educativo

**6. Bibliografía**

## 1. Introducción

Un Sistema Operativo (SO), de manera general está representado por el conjunto de programas que se ejecutan dentro de algún equipo que cumple en primera instancia la función de permitir la interacción del usuario con la máquina, para tal fin es necesario que el SO, pueda gestionar todos los recursos del equipo, lo cual involucra a los elementos hardware (monitor, teclado, mouse, unidades de almacenamiento, impresoras, etc.) así como al software del equipo.

Los equipos informáticos (PC, laptops, servidores, tablets, smartphones y cualquier otro) tienen y necesitan un SO que permitan su normal funcionamiento, administrando los recursos del equipo, coordinando el funcionamiento del hardware y organizando los archivos y directorios en los dispositivos de almacenamiento.

Dentro de la amplia variedad de sistemas operativos existentes, podemos diferenciar entre sistemas operativos monousuario, esto es gestionados por un único usuario en un momento dado y por sistemas multiusuario, accedidos por múltiples usuarios en un mismo momento.

Ejemplos de sistemas monousuario son: **ChromeOS, Android o MS-DOS**; Sistemas operativos multiusuario son: **Linux, Windows Server o Unix**. Algunos sistemas operativos no están realmente definidos al respecto, como es el caso de **Windows 10**, que dependiendo de la versión tiene más o menos restringidos los servicios del sistema, pero es fácilmente convertible en un servidor multiusuario para servicios específicos.

## 2. Clasificación de los Sistemas Operativos

Para hacer una clasificación de los Sistemas Operativos según su tipo, hay que tener en cuenta una serie de parámetros:

- Número de usuarios
- Número de procesos
- Número de procesadores
- Tiempo de respuesta

### 2.1. Segundo el número de usuarios

- **Monousuario.**

Sólo un usuario trabaja con un ordenador. En este sistema todos los dispositivos de hardware están a disposición de dicho usuario y no pueden ser utilizados por otros hasta que éste no finalice su sesión.

Los SO monousuario más claros son: **ChromeOS, Android o MS-DOS**.

- **Multiusuario.**

En este sistema, varios usuarios pueden utilizar simultáneamente los recursos del sistema. Pueden compartir, sobre todo, los dispositivos externos de almacenamiento y los periféricos de salida, fundamentalmente impresoras.

Ejemplos de sistemas multiusuario son Linux, Novell, todos los sistemas Windows desde la aparición de Windows NT, (Windows XP, 7, 8 y 10) etc.

## 2.2. Segundo criterio de clasificación: Según el número de procesos

Esta clasificación se hace atendiendo al número de programas o procesos que puede realizar simultáneamente el ordenador o sistema informáticos:

- **Monoprogramación** o monotarea.

En este caso, el sistema solamente puede ejecutar un programa a la vez. De esta forma, los recursos del sistema estarán dedicados al programa hasta que finalice su ejecución.

Esto no impide que el sistema pueda ser multiusuario; es decir, varios usuarios pueden intentar ejecutar sus programas en el mismo ordenador, pero de forma sucesiva. Para ello, se tienen que establecer las correspondientes colas o prioridades en la ejecución de los trabajos.

En este sistema la atención del procesador estará dedicada a un solo programa hasta que finalice.

Un claro ejemplo de sistema monoprogramación es el MS-DOS.

En este caso un programa utilizará parte de todos los procesadores. Si llega otro nuevo programa para ser ejecutado, se utilizarán también todos los procesadores, y así hasta su total utilización. De esta forma trabajarán, todos, pero es evidente que lo harán a bajo rendimiento.

- **Multiprogramación** o multitarea.

Con estos sistemas se pueden ejecutar varios programas o procesos concurrentemente. Para ello la CPU compartirá el tiempo de uso del procesador entre los diferentes programas que se van a ejecutar.

Así, todos los procesos tardarán individualmente más tiempo en ejecutarse; pero, comparándolo con la monoprogramación, el tiempo medio de espera será mucho menor.

Como ejemplos de sistema en multiprogramación podemos hablar de Windows en cualquier versión desde Windows NT o XP, de Linux, Android o en general de cualquier sistema operativo actual.

## 2.3. Tercer criterio de clasificación: Según el número de procesadores del sistema informático

Esta clasificación atiende a que el sistema u ordenador cuente con uno o varios procesadores para realizar los procesos:

- **Monoproceso**. En este caso, el ordenador consta de un único procesador.

Todos los trabajos pasarán por él.

Ejemplo de SO monoproceso es MS-DOS.

- **Multiproceso.**

El ordenador cuenta con varios procesadores. Estos procesadores pueden actuar de dos formas diferentes:

- **Multiproceso simétrico (SMP):**

el sistema utilizará la totalidad de los procesadores para realizar todas las tareas. Cada

- **Multiproceso asimétrico (AMP):**

Existen ordenadores que irán saturando de trabajo a sus procesadores poco a poco. Con la primera tarea utilizará el primer procesador; si entra otra tarea, se utilizará lo que reste de potencia del primer procesador y lo necesario del segundo. Los demás procesadores se irán utilizando de forma sucesiva. De esta forma pueden quedar procesadores inactivos.

Son SO multiproceso (según el ordenador y teniendo en cuenta otras muchas condiciones) los da la familia Windows Server, Windows XP; Vista y 7, Windows 8 y 10, y muchas versiones de UNIX/Linux y Novell.

## 2.4. Segundo el tiempo de respuesta

Esta clasificación se hace teniendo en cuenta el tiempo que tarda el sistema en obtener los resultados después de lanzar un programa a ejecución:

- **Tiempo real.** La respuesta es inmediata (o casi inmediata) tras lanzar un proceso.
- **Tiempo compartido.** Cada proceso utilizará fracciones de tiempo de ejecución de la CPU hasta que finalice. En este caso, parece que el usuario dedica la CPU exclusivamente para él; pero esto no es cierto, ya que, aunque el usuario no lo perciba, la CPU está dedicada a varios procesos a la vez. Todos los SO multiusuario ofrecen a los usuarios un tiempo de respuesta compartido.

## 3. Sistema operativo monousuario: Android

Android es un sistema operativo monousuario desarrollado por Google, basado en el Kernel 2.5 de Linux y otro software de código abierto. Está orientado a la plataforma ARM aunque existen versiones para x86 (Chrome OS).

Android es software libre bajo licencia Apache y actualmente (año 2020) es el sistema operativo móvil más utilizado en el mundo.

Su nombre proviene de la famosa novela de Philip K. Dick *Do Androids Dream of Electric Sheep? (¿sueñan los androides con ovejas eléctricas?)* que fue adaptada en la película *Blade Runner* de Ridley Scott.

La compañía que lo desarrolló fue Android Inc, apoyada económicamente por Google y comprada por este en 2005, pero no fue presentado hasta el año 2007 tras la fundación del

Open Handset Alliance que engloba numerosos fabricantes de hardware relacionados con la telefonía móvil.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan sobre un Middleware que las abstrae de la complejidad de las comunicaciones del sistema y de la máquina en particular, mediante un framework Java.

Bajo esta máquina virtual Java, se encuentra el núcleo del sistema escrito en lenguaje C y que incorpora una potente biblioteca que incluye un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL, un motor de renderizado WebKit (Safari), un motor gráfico SGL - SSL y una biblioteca estándar de C (Bionic).

### **3.1. Máquina virtual java**

Uno de los elementos clave de Android es la máquina virtual Java. En lugar de utilizar una tradicional máquina virtual Java (VM), tales como Java ME (Java Mobile Edition), Android utiliza su propia máquina virtual personalizada diseñada para asegurar que la multitarea se ejecuta de manera eficiente en un único dispositivo.

Inicialmente se utilizó la máquina virtual Dalvik con la librería de ejecución JIT (Just In Time) que traducía a bytecode Java cualquier aplicación en el momento de su ejecución.

En la versión 4.4 de Android se introdujo la máquina ART (Android RunTime) que compila la aplicación a Java bytecode en el momento de su instalación; a partir de la versión 5 ART reemplazó por completo a Dalvik.

Las aplicaciones (Apps) en Android se distribuyen por tanto en archivos .APK (Android Application PackAge, similar al JAR del Java de Oracle) y son compiladas en el momento de su instalación por ART a Java bytecode.

La máquina virtual Java de Android no gestiona directamente la memoria o los procesos, sino que utiliza el núcleo de Linux subyacente para manejar la planificación de procesos y la gestión de la memoria.

Android funciona en modo multitarea ejecutando múltiples instancias de la máquina virtual java, una por cada tarea, lo que crea un entorno muy confiable, similar a un sandbox.

### **3.2. Gestión de Memoria en Android.**

La plataforma de Android se basa en la premisa de que la memoria disponible es una pérdida de memoria, de modo que intenta utilizar toda la memoria disponible en todo momento.

Android Runtime (ART) y la máquina virtual Dalvik usan las funciones de paginación y mapeo de memoria (mmapping) para administrar la memoria.

### 3.2.1. Jerarquía de Memoria

La jerarquía de Memoria en Android se basa en la gestión de tres niveles: RAM, zRAM y ROM.

- La memoria RAM es la memoria principal del dispositivo y se gestiona mediante paginación.
- La zona libre de RAM se utiliza como una memoria de intercambio SWAP en la que se almacenan datos y programas de forma comprimida, aumentando así su aprovechamiento en término de capacidad, de ahí el nombre de zRAM, para diferenciarla.
- La memoria ROM está compuesta por dispositivos extremadamente lentos compuestos por memoria EEPROM Flash Integrada y opcionalmente una tarjeta SD.
- Existe un nivel cero en esta jerarquía compuesto por la memoria caché y registros del microprocesador, pero Android no los gestiona directamente.

### 3.2.2. Paginación de la memoria.

La memoria RAM está dividida en páginas. Por lo general, cada página tiene 4 KB de memoria.

Las páginas se consideran **libres o usadas**. Las páginas libres son memoria RAM sin usar. Las páginas usadas son memoria RAM que el sistema está utilizando de manera activa y se agrupan en las siguientes categorías:

- Almacenamiento en caché: Memoria respaldada por un archivo de almacenamiento (por ejemplo, código o archivos asignados a la memoria). Hay dos tipos de memoria caché:
  - Privada: Es propiedad de un proceso y no se comparte.
  - Compartida: Es utilizada por varios procesos.
- Anónima: Memoria no respaldada por un archivo en almacenamiento (por ejemplo, asignada por mmap() con la marca MAP\_ANONYMOUS).

Las páginas a su vez pueden ser **Limpias o Sucias**: Las páginas limpias contienen una copia exacta de un archivo (o parte de un archivo) que existe en el almacenamiento. Una página limpia se convierte en una página sucia cuando ya no contiene una copia exacta del archivo (por ejemplo, del resultado de la operación de una app).

Es posible borrar las páginas limpias porque siempre se pueden volver a generar con los datos del almacenamiento. En cambio, no se pueden borrar las páginas sucias, ya que se perderían los datos.

### 3.2.3. Algoritmo de Reemplazo de página.

El algoritmo de reemplazo de página es el mismo que utiliza el núcleo Linux 2.6 y que se denomina “Page Frame Reclamation Algorithm”. Es una modificación del algoritmo LRU pero basado en listas y con prioridad para los procesos activos.

Las páginas de memoria en uso se dividen en dos listas por cada zona de memoria, una para aplicaciones en uso y otra para aplicaciones inactivas. Las páginas de memoria de la lista de inactivas pueden además ser marcadas como swappables o descartables dependiendo del tiempo de inactividad y previsión de uso.

El algoritmo PFRA Sólo se ejecuta cuando la memoria empieza a escasear, momento en el cual selecciona todas las páginas marcadas como “descartables/limpias” y a continuación se seleccionan las páginas menos usadas recientemente (LRU) de la lista de aplicaciones inactivas.

Este algoritmo se ejecuta de forma incremental, intentando liberar el máximo de memoria, pero cargando en zRAM/swap páginas inactivas sucias, volcando a memoria masiva (limpiando) solo en las últimas iteraciones si no se alcanza liberar la memoria deseada.

La implementación en Android de este algoritmo la realiza el demonio kswapd que se ejecutará automáticamente al alcanzarse un umbral mínimo de memoria disponible y dejará de utilizarse cuando se alcance un umbral máximo determinado también de antemano.

Si tras la ejecución de kswapd aún no hay suficiente memoria para la carga de una determinada aplicación en memoria, se ejecutará un asistente de limpieza (LMK) que sacará de memoria procesos activos, según un nivel de prioridad asignado, comenzando con los procesos en segundo plano menos usados, y siguiendo incluso con la app de inicio, servicios, apps en primer plano, procesos del sistema, etc.

### 3.3. Gestión de Procesos en Android.

#### 3.3.1. Procesos:

Con frecuencia la documentación de Android se refiere a la gestión de procesos como **ciclo de vida** de Android.

Android implementa la **multitarea** mediante la ejecución de **múltiples hilos** para cada proceso; el proceso más simple consta de un sólo hilo principal que sólo es lanzado en el caso de que no exista ninguna instancia del proceso en ejecución.

De igual forma un proceso puede utilizar múltiples hilos, uno por cada subprocesso. También se da la situación en que un proceso comparte subprocessos con otros, lanzando para ello el sistema los hilos necesarios.

#### 3.3.2. Tipos de procesos en Android.

El tipo de proceso en Android determina el comportamiento del planificador respecto a este: no es lo mismo un proceso “activo” y “visible” cuya finalización por parte del sistema tendría una **experiencia muy negativa** para el usuario, que por ejemplo un servicio del sistema.

Android, por tanto, clasifica los procesos como:

- **Activos:** Se están ejecutando y además son requeridos para la interacción con el usuario, bien sean procesos en primer plano o servicios dando soporte a éstos.

- **Visibles:** Un nivel por debajo de Activo, ya que un proceso visible estaría en segundo plano.
- **Servicio:** Procesos que no son visibles de forma directa, pero dan un servicio importante al usuario, por ejemplo mantener una conexión a Internet. Estos procesos no deben ser finalizados a no ser que su mantenimiento suponga una merma para un proceso activo.
- **Segundo Plano:** Procesos en ejecución, pero no visibles, a los que el sistema intentará dotar de recursos siempre que haya disponibles.
- **Vacíos:** Es una forma rápida de inicialización: por ejemplo un proceso que intenta ser lanzado pero el liberador de memoria está ocupado haciéndole hueco.

### 3.4. Planificación de procesos

La planificación en Android es **expropiativa** lo cual debe ser tenido en cuenta por el programador, ya que si no se definen bien los diferentes componentes de la aplicación esta puede ser finalizada mientras está efectuando una tarea relevante para el sistema o para el usuario.

El algoritmo utilizado por el planificador es **Round-Robin con prioridad** basada en una **jerarquía de importancia** según el tipo de proceso (visto anteriormente) y otros factores:

El sistema mantiene una lista pseudo-LRU (Last Recently Used) para evitar que los procesos se apropien de los recursos, pero ordenada según prioridades. La prioridad de un proceso se puede incrementar si este es un subproceso de mayor prioridad, es decir, la prioridad de un conjunto de procesos interdependientes se iguala a la prioridad del componente más prioritario del conjunto.

Cuando un proceso pasa al estado de espera o de finalización, este se almacena en caché (zRAM). El sentido de almacenar también los procesos finalizados es acelerar su carga en RAM si se vuelven a solicitar.

Android asegura la respuesta de cualquier app lanzada por el usuario, deteniendo y matando a los procesos que impiden la fluidez y liberando recursos para las aplicaciones más prioritarias.

### 3.5. Gestión de Usuarios en Android

Android sólo puede ser accedido por un usuario en un momento dado, es decir por el usuario principal.

En algunos sistemas (Samsung, Xiaomi) es posible mantener usuarios adicionales mediante la adición de capas de software sobre el sistema operativo que emule el uso por varios usuarios, por ejemplo para control parental o para tener un doble uso del teléfono en casa y en el trabajo.

Sin embargo en el sistema sólo hay en realidad un único usuario, al que se le añaden o restringen permisos y visores sobre aplicaciones.

Este usuario al que se le ligan las distintas cuentas de proveedores (Google, Xiaomi, Amazon, etc.) viene con los permisos restringidos por defecto.

Si es necesario dar mayores niveles de prioridad al usuario debemos acceder a la configuración del teléfono para activar los permisos completos. A este proceso se le suele llamar ruteo del teléfono, porque al hacerlo activamos los permisos completos del usuario root en el kernel Linux. Con frecuencia esta escalada de privilegios está restringida y a apareja la pérdida de la garantía y otras ventajas de la marca.

## 4. Sistema Operativo Windows Server

Windows Server es un sistema Operativo producido por Microsoft diseñado para servidores, el cual está construido sobre el núcleo de Windows NT.

La versión más extendida hoy en día (año 2020) es la **Windows Server 2008**, si bien existe una versión de reciente lanzamiento: **Windows Server 2019**.

### 4.1. Arquitectura de Windows Server

La arquitectura de Windows Server es altamente modular y se basa en dos capas principales:

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

**Modo núcleo:** Tiene acceso total a la memoria del sistema y los dispositivos externos. Los núcleos de los sistemas operativos de esta línea son todos conocidos como **núcleos híbridos**, aunque hay que aclarar en realidad es esencialmente un **núcleo monolítico que está estructurado al estilo de un micronúcleo**. La arquitectura dentro del modo núcleo se compone de lo siguiente:

- Un núcleo híbrido.
- Una Capa de Abstracción de Hardware (HAL).
- Controladores o drivers.
- Executive: Sobre el cual son implementados todos los servicios de alto nivel.

El modo núcleo de la línea de Windows Server está compuesto por subsistemas capaces de pasar peticiones de E/S a los controladores apropiados usando el gestor de E/S. Dos subsistemas crean la capa del modo usuario:

- el **subsistema de Entorno** (ejecuta aplicaciones escritas para distintos tipos de sistemas operativos), y el
- **subsistema Integral** (maneja funciones específicas de sistema de parte del subsistema de Entorno). El modo núcleo en Windows Server tiene acceso total al hardware y a los recursos del sistema de la computadora a la vez que impide a los servicios del modo usuario y las aplicaciones acceder a áreas críticas del sistema operativo a las que no deberían tener acceso.

El **Executive** se relaciona con todos los subsistemas del modo usuario. Se ocupa de la entrada/salida, la gestión de objetos, la seguridad y la gestión de procesos.

El núcleo se sitúa entre la Capa de Abstracción de Hardware (**HAL**) y el Executive para proporcionar sincronización multiprocesador, hilos y programación y envío de interrupciones, y envío de excepciones.

El núcleo también es responsable de la inicialización de los controladores de dispositivos al arrancar.

Hay **tres niveles de controladores** en el modo núcleo:

- controladores de alto nivel,
- controladores intermedios y
- controladores de bajo nivel.

El Modelo de controladores de Windows (en inglés Windows Driver Model, **WDM**) se encuentra en la capa intermedia y fue diseñado principalmente para mantener la compatibilidad en binario y en código fuente entre Windows 98 y Windows Server. Los de más bajo nivel también son un legado de los controladores de dispositivos de Windows NT que controlan directamente un dispositivo, o pueden ser un bus hardware PnP.

## 4.2. Executive

El Executive se relaciona con todos los subsistemas del modo usuario. **Se encarga de la Entrada/Salida, la gestión de objetos, la seguridad y la gestión de procesos.** Está dividido informalmente en varios subsistemas, entre los que se encuentran el Gestor de Caché, el Gestor de Configuración, el Gestor de Entrada/Salida, las Llamadas a Procedimientos Locales, el Gestor de Memoria, el Gestor de Objetos, la Estructura de Procesos, y el Monitor de Referencias de Seguridad. Todos juntos, los componentes pueden ser llamados **Servicios Executive** (nombre interno Ex). Los Servicios del Sistema (**nombre interno Nt**), por ejemplo las llamadas al sistema, se implementan en este nivel también, excepto unas pocas que son llamadas directamente dentro de la capa del núcleo para obtener un mejor rendimiento.

Los subsistemas executive son los siguientes:

El **Gestor de Objetos** (nombre interno Ob) es un subsistema especial del Executive por el cual todos los otros subsistemas del Executive, especialmente las llamadas al sistema, deben pasar para obtener acceso a los recursos de Windows 2000. Esto hace que sea esencialmente un servicio de infraestructuras de gestión de recursos.

**Controlador de Caché** (en inglés Cache Controller, nombre interno Cc): está estrechamente relacionado con el Gestor de Memoria, el Gestor de Entrada/Salida y los controladores de Entrada/Salida para proporcionar una caché común para archivos frecuentes de E/S. El Gestor de Caché de Windows opera únicamente con bloques de archivo (más que con bloques de dispositivo), para realizar operaciones consistentes entre archivos locales y remotos, y asegurar un cierto grado de coherencia con las páginas en

memoria de los archivos, ya que los bloques de caché son un caso especial de las páginas en memoria y los fallos caché son un caso especial de los fallos de página.

**Gestor de Configuración** (en inglés Configuration Manager, nombre interno Cm): implementa el registro de Windows.

**Gestor de E/S** (en inglés I/O Manager, nombre interno Io): permite a los dispositivos comunicarse con los subsistemas del modo usuario. Se ocupa de traducir los comandos de lectura y escritura del modo usuario a IRPs de lectura o escritura que envía a los controladores de los dispositivos. También acepta peticiones de E/S del sistema de archivos y las traduce en llamadas específicas a los dispositivos. Puede incorporar controladores de dispositivo de bajo nivel que manipulan directamente el hardware para leer la entrada o escribir una salida. También incluye un gestor de caché para mejorar el rendimiento del disco guardando las peticiones de lectura y escribiendo a disco en segundo plano.

**Llamada a Procedimientos Locales** (en inglés Local Procedure Call (LPC), nombre interno Lpc): proporciona comunicación entre procesos a través de puertos con conexión semántica. Los puertos LPC son usados por los subsistemas del modo usuario para comunicarse con sus clientes, por los subsistemas Executive para comunicarse con los subsistemas del modo usuario, y como base para el transporte local para MSRPC.

**Gestor de Memoria** (en inglés Memory Manager, nombre interno Mm): gestiona la memoria virtual, controlando la protección de memoria y el paginado de memoria física al almacenamiento secundario, e implementa un gestor de memoria física de propósito general. También implementa un parser de Ejecutables Portables (en inglés, Portable Executable, PE) que permite a un ejecutable ser mapeado o liberado en un paso único y atómico.

**Estructura de Procesos** (en inglés Process Structure, nombre interno Ps): gestiona la creación y finalización de procesos e hilos, e implementa el concepto de trabajo (job), un grupo de procesos que pueden ser finalizados como un conjunto, o pueden ser puestos bajo restricciones compartidas (como un máximo de memoria asignada, o tiempo de CPU).

**Gestor de PnP** (en inglés PnP Manager, nombre interno Pnp): gestiona el servicio de Plug and Play, mantiene la detección de dispositivos y la instalación en el momento del arranque. También tiene la responsabilidad de parar y arrancar dispositivos bajo demanda: esto puede suceder cuando un bus (como un USB o FireWire) detecta un nuevo dispositivo y necesita tener cargado un driver para acceder a él. En su mayor parte está implementada en modo usuario, en el Servicio Plug and Play, que gestiona las tareas, a menudo complejas, de instalación de los controladores apropiados, avisando a los servicios y aplicaciones de la llegada de nuevos servicios, y mostrando el GUI al usuario.

**Gestor de Energía** (en inglés Power Manager, nombre interno Po): se ocupa de los eventos de energía (apagado, modo en espera, hibernación, etc.) y notifica a los controladores afectados con IRPs especiales (IRPs de Energía).

**Monitor de Referencias de Seguridad** (en inglés Security Reference Monitor (SRM), nombre interno Se): es la autoridad principal para hacer cumplir las reglas del subsistema de seguridad integral.<sup>6</sup> Determina cuándo se puede acceder a un objeto o recurso, a través

del uso de listas de control de acceso (en inglés Access Control List, ACL), que están formadas por entradas de control de acceso (en inglés Access Control Entries, ACE). Los ACEs contienen un identificador de seguridad (en inglés, Security Identifier, SID) y una lista de operaciones que el ACE proporciona a un grupo de confianza — una cuenta de usuario, una cuenta de grupo, o comienzo de sesión — permiso (permitir, denegar, o auditar) a ese recurso.

### 4.3. Capa de abstracción de hardware – HAL

La Capa de Abstracción de Hardware, o HAL (en inglés Hardware Abstraction Layer), es una capa que se encuentra entre el hardware físico de la computadora y el resto del sistema operativo. Fue diseñado para ocultar las diferencias de hardware y por tanto proporciona una plataforma consistente en la cual se puedan ejecutar las aplicaciones. La HAL incluye código dependiente del hardware que controla las interfaces de E/S, controladores de interrupciones y múltiples procesadores.

En particular, la "abstracción hardware" no implica abstraer el conjunto de instrucciones, que generalmente se engloban bajo el concepto más amplio de portabilidad. La abstracción del conjunto de instrucciones, cuando es necesario (como para gestionar varias revisiones del conjunto de instrucciones del x86, o la emulación de un coprocesador matemático inexistente), es realizada por el núcleo.

A pesar de su propósito y su posición dentro del diseño de la arquitectura, el HAL no es una capa que se encuentre completamente debajo del núcleo de la misma forma que el núcleo se encuentra debajo del Executive: todas las implementaciones conocidas del HAL dependen de alguna manera del núcleo, o incluso del Executive. En la práctica, esto significa que el núcleo y las variaciones del HAL se distribuyen conjuntamente, generados específicamente para trabajar juntos.

### 4.1. Núcleo

El núcleo del sistema operativo se encuentra entre el HAL y el Executive y proporciona sincronización multiprocesador, hilos y envío y planificación de interrupciones, gestión de interrupciones y envío de excepciones. También es responsable de la inicialización de controladores de dispositivos que son necesarios en el arranque para mantener el sistema operativo funcionando. Esto es, el núcleo realiza casi todas las tareas de un micronúcleo tradicional.

El núcleo a menudo interactúa con el gestor de procesos. El nivel de abstracción es tal que el núcleo nunca llama al gestor de procesos: únicamente se permite lo contrario.

### 4.2. Gestión de procesos

Los procesos en Windows Server se implementan como objetos y son accedidos mediante servicios de objetos. Un proceso NT tiene asociados varios hilos que se ejecutan en su espacio de direccionamiento. Así mismo, el diseño de los procesos de Windows Server está dirigido por la necesidad de dar soporte a varios entornos de sistemas operativos. Por lo cual tal como lo indican **Alvear Luis y Álvarez David (2012)** "la estructura nativa de los

procesos y de los servicios que brinda el núcleo de NT 6.1 es relativamente simple y de propósito general, permitiendo a cada subsistema emular la estructura y la funcionalidad particular de los procesos de un sistema operativo."

Las características más importantes de los procesos de NT son las siguientes:

- Los procesos de Windows Server se implementan como objetos.
- Un proceso ejecutable puede tener uno o más hilos.
- Los objetos proceso y los objetos hilo tienen capacidades predefinidas de sincronización.
- El núcleo de Windows Server no conserva ninguna relación entre los procesos que crea, incluyendo las relaciones padre-hijo.

Cada proceso en Windows Server tiene una señal de acceso que le sirve para cambiar sus propios atributos. También tienen que ver con el proceso una serie de bloques que definen el espacio de direcciones virtuales asignado. El proceso no puede modificar directamente estas estructuras, sino que debe depender del administrador de memoria virtual, quien le proporciona al proceso un servicio de asignación de memoria. Finalmente, el proceso incorpora una tabla de objetos, con los descriptores de otros objetos que conoce.

#### 4.2.1. Procesos e hilos

Los hilos son similares a los procesos ya que ambos representan una secuencia simple de instrucciones ejecutada en paralelo con otras secuencias.

La diferencia más significativa entre los procesos y los hilos, es que los primeros son típicamente independientes, llevan bastante información de estados, e interactúan sólo a través de mecanismos de comunicación dados por el sistema.

Por otra parte, los hilos generalmente comparten la memoria, es decir, acceden a las mismas variables globales o dinámicas, por lo que no necesitan costosos mecanismos de comunicación para sincronizarse.

Los estados de un hilo/proceso en Windows Server son los siguientes:

- **Alerta:** Seleccionado como el siguiente hilo a ser ejecutado en un procesador dado. Operación (planificación) previa a la invocación de un cambio de contexto.
- **Bloqueado:** Espera a que un objeto de sincronización pase a una situación de marcado ("signaled") que indique la llegada del evento de espera.
- **Listo:** El hilo listo para ejecutarse
- **Ejecución:** El hilo está ejecutando
- **Standby:** El hilo ha sido seleccionado para ser ejecutado en un procesador particular.
- **Waiting:** Se encuentra bloqueado esperando un evento
- **Terminated:** Finalización del hilo .



Fig. Transiciones entre estados de **hilos/procesos** en Windows Server

#### 4.2.2. Algoritmo de planificación

Desde Windows NT, se ha planificado threads por prioridades, siguiendo un esquema parecido al de UNIX:

Contando con 32 niveles de prioridad divididos en dos clases. Los 16 niveles superiores, de prioridades estáticas, constituyen la clase de tiempo real. Los 16 inferiores (clase variable), donde se ubican los threads de usuario, son de tiempo compartido y se gestionan con disciplina **FCFS** (Planificación primero en entrar, primero en ser servido).

En un multiprocesador con N procesadores, la planificación de threads en Windows se basa en dos criterios: (1) afinidad al procesador, y (2) asignar N-1 procesadores a los N-1 threads más prioritarios y el restante a todos los demás.

La política de planificación para los 16 niveles superiores es Round Robin (túnel circular) el cual pertenece a los algoritmos **apropiativos**, Uno de los algoritmos más antiguos, simples, equitativos y de mayor uso y es un método para seleccionar todos los elementos en un grupo de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.

Para la utilización equitativa de los recursos del equipo, Round Robin limita el tiempo de proceso a un quantum de 10 a 100 milisegundos. Cuando el tiempo pasa, el proceso es expropiado e insertado al final de la cola de listos.

Si hay n procesos en la cola de listos y el quantum es q, cada proceso recibe  $1/n$  del tiempo de CPU en intervalos de q unidades de tiempo como mucho. Ningún proceso espera más de  $(n-1)$  unidades de tiempo.

La desventaja principal de este algoritmo es que cambia los procesos en ejecución con demasiada frecuencia, lo que supone una pequeña pérdida de tiempo.

### 4.3. Gestión de Memoria

Windows server utiliza **paginación multinivel** por lo que cuando la cantidad de memoria que usan todos los procesos existentes supera la RAM disponible, el sistema operativo mueve las páginas de uno o más espacios de direcciones virtuales a la unidad de disco del equipo.

Esto libera ese marco de RAM para darle otros usos. Estas páginas se almacenan en uno o más archivos (archivos **Pagefile.sys**) en la raíz de una partición. Puede haber solo un archivo de estos en cada partición de disco. La ubicación y el tamaño del archivo de paginación están configurados en **Propiedades del sistema**. (en Avanzado/Rendimiento/Configuración).

El administrador de memoria realiza también E/S más grandes al escribir datos en el **archivo de paginación**. Mientras que Windows Server 2003 a menudo realizaba escrituras incluso inferiores a 64 KB., en Windows Server 2008, el administrador de memoria produce comúnmente escrituras de 1 MB.

El tamaño ideal del archivo de paginación depende de la cantidad de memoria física instalada y la memoria virtual que requiere la carga de trabajo; Mas sin embargo en los sistemas de servidores, suele ser recomendable contar con suficiente memoria RAM para que nunca falte y para que el archivo de paginación prácticamente no se use.

### 4.4. Sistema de archivos

El administrador del sistema de archivos Windows Server, está diseñado para trabajar en un amplio rango de entornos de cómputo y algunos otros sistemas operativos. Como resultado, la estructura de **su almacenamiento de archivos es flexible**.

Windows 2000 Server y Windows Server 2003 agregaron el almacenamiento dinámico, para creación dinámica y cambios a volúmenes.

En Windows Server 2008 se pueden reducir y extender los volúmenes dentro de un disco de almacenamiento básico, con el mismo efecto que particionar de nuevo el disco, pero existen otras características avanzadas, como volúmenes abarcando dos o más discos en demanda de almacenamiento dinámico. También es posible elegir (disco por disco) el tipo de almacenamiento que se desea utilizar, pero sólo puede usarse un tipo de almacenamiento por unidad. Así que, para tener ambos tipos de almacenamiento en un equipo, es necesario que se cuente con dos o más unidades .

El sistema de archivo de Windows Server puede abarcar **mucho más de un disco**: desde todos los discos de un solo equipo a todos los discos en una red y, también, volúmenes almacenados fuera de línea.

La administración de este sistema es significativa y, por tanto, Windows Server tiene un juego importante de herramientas para manejar la administración del sistema de archivos. Entre estas herramientas figuran:

- Administración de discos,

- Administración de volúmenes dinámicos,
- Sistema de archivos distribuido,
- Respaldo de disco y recuperación.

Otra de las características de Windows Server es que él cuenta con funcionalidad nativa de almacenamiento en capas (tiered) La mayor mejora del WS 2012 R2 es la inclusión de la funcionalidad de almacenamiento en capas de manera nativa, lo cual es una expansión de **Windows Storage Spaces**. Esto significa que un administrador podrá añadir discos de estado sólido (SSDs) y discos duros tradicionales (HDDs) en un espacio de almacenamiento, y que el motor de espacio de almacenamiento automáticamente establecerá diferencias entre ambos tipos de almacenamiento. Así, **Windows moverá bloques de almacenamiento que son leídos con mayor frecuencia a los SSDs, mientras que la información que es menos accedida irá a los HDDs**

Microsoft está poniendo a disposición nuevas opciones de tolerancia a fallos. La opción de doble paridad ahora crea una estructura de disco lógico (parecida a RAID 6).

Windows Server proporciona varias mejoras que ayudan a los administradores a mantener el funcionamiento óptimo de su servidor, entre ellas, la **reparación en línea de la coherencia de NTFS**, una infraestructura nueva para la generación de informes de errores de hardware y extensiones del comprobador de controladores.

Con los dispositivos de almacenamiento de varios terabytes actuales, la desconexión de un volumen para comprobar la coherencia puede tener como resultado una interrupción del servicio de varias horas. Con el reconocimiento de que muchos daños en disco se encuentran en un solo archivo o en una parte de los metadatos, Windows Server implementa una característica nueva de recuperación automática de NTFS para reparar daños mientras el volumen permanece en línea .

Cuando NTFS detecta los daños, impide el acceso a los archivos dañados y crea un subprocesso de trabajo del sistema que ejecuta correcciones de tipo **Chkdsk** en las estructuras de datos dañadas, y permite el acceso a los archivos reparados al terminar.

#### 4.5. Active Directory y seguridad del sistema

Uno de los aspectos esenciales de un sistema operativo multiusuario es la seguridad y protección del mismo.

Así **Tanenbaum (2009)** explica: " ... los problemas generales involucrados en el proceso de evitar que personas no autorizadas lean o modifiquen los archivos ... " y protección como" ... los mecanismos específicos del sistema operativo que se utilizan para salvaguardar la información en la computadora." Tanenbaum (2009).

La seguridad dentro de los sistemas operativos: involucra tener control sobre los procesos que se ejecutan; así como preservar la integridad, confidencialidad y disponibilidad de sistema, protegiéndose de ataques de manera tal que tal que se garantice su integridad. **Seguridad es mantener el control.** Seguridad es preservar:

- Integridad.

- Confidencialidad.
- Disponibilidad.

La **estrategia central de seguridad** de Windows Server es el uso de **Active Directory** para almacenar **cuentas de usuario** y proporcionar servicios de autenticación, que permitirán a los usuarios identificados tener accesos y/o privilegios sobre los recursos del sistema.

La autenticación de los usuarios de un equipo puede ser de manera sencilla mediante la utilización de ID de usuario y contraseña de acceso, o puede basarse en métodos más sofisticados como puede ser la utilización de tarjetas inteligentes o dispositivos biométricos.

Windows Server 2008 además incorpora varias medidas extra para el control de la seguridad:

- **BitLocker**, un software que utiliza al chip Módulo de Plataforma Segura (TPM por sus siglas en inglés Trusted Platform Module) de hardware para proporcionar cifrado de disco para los datos y los volúmenes del sistema.
- **RMS** Administrador de Almacenamiento Extraíble (por sus siglas en inglés Removable Storage Manager), mediante el cual es posible administrar y catalogar las unidades extraíbles del almacenamiento (memorias USB), de tal manera que se puedan activar y desactivar de acuerdo a las necesidades de la organización o preferencia de los administradores del sistema
- **NAP** Protección de Acceso a Redes (por sus siglas en inglés Network Access Protection)
- **RODC** Controlador de dominio de sólo lectura (por sus siglas en inglés Read Only Domain Controller).
- **PatchGuard**, que reducen la exposición a ataques del núcleo, lo que produce un entorno de servidor más seguro y estable.

#### 4.6. Hipervisor

El hipervisor, también llamado monitor de máquina virtual (VMM), es el núcleo central de algunas de las tecnologías de virtualización de hardware más populares y eficaces, entre las cuales se encuentran las de Microsoft: Microsoft Virtual PC, Windows Virtual PC, Microsoft Windows Server e Hyper-V.

Los hipervisores son aplicaciones que presentan a los sistemas operativos virtualizados (sistemas invitados) una plataforma operativa virtual (hardware virtual), a la vez que ocultan a dicho sistema operativo virtualizado las características físicas reales del equipo sobre el que operan.

Con el uso de hipervisores es posible conseguir que múltiples sistemas operativos compitan por el acceso simultáneo a los recursos hardware de una máquina virtual de manera eficaz y sin conflictos.

#### 4.7. Características Avanzadas

Con **Windows Server 2019**, se introducen varias características avanzadas que mencionamos a continuación:

**Escenarios de nube híbrida:** El enfoque híbrido de este SO es debido a que combina entornos locales y de la nube, lo cual permite la sincronización de servidores de archivos, ampliación de Active Directory y copias de seguridad en la nube. Adicionalmente, los clientes podrán integrar fácilmente servicios como AzureBackup, Azure File Sync y recuperación ante desastres, sin que sus aplicaciones o infraestructura se vean afectadas.

**Seguridad:** Esta versión incorpora nuevas funciones de seguridad basadas en tres pilares fundamentales: proteger, detectar y dar respuesta. Windows Server 2019 cuenta con Windows Defender **Advanced Threat Protection** (ATP) (un nuevo conjunto de capacidades de prevención de intrusión de host), que ofrece:

- Gestión centralizada de la seguridad.
- Detección de ataques y exploits zero-day, es decir ataques cibernéticos al sistema el mismo día en que se detecta la vulnerabilidad del mismo.
- Attack Surface Reduction (ASR), para bloquear archivos maliciosos.
- Protección de red.
- Acceso controlado a carpetas.
- Protección contra explotaciones de vulnerabilidades.

**Windows para Linux (WSL):** Se ha incluido contenedores Linux en Windows para que los desarrolladores puedan ejecutar, en una única infraestructura, ambos tipos de contenedores. De igual manera, los usuarios de Linux podrán ejecutar sus scripts en Windows gracias al subsistema WSL. Además, con esta nueva versión se admitirán máquinas virtuales Linux.

**Infraestructura hiperconvergente (HCI):** HCI es una infraestructura de TI definida por software que virtualiza todos los elementos de los sistemas convencionales definidos por hardware.

## 5. Conclusión.

### 5.1. Sistema educativo

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)
- Implementación de sistemas operativos (ASIR) (PES/SAI)

## 6. Bibliografía

- Tanenbaum, Andrew S. Sistemas operativos modernos 3<sup>a</sup> Ed. Ed. Prentice Hall.
- Stalling, William. Sistemas operativos. Ed. Prentice Hall.
- Sistemas operativos monopuesto. Laura Raya González, Miguel A. Martínez Ruiz. Ed. Ra-Ma.
- Sistemas operativos monopuesto. María del Mar Alegre Ramos. Ed. Paraninfo.
- Administración de Sistemas Operativos. Julio Gómez López, Oscar David Gómez López. Ed. RA-MA.
- El libro oficial de Ubuntu Server. Kile Rankin, Benjamín Mako Hill. Ed. Anaya - Prentice Hall.
- Sistemas Informáticos. José Luís Raya Cabrera. Laura Raya González Javier S. Zurdo. Ed. RA-MA.
- Sistemas Informáticos. Isabel M.<sup>a</sup> Jiménez Cumbreras. Editorial Garceta 2017.
- <https://es.wikipedia.org/wiki/Android>
- <https://androidos.readthedocs.io/>
- <https://developer.android.com/topic/performance/memory-management?hl=es>
- [https://es.wikipedia.org/wiki/Arquitectura\\_de\\_Windows\\_NT](https://es.wikipedia.org/wiki/Arquitectura_de_Windows_NT)