

# SISTEMAS OPERATIVOS MULTIMEDIALES

---

## 1) ¿Qué se entiende por multimedia? ¿Y por sistema de información multimedial?

El término 'multimedia' hace referencia a la capacidad de presentar simultáneamente tipos variados de información, como parte de un diseño común.

La mayoría de la gente emplea la palabra multimedia para indicar un documento con dos o mas formas de "media" (video, audio, etc), esto es media que puede ser reproducida en un intervalo de tiempo.

## 2) ¿Cuáles son las exigencias de soporte multimedial para los sistemas operativos?

Las áreas donde la multimedia debe ser soportada es en la reproducción y edición de video, música, en juegos de computadora y en la provisión de video por demanda, teniendo en cuenta que la multimedia maneja un promedio de datos extremadamente alto y requiere de reproducción en tiempo real.

Para soportar todo esto se necesita un sistema operativo con un adecuado file system, scheduling de proceso y scheduling de disco.

## 3) ¿Qué es la calidad de servicio en un sistema operativo para multimedia? ¿Qué nuevas operaciones de planificación aparecen?

La calidad de servicio (QoS) consiste en un conjunto de parámetros que se deben prever y satisfacer para reproducir multimedia en condiciones aceptables, incluye parámetros como promedio de ancho de banda disponible, pico de ancho de banda disponible, "delay" mínimo y máximo, y probabilidad de pérdida de bit (bit loss).

Las operaciones de planificación consisten en reservar capacidad (o recursos) para aceptar nuevos usuarios. Los recursos reservados incluyen una porción de CPU, buffer de memoria, capacidad de transferencia de disco, y ancho de banda de la red. Los servidores multimediales necesitan esquemas de reserva de recursos y algoritmos de control de admisión para decidir cuando pueden tomar mas trabajo.

## 4) Describa una posible organización en subarchivos de un archivo multimedia. ¿Qué nuevas operaciones sobre los archivos aparecen?

Si tenemos en cuenta que una película viene acompañada de sonido y subtítulos podemos pensar que esta está representada por un archivo de video, varios archivos de sonido y de texto. En consecuencia el sistema de archivos necesita mantener múltiples subarchivos por archivos. Una posible organización es manejar cada subarchivo como un archivo tradicional (ejemplo: con inodes) y tener una nueva estructura de datos que liste todos los subarchivos por archivo multimedia. Otra forma es inventar una especie de inodo de dos dimensiones, con cada columna listando los bloques de cada subarchivo. En general la organización debe ser tal que haya opción para cambiar dinámicamente de subtítulos e idioma mientras se reproduce la película.

## 5) Describa someramente la codificación de audio y video.

Codificación de audio: las señales de audio analógicas pueden ser convertidas a digital con un conversor analógico/digital, este toma una señal analógica y la convierte en una salida binaria. Para generar una señal digital debemos "muestrear" la señal analógica cada  $\Delta T$  segundos. La representación digital no es exacta, el error introducido por el número finito de bits por muestra se llama error de cuantización, si este es muy grande el oído puede percibirlo.

Codificación de video: para comprender la codificación de video, debemos empezar por video en blanco y negro (analógico). Para representar la imagen la cámara escanea un haz de electrones rápidamente a lo largo de la imagen grabando la intensidad de luz. Al finalizar el escaneo, llamado frame, el haz vuelve a comenzar. La intensidad es transmitida y los receptores repiten el proceso de escaneo para reconstruir la imagen. El sistema NTSC tiene 525 líneas de escaneo, y 30

cuadros por segundo. Existen dos técnicas para mostrar la imagen, la primera reconstruye la imagen presentando las líneas escaneadas desde la parte superior hasta la inferior y se la conoce como progresiva, mientras que la técnica de entrelazado (interlacing), primero muestra las líneas impares y luego las pares.

El video a color (analógico) utiliza la misma técnica, pero utiliza tres haces que representan los colores rojo, verde y azul (RGB), que son combinados para mandar una única señal.

La representación más fácil de video digital es una secuencia de cuadros que consiste en una grilla rectangular de pixeles, con 8 bits por pixel es suficiente para representar los colores RGB dando 16 millones de colores. Para dar una buena imagen con movimientos suaves es suficiente mostrar 25 cuadros por segundo. Los monitores de computadora utilizan la técnica progresiva para mostrar la imagen.

#### 6) Describa someramente las operaciones de la técnica de compresión MPEG.

MPEG consiste en una técnica para comprimir video, es un estándar y presenta los algoritmos MPEG-1 y MPEG-2. Ambas versiones aprovechan las 2 principales redundancias que existen en video, espacial y temporal. La redundancia espacial se lleva a cabo comprimiendo cada cuadro por separado con JPEG, mientras que la redundancia temporal consiste en la comparación entre cuadros consecutivos.

Se distinguen tres tipos de cuadros:

- *I frames*: cuadros codificados con JPEG
- *P frames*: diferencias en bloques entre el cuadro actual y el anterior
- *B frames*: diferencias entre el actual, el anterior y el siguiente cuadro.

P frames y B frames se basan en la idea de macrobloques (grilla de bloques que cubren toda la imagen) para detectar qué partes son iguales y cuales son distintas en la sucesión de cuadros.

#### 7) ¿Qué es el video on demand y el near video on demand? ¿Qué exigencia aparece sobre el sistema operativo?

Video on demand consiste en un servicio en donde los consumidores seleccionan un video usando el control remoto (o mouse) y este se reproduce en su televisión (o monitor).

Near video on demand, es similar pero la reproducción se lanza a un determinado horario y se relanza cada delta T minutos (comienza a las 8, se relanza 8:05, 8:10 etc.), o sea que el video no comienza al instante que es pedido sino casi “cerca” al momento del pedido.

La exigencia que aparece en el sistema operativo es una adecuada administración de los streams transmitidos, reserva de recursos y provisión de los bloques de datos sin demora.

#### 8) ¿En la jerga de streaming, qué se conoce como tecnología push y pull?

Pull: los usuarios hacen llamados reiterados para que los datos sean entregados. Es una secuencia donde para que un bloque de datos sea entregado primero debe ser requerido.

Push: el usuario hace un system call de start y el servidor de video comienza a entregar los datos sin necesidad de ser requeridos.

#### 9) Compare los modelos de archivos multimediales con bloques pequeños y con bloques grandes (con y sin fragmentación interna) en cuanto a: uso de RAM, desperdicio de espacio en disco, tiempo I/O, cue, rew.

Organización con bloque pequeño: los bloques de disco son mas chicos que el tamaño promedio de los frames. La idea es tener una estructura de dato con un índice de frame apuntado al inicio del frame. Cada cuadro consiste de audio, video y texto como una secuencia contigua de bloques de disco.

Organización con bloque grande: pone múltiples frames en cada bloque, requiere de un índice de bloques. Un bloque puede no contener una cantidad pareja de cuadros, dando lugar a dos opciones, como primer opción cuando un frame no entre en el bloque, se deja el espacio libre. Como segunda opción se puede llenar cada bloque hasta el final, partiendo los frames entre los bloques, introduciendo la necesidad de hacer seeks entre frames.

Comparación:

|                                      | <b>Bloque pequeño</b>               | <b>Bloque Grande sin fragmentación</b>  | <b>Bloque grande con fragmentación</b>   |
|--------------------------------------|-------------------------------------|---|--|
| <b>RAM</b>                           | Mucho uso                           | Poco uso  | Poco uso   |
| <b>Desperdicio espacio</b>           | Bajo                                | Mayor   | No hay   |
| <b>Tiempo I/O</b>                    |                                     |   |  |
| <b>Fast forward y fast backguard</b> | Es posible a través de los I frames | No es posible de hacer, directamente, solución: otro archivo que cuando se reproduce de la sensación que lo hace en 10x | No es posible de hacer directamente, solución: otro archivo que cuando se reproduce de la sensación que lo hace en 10x |

**10) ¿Cómo varía la exigencia de uso de los archivos en un servidor de near video on demand?**

**11) ¿Cómo se usa la ley de Zipf para colocar varias películas en el mismo disco?**

Para películas en servidores, la ley de Zipf dice que la película más popular es generalmente dos veces más elegida que la segunda más popular, tres veces mas que la tercera más popular y así sigue.

Una conclusión es que se necesita tener muchas películas para establecer el ranking de popularidad.

Conociendo la popularidad relativa de las diferentes películas es posible modelizar la performance de un servidor y utilizar esa información para colocar los archivos de video. La estrategia es simple, consiste en colocar la película más popular en el medio del disco, con la segunda y tercer película a ambos “costados”, mas afuera de estas la cuarta y la quinta etc.

**12) Describa el proceso por hard de la multimedia según tecnologías conocidas como DSP, ASIC, y Stream processor.**

Para la próxima edición ...

# PALM OS

---

## 1) Describa los componentes de hard de una plataforma Palm.

Componentes de hard de una Palm:

- Pantalla que recibe eventos (taps) del usuario por medio de un lápiz.
- Zona de la pantalla permitida para escritura, mediante ciertos jeroglíficos.
- Cuatro botones programables, que de fábrica, llaman a las aplicaciones comunes (agenda, schedule, memo, To Do).
- Botón de encendido (que también sirve para encender la luz fluorescente).
- Sócalo de sincronización, para conectarla con la PC.
- Conector infrarrojo para comunicarse con otra Palm u otros dispositivos (ejemplo: Televisión).

## 2) Cuales son los modos de consumo de potencia de una Palm.

Modo Sleep: se mantiene el flujo de energía a la memoria, pero se la quita a la pantalla y a los circuitos de entrada.

Modo Doze (siesta): el procesador se detiene después de ejecutar cada instrucción (debe tener un micro específicamente diseñado para ese propósito).

Modo Running: mientras se esta ejecutando una instrucción.

## 3) Que hay en RAM y en ROM, que parte puede asimilarse a un file system y que parte a la memoria RAM de una computadora de propósito general.

En ROM esta el sistema operativo y las aplicaciones básicas que vienen por default en la palm. En la RAM se cargan los programas nuevos del usuario (que persisten mientras se le provea energía).

La RAM se comporta (a diferencia de la pc) de manera similar a un file system, posee las primitivas conocidas y es administrada de esa manera. El análogo a la RAM de la computadora personal es la ROM de la Palm ya que esta si esta organizada de esa manera.

En algunas Palm, el SO se aloja en la RAM, con lo cual es posible cargarle una nueva versión del SO.

## 4) ¿Qué es un *soft reset*, *soft reset+page up* y un *hard reset*?

Tipos de Reset: Soft, Warm, Hard.

Soft reset: vuelve a lanzar todas las aplicaciones que estaban lanzadas (esto se hace principalmente cuando se colgó o no funciona algo en particular pero se desea seguir trabajando en el ambiente actual).

Hard reset: borra toda la ram (esto se hace principalmente cuando se desea dejar la configuración de la palm en cero para cargarle otros programas).

Warm reset: a diferencia del soft reset, las aplicaciones quedan cerradas (esto se hace principalmente después de un cuelgue “importante” y cuando no se desea seguir en el mismo ambiente).

Igualmente, como recomendación del profesor, habria que bajarse un POSE para ver mejor la funcionalidad de la Palm para los “Palm-disabled” (<http://www.palm.com>).

## 5) ¿Qué es POSE?

POSE (Palm Operating System Emulator) es un programa que se ejecuta en una pc de escritorio que emula la palm. Se usa generalmente para desarrollo (prueba de aplicaciones) y se le debe cargar la ROM del modelo de Palm requerido antes de ser usado.

## 6) ¿Cómo es la sincronización, backup e instalación de aplicaciones en una palm?

El objetivo de la sincronización de una Palm con la PC es la de tener la misma información en ambos lados, tanto para consulta (en la PC mediante un soft) como para recuperación de datos en caso de pérdida.

La comunicación entre la Palm y la PC se hace mediante un programa sincronizador, y apoyando la Palm en una plataforma que se conecta al puerto serie de la PC. La transmisión de datos puede ser de las siguientes formas:

- Palm a PC: Lo que está en la Palm es lo nuevo, y pisa a lo que está en la PC
- PC a Palm: (lo opuesto)
- Sincronización: en este caso, el programa que sincroniza lleva un registro de qué cambios hubo tanto en la Palm como en la PC y actualiza a ambos con la última información. Esto funciona bien cuando se sincroniza con una sola PC, pero si a la misma Palm se la quiere sincronizar con más de una PC, habrá que tener ciertas precauciones para no perder información.

## 7) ¿Cómo es el lanzamiento de una aplicación Palm? ¿Qué son los launch codes?

Launch code: es lo que se le da a la aplicación cuando se la ejecuta (lanza).

Tipos de lanzamiento:

- Normal: la aplicación da entrada a su Stara app
- Initializing: aparecieron nuevos datos globales para la aplicación a través de la sincronización.
- Notification: pueden o no interesar a las aplicaciones (sync, alarmas, resets)

## 8) ¿Cómo es la arquitectura de eventos y las categorías Pen Event, Key Event y UIEvent?

El Event Manager es el que maneja todos los tipos de eventos en la Palm. El GetEvent (en el Event Manager) es el que espera un evento sobre la parte de digitalización y ejecuta lo que corresponda dependiendo del evento. Si recibe más de un evento, decide cuál tiene mayor prioridad. PenEvent, KeyEvent y UIEvent son eventos generados por el lápiz, por una tecla presionada o mediante la interfaz de usuario, respectivamente.

## 9) ¿Qué son los Hooks o Hacks y que es un hackmaster?

Hooks o Hacks son programas que están por debajo de los eventos, los captura antes que el GetEvent y después genera el evento que corresponda.

El Hackmaster es el que maneja las secuencias de los hacks.

## 10) Describa las particularidades de la administración de memoria como File System.

Las particularidades se enumeran a continuación:

- El acceso se hace **sin buffers** (obvio pero hay que dejarlo claro)
- Los datos se dividen en registros dispersos en la memoria (como los de un file system en disco). Estos registros se llaman **chunks** (espacios de datos contigua) Una Data-Base es una lista de "**chunks**".
- Los frames de memoria se llaman "**heaps**".
- La alocaión de una Data-Base es una lista de pares de mapeo (chunk, heap).
- La defragmentación consiste en mover chunks a heaps contiguos.

Notar la similitud con los file-systems y la "naturalidad" del concepto de Virtual File System como "super-clase" de las distintas implementaciones de file-systems.

### 11) ¿Qué diferencia hay entre Databases y Resources?

Ambos sirven como soporte de información, y la diferencia principal radica en que el primero (pdb) es para datos ordenables y el segundo (prc) no lo es.

### 12) Describa las facilidades de expansion.

El Expansion Manager (con funcionamiento similar al autorun en los CDs) es el que detecta hardware de expansión, y le manda información Slot Driver que es el driver físico que provee las operaciones de bajo nivel.

Hay interfaces de expansión series (bit a bit) y de bloques. Se accede a través de un Virtual File System ya que las "tarjetas" de expansión tienen su propio file system y "responden" a un conjunto de primitivas más o menos standard (en general, son VFAT con una limitación en la cantidad de subdirectorios). Las aplicaciones (y los datos) **no se leen directamente** (por performance y gasto de potencia) **sino por copy** (posiblemente parcial) & run. Hay un Launch-code especial para lanzar aplicaciones por Copy&Run, y otro para notificar inserción.

### 13) Describa las facilidades de intercambio de objetos.

Se utiliza el "planchado" o Flatening de objetos, que es como la "serialización" en Java, para poder manejar los objetos.

Utiliza el Typed Data Objects, que es el mismo estándar que usa MIME.

La aplicación se comunica con el Exchange Library para el intercambio de objetos.

Algunas Exchange Librarys:

- Infrarrojo (IrDA).
- Local Exchange.
- BlueTooth (para comunicarse con dispositivos celulares).
- Messaging System (ej TCP/IP).

(para más información: <http://www.imc.org/pdi>).

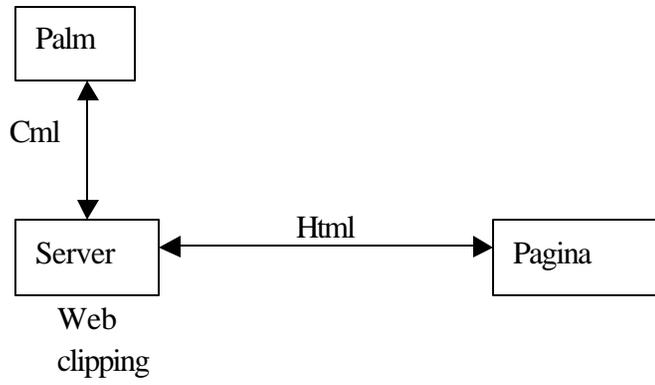
### 14) Describa la arquitectura general de IrDA y de las Comunicaciones Series (probablemente deba tener conocimientos de redes para esto).

No es tomable, me interesaba que supieran que IrDA es un Standard y que **NO es el usado por Win KK**.

### 15) ¿Qué es y como funciona el web clipping?

Para poder adaptar las páginas web creadas para pc de escritorio a su visualización en palm se le deben hacer modificaciones para poder soportar su interfaz gráfica y además achicar el tamaño de la misma para ser enviado en forma inalámbrica (por su alto costo).

Esto se hace con un programa en el servidor que toma las páginas web y les aplica un "filtro" convirtiéndolas en páginas CML (Compressed Markup Language).



**16) ¿Qué es un *conduit*?**

Un Conduit es una DLL de Windows que proporciona un puente entre el uso de la PALM y el de una PC de manera conjunta. Es responsable de los datos de la aplicación durante una sincronización entre la PALM y la computadora personal.

# WINDOWS NT / 2000

1) Indique en que *feature* (característica) de NT aparecen visibles los siguientes criterios de diseño: extensibilidad, portabilidad, privacidad, compatibilidad (según Microsoft).

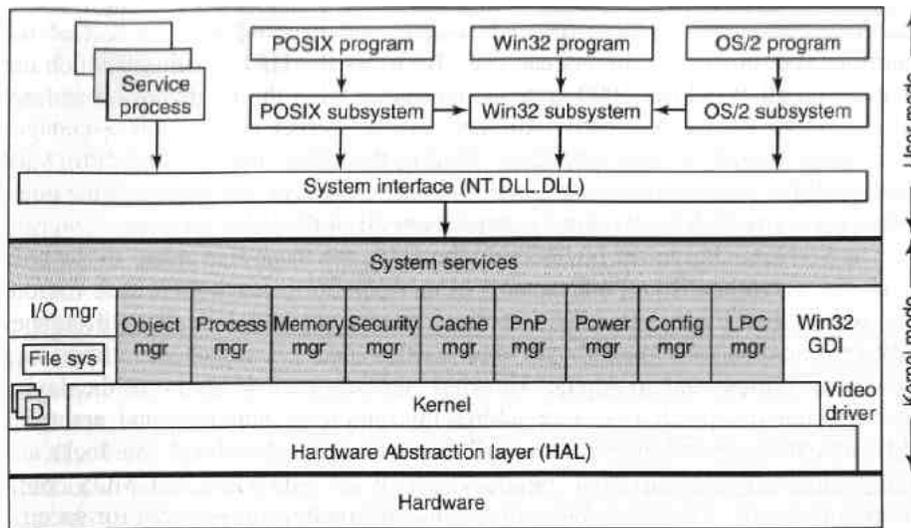
Extensibilidad: permite adaptarse a desarrollos de hardware y a la creación de nuevos dispositivos. NT (original) lograba la extensibilidad por medio del concepto de subsistemas que permitiría agregar nuevos ambientes de ser necesario y de los "drivers" cargables en memoria por medio del conocido mecanismo del Config.sys.

Portabilidad: permite migrar de plataforma. El concepto de HAL (*Hardware Abstraction Layer*) es el que libera al hard de las diferentes arquitecturas (en realidad del "chipset" ya que la diferencia del procesador viene dada por la compilación del Kernel para diferentes plataformas). Al quedar NT solo para plataformas Intel (no soporta mas Alpha ni Power-PC) la portabilidad queda restringida a Intel 32 (IA32) o Intel 64 (IA64, si es que esto existe).

Privacidad: cumple con los requerimientos de seguridad del Libro Naranja a un nivel C2.

Compatibilidad: Las aplicaciones de Win16 que no acceden al hard en forma directa se soportan. No debería haber aplicaciones Win32 que lo hicieran para permitir la unión de las plataformas Win kk con las NT para el 2000. Pero la historia dijo otra cosa. El intento de soportar Subsistemas OS/2 y Posix es también otro ejemplo de como Microsoft uso estos objetivos de diseño como propaganda para "vender" NT a los técnicos que no le creyeron y que no se equivocaron. Una vez impuesto el marketing, son solo recuerdos y papeles en el escritorio de algún Juez (demócrata).

2) Indique en un diagrama la estructura de la Arquitectura de NT y las funciones generales (una línea) de cada elemento.



The executive:

Se llama así a el conjunto de funciones integradas por los "manager", el "system services" y el "system interface".

Object Manager:

Maneja todos los objetos conocidos en el sistema, esto incluye los procesos, threads, archivos, directorios, semáforos, dispositivos de i/o, timers, etc.

Process Manager:

Es el encargado de manejar los procesos y los threads, incluyendo su creación y su destrucción.

Memory Manager:

Es el encargado de administrar la demanda y paginado de la memoria virtual.

#### Security Manager:

Hace cumplir el mecanismo elaborado de la seguridad de Windows 2000, que resuelve los requisitos de U.S. Dept. of Defense's Orange Book C2.

#### Cache Manager:

Mantiene los bloques de disco más recientemente usados en memoria para acelerar el acceso a ellos en caso que se necesiten otra vez .

#### PnP Manager:

Se encarga de todos los dispositivos plug and play y de notificar sus acoplamientos al sistema.

#### Power Manager:

Administra el uso de la energía. Esto consiste en apagar el monitor y los discos después de que hayan estado inactivos por un cierto tiempo.

#### Config Manager:

Está a cargo de la administración del "registry". Agrega nuevas entradas y devuelve las claves cuando se hace su pedido.

#### LPC Manager:

El LPC (*Local Procedure Call*) prevé una comunicación entre procesos altamente eficiente usada entre los procesos y sus subsistemas.

### 3) ¿Qué se almacena y como es la estructura del "registry"?

Windows 2000 necesita guardar una gran cantidad de información, que es crucial para su funcionamiento, que tiene que ver con el hardware, software y los usuarios. A partir de Windows 95 se decidió juntar todos los archivos .ini (que era donde estaba guardada toda esta información) en una base de datos central llamada "Registry".

La estructura del "registry":

*Figura Pág. 776*

### 4) ¿Qué es un Sistema Operativo de Microkernel? ¿Cómo se implementa y cambia el concepto de NT?

Una tendencia en sistemas operativos modernos es tomar la idea de mover la mayor cantidad de código posible a capas más altas y quitar tanto como posible desde modo del "kernel", dejando un "microkernel" mínimo.

El acercamiento a esto se hace implementando la mayor parte del sistema operativo en procesos del usuario. Para solicitar un servicio, tal como lectura de un bloque de un archivo, un proceso del usuario (ahora llamado "Proceso del Cliente") envía la petición a un "Proceso del servidor", que después hace el trabajo y envía la respuesta.

NT lo implementa en su primera versión (3.5) logrando un sistema muy estable e imposible de correr en aplicaciones comerciales. Con NT 4.0 abandona la implementación de Microkernel (aunque queda como figura en los libros y como "modelo de diseño" pero no de implementación) y vuelve a los sistemas Monolíticos. No hay un sistema comercial de microkernel (la palabra clave es "comercial") en la actualidad. La "tendencia" está marcada por el mundo académico y, a pesar que los académicos estamos todos de acuerdo con esos principios, el divorcio con la realidad de lo que se usa comercialmente es para pensarlo. Recién cuando se monte algo así como "virtual PC" sobre un sistema con Microkernel se verá si es algo mas que una preciosa moda académica. A pesar del Ghz de reloj y de los poderosos RISCs y mainframes, sigue habiendo dos problemas:

- *Performance:* en monolítico se comparten datos de una forma que cualquier profesor de diseño calificaría como pésimo acople, que en Microkernel se transforman en burocráticos mensajes) y
- \$\$\$: para lograr un Sistema Operativo estabilizado hay que pensar en al menos 5 años con equipos a full, y encima hay que venderlo.

### 5) ¿Cuál es el concepto de Objeto usado en NT/2000? ¿Cuáles son Control Objects y Cuales Dispatcher Objects?

NT no esta orientado a Objetos en la forma entendida por la Programación Orientada a Objetos.

Un objeto para Windows 2000 NT es una unidad de encapsulamiento y una unidad de Acceso Uniforme. Se clasifican en 2 tipos básicos, que son Control Objects y Dispatcher Objects. Los primeros incluye los objetos que modifican el flujo de control, los objetos de interrupción y los DPC (Deferred Procedure Call) y APC (Asynchronous Procedure Call). Los DPC se encargan del manejo de las tareas diferidas, por ejemplo al presionarse una tecla, el proceso de interrupción del teclado

lee el código de la tecla desde un registro y lanza la interrupción, esta interrupción puede ser menos importante que un proceso que se esté corriendo, con lo cual la ejecución de la misma se hace en un tiempo diferido.

Los APC son como los DPC excepto que se ejecutan en el contexto de un proceso específico. Son objetos asincrónicos que no están en modo kernel.

Los Dispatcher Objects incluyen los semáforos, mutexes, eventos, timers y otros objetos en que los threads pueden esperar (Ej. Cola, pila). Estos objetos pueden causar el bloqueo de los procesos.

## 6) ¿Cómo es el mecanismo de ejecución usando subsistemas?

*Figura Pág. 794*

Los procesos de usuarios tienen llamadas a la WIN32 API a través de las dll. Estas dll pueden interactuar directamente con el sistema operativo o bien enviar un mensaje al subsistema de win32, que se encarga de hacer un cierto trabajo, para luego llamar al sistema operativo. En algunos casos puede pasar que el subsistema realice todo el trabajo en modo usuario sin pasar a la llamada al sistema operativo que está en modo kernel.

## 7) ¿Cuál es la relación entre LPC y la API de NT y WIN32?

LPC (Local Procedure Call) es la forma en que se comunican las distintas partes del Executive. Es una comunicación vía pasaje de mensajes en la que el llamador se bloquea hasta que el llamado tenga un resultado y la devuelva. Técnicamente se llama "pasaje sincrónico de mensajes". Las API encapsulan todo esto en una serie de funciones de biblioteca y "serializan" las llamadas, aumentando la estabilidad a costa de la performance. Las aplicaciones de Microsoft usan LPC pero la interface de LPC no está documentada y Microsoft se reserva el derecho de cambiarla. Los desarrolladores deben usar las API. (Esto es la continuación de la política Undocumented Windows). En el escritorio de un juez están los pedidos del management de Microsoft para cambiar la arquitectura externa de las API de forma tal de perjudicar a ciertos programas de Groupware (¿Lotus Notes?) que el Juez decidió mantener en secreto por razones obvias (es lo mismo que decirle a todo el mundo que NO use esa aplicación ya que NO va a andar por diseño).

## 8) ¿Qué relación hay entre Jobs, Process, Threads y Fibers?

Primero veamos rápidamente que es cada una de estas:

*Figura Pág.. 797*

La relación que hay en Windows 2000 es que cada proceso contiene por lo menos un thread, el cual contiene por lo menos un Fiber. Además los procesos pueden agruparse en Jobs por ciertos propósitos de la administración de recursos.

Cambiar de threads en Windows 2000 es relativamente costoso ya que esto requiere entrar y salir del modo kernel. Para simular un paralelismo más rápido, Windows 2000 provee los fibers, que son como threads, pero solo en modo usuario.

## 9) ¿Cómo está organizada la memoria de un proceso?

En Windows 2000 cada proceso tiene su espacio de direcciones virtuales. Estas son un número de 32 bits, con lo cual cada proceso tiene 4 GB de memoria virtual.

Hasta los 2 GB de memoria están disponibles para el código y los datos de proceso. Los 2 GB hasta completar los 4 GB están protegidos para el kernel.

El espacio de dirección virtual es demandado por páginas, con un tamaño fijo de la página. (de 4KB en los Pentium). Un ejemplo:

*Figura Pág. 813*

## 10) ¿Cuáles son los estados de una página virtual?

Hay tres estados:

Free pages: esta sin uso y usarla causaría un “page fault”.

Committed: cuando los datos o código ya están en la página.

Reserved: significa que no se puede usar hasta que no se quite la reservación.

### **11) Describa el algoritmo de balanceo de paginas.**

El algoritmo de balanceo de paginas utilizado por Windows 2000 funciona de la siguiente manera:

El sistema hace una tentativa de mantener un número substancial de páginas libres en memoria para que cuando ocurre una “page fault” poder demandar una página libre, sin la necesidad de escribir una página al disco.

Como consecuencia de esta estrategia, la mayoría de las “page fault” se pueden satisfacer con una sola operación del disco (lectura en la página), en vez de dos (escritura de una página cambiada y después una lectura de la página necesaria).

Por supuesto, las páginas en la lista disponible tienen que venir de alguna parte, así que el algoritmo depende de las paginas que tiene el proceso que pide una pagina.

Si el proceso tiene menos paginas que el mínimo establecido, el algoritmo le da una pagina libre al suceder el “page fault”, si el proceso esta entre el mínimo y el máximo, el algoritmo le da una pagina y le quita otra al suceder el “page fault”, y finalmente si el proceso esta por encima del máximo de paginas el algoritmo le saca 2 paginas y le da una al suceder el “page fault”.

### **12) Describa el proceso de Login de un usuario NT**

Si Windows 2000 NT está configurado con conexión segura, esto significa que el administrador de sistema puede requerir a todos los usuarios tener una contraseña para abrirse una sesión. Si el usuario y contraseña ingresados son válidos, el sistema crea un proceso con un access token asociado. Entre las elementos que tiene el access token, esta el SID que es el identificador que representa al usuario en el sistema por razones de seguridad. Todos los procesos, que de aquí en mas lance este proceso inicial, heredaran el mismo access token.

### **13) ¿Cómo Funciona el Local Procedure Call?**

Quien quiere un servicio conoce de antemano a quien enviarle un mensaje. Formatea (encapsula) el pedido en un formato standard donde indica entre otras cosas quien es (origen), a quien va dirigido (destino) el tipo de servicio (inmediato, diferido, diferible o urgente) y los datos referentes al servicio. El Kernel ubica al destino y le entrega el mensaje. De acuerdo con el tipo, el kernel pasa el llamado a bloqueado y le da control al scheduler. Lo que hace el proveedor del servicio depende del tipo de servicio, puede atenderlo de inmediato, diferirlo o atenderlo pero en modo interrumpible. Tarde o temprano el proveedor llega a un resultado, lo formatea en una cápsula similar a la que recibió y el Kernel lo entrega al que pidió. Hasta aquí es un Cliente Servidor con algunas peculiaridades. Esto implica copiar los pedidos del espacio del cliente al del Kernel, del Kernel al servidor y otras dos copias para la vuelta. Si el llamado se hace vía API, hay que sumarle las copias clásicas de los llamados a funciones. Para el caso de grandes cantidades de datos (como el video o el acceso a disco) estas copias son prohibitivas por el espacio y el tiempo que requieren. En la mayor parte de los LPC se pasan entonces punteros en vez de datos perdiéndose todas las ventajas del encapsulamiento y volviendo al sistema de la "gran área (bodoque) común de datos" que era denostado en la propaganda de NT 3.5. Por supuesto no hay documentación para que una aplicación de alguna otra parte que no sea de Redmont use LPC, o sepa bien qué hace cuando usa el área común de datos.

### **14) Describa los elementos que intervienen en la seguridad.**

Elementos:

Todos los Usuarios y Grupos están identificados por un SID (Security ID). Estos son números binarios con un breve encabezamiento seguido de un gran componente al azar.

Cada proceso tiene un Access Token, que especifica su SID y otras propiedades. Tiene dos propósitos: uno es tener en un solo lugar toda la información necesaria para la seguridad para apresurar la validación de los accesos, y permite que cada proceso modifique sus características de seguridad de manera limitada sin afectar otros procesos del mismo usuario. Su estructura es: Figura Pág. 846.

La DACL por defecto (Discretionary Access Control List): determina qué usuarios y grupos pueden tener acceso a este objeto y para que operaciones

Security Descriptor.

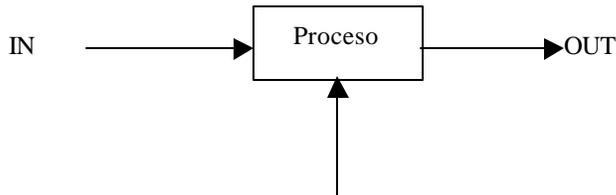
SACL (System Access Control List): especifica qué clases de operaciones en el objeto deben generar mensajes que se registran en el log.

# REAL TIME

---

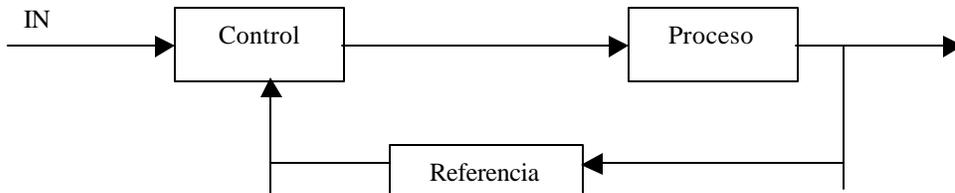
## 1) De ejemplos de los modelos generales de un Sistema de Control de Procesos. Con y sin realimentación.

Sin realimentación (lazo abierto):



Ejemplo: bajar la temperatura cuando calienta el termotanque. Hay un proceso que hace variar la salida de acuerdo a la entrada, pero en ningún momento se verifica si la salida es la deseada.

Con realimentación (lazo cerrado):



Ejemplo: combinar el agua fría y el agua caliente en una ducha de manera de obtener la temperatura correcta. Se va agregando agua caliente (o fría) y se va “testeando” la salida de la ducha mediante el tacto para comprobar si se tiene la salida requerida. El resultado del “test” se utiliza para saber si al instante siguiente se agregará mayor cantidad de agua caliente o de agua fría.

## 2) De ejemplos de un modelo de Control de Procesos en el que el ser humano intervenga en el lazo.

En el ejemplo anterior de la ducha interviene el ser humano.

## 3) De ejemplos de un modelo de Control de Procesos en el que la simulación y el ser humano intervengan en el lazo.

## 4) ¿Qué características presenta un Sistema de Tiempo Real? Explique la división entre *Hard*, *Soft* y *Firm Real Time* (RT).

Características de un sistema de RT:

- Es cualquier actividad de proceso de información que tiene que responder a estímulos generados externamente dentro de un plazo especificado y finito.
- El instante en que se produce el resultado del sistema es significativo.
- Debe ser lo suficientemente rápido para responder aquello para lo cual está diseñado.
- El tiempo interno es igual al tiempo externo.

Hard RT (Crítico): es inadmisibles que se pierda algún plazo.

Soft RT (Acrítico): ocasionalmente puede perderse algún plazo.

Firm RT (Firme): el plazo no es crítico, pero una respuesta tardía no sirve.

### 5) ¿Qué es una *Task* desde el punto de vista de un Sistema de RT? ¿Cuáles son los atributos importantes?

En un sistema de RT cada estímulo del entorno activa una o más tareas. Una tarea es una secuencia de instrucciones que se ejecuta en concurrencia con otras tareas. La ejecución de las tareas se multiplexa en el tiempo en uno o más procesadores.

Existen 3 tipos de tareas:

- *Periódicas*: se repiten cada un tiempo específico y determinado.
- *Aperiódicas*: surgen en cualquier momento.
- *Esporádicas*: se sabe que deben aparecer en un intervalo de tiempo, pero no se sabe bien cuándo.

Los atributos importantes de una tarea periódica son: tiempo de ejecución (o de utilización del procesador), release time (tiempo en que es lanzada la tarea) y deadline (tiempo en que finaliza la tarea). Las tareas esporádicas son tratadas como periódicas en el peor caso, y para atender a las aperiódicas se pueden usar los ciclos de procesador que no tienen uso en el diagrama de tareas a través del tiempo, o sino se puede “crear” una tarea periódica cuya tarea sea atender a la aperiódica. Por lo tanto los atributos importantes de las tareas periódicas son importantes también para las demás tareas.

### 6) ¿Cuándo un proceso es controlado por tiempo y cuándo por eventos?

Existen tres tipos de tareas o procesos: periódicos, aperiódicos y esporádicos.

- *Periódicos*: generalmente muestran datos o forman parte de un bucle de control. Son controlados por tiempo.
- *Aperiódicos*: son controlados por eventos externos que se producen en forma aleatoria.
- *Esporádicos*: se activan en respuesta a eventos externos al sistema de RT.

### 7) Describa los pasos de un diseño de un Sistema de RT. Qué es *time constraining*?

Para empezar se necesita que el hardware y el software sean confiables y seguros.

Se deben conocer las facilidades de tiempo real, es decir, los tiempos en que las operaciones (tareas) deben ejecutarse y completarse.

Requisitos para un Sistema Operativo de Tiempo Real: multitarea con sincronización (ya que la concurrencia de procesos es muy importante en RT), cambio de contexto veloz, rápida respuesta a interrupciones, no debe usar memoria virtual, debe poseer un scheduler de RT, time out, alarmas, primitivas de demora, y debe ser previsible y adaptable al hard.

El scheduler del Sistema Operativo debe garantizar que una tarea vaya a realizarse, ya que esto es tan importante como realizarla.

Requisitos para un Lenguaje de Programación de Tiempo Real: debe poseer tipos fuertes (ya que de lo contrario es más probable el hecho de cometer errores), asignación dinámica de memoria, manejo de excepciones (para todo aquello que no es normal), multithreading, time constrains, soporte de distribución, etc.

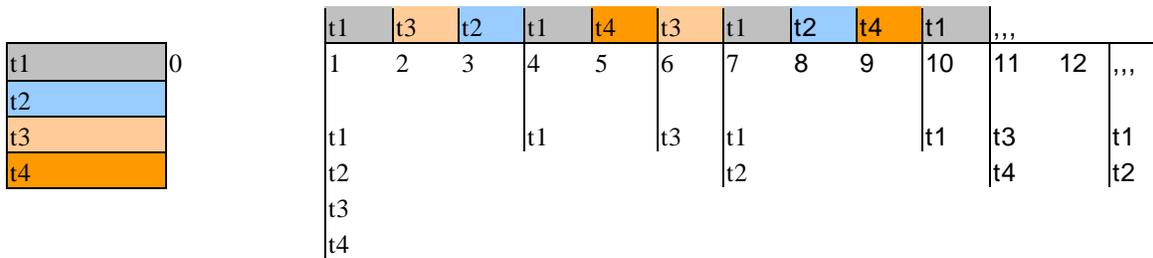
### 8) Cómo funciona el scheduling *Rate Monotonic*?Cuál es la prueba de *schedulability*?

Asigna a cada tarea a realizar una prioridad inversamente proporcional a su período, es decir, que las tareas de menor período tendrán mayor prioridad.

Prueba de *schedulability*:

Ejemplo 1:

4 tareas: t1 (3,1) - t2 (6,1) - t3 (5,1) - t4 (10,3) → (Período, Tiempo de Ejecución)  
 → orden de prioridad: t1 > t3 > t2 > t4

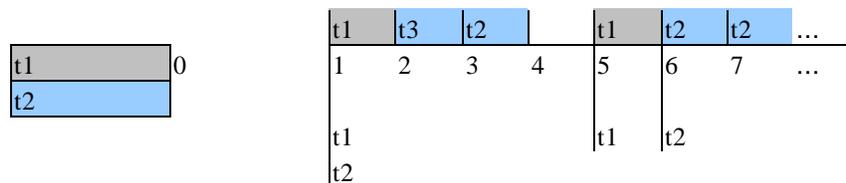


Nunca se llega a cumplir con la tarea número 4, que es la de menor prioridad y mayor duración.

Existe una manera de medir la carga de un procesador para un conjunto de tareas, y es mediante el factor de utilización del procesador:  $U = \sum(e_i/T_i)$ , siendo  $e_i$  el tiempo de ejecución de la tarea  $i$  y  $T_i$  el período de la misma. Para el Rate Monotonic, los plazos están garantizados para todas las tareas si se cumple que  $U \leq N \cdot (2^{1/N} - 1)$ , siendo  $N$  la cantidad de tareas (esta condición es suficiente pero no necesaria). En el caso anterior se tiene que:  $U = 1/3 + 1/6 + 1/5 + 3/10 = 1 > 0,757$ , por eso mismo es que no se puede garantizar que todas las tareas se cumplan en los lapsos estipulados (ej: t4).

Ejemplo 2:

2 tareas: t1 (4,1) - t2 (5,2) → orden de prioridad: t1 > t2

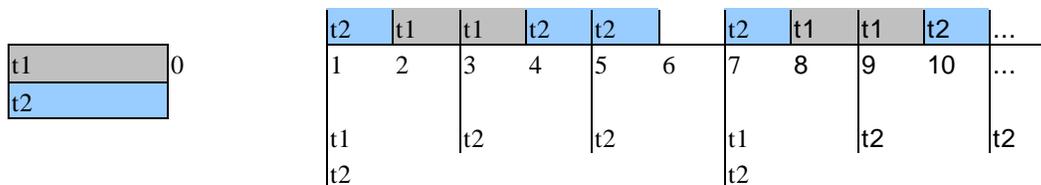


Aquí las tareas se cumplen correctamente en los plazos que se pide.  
 $U = 1/4 + 2/5 = 0,65 \leq 0,828$

**9) Planifique por RM 2 tareas con Periodos 4 y 5 y tiempo de uso 1 y 2 respectivamente. Idem para Periodos 6 y 2 y utilización 1 y 1 respectivamente.**

El primero es el ejemplo 2 del punto 9: t1 (4,1) - t2 (5,2)

Para el segundo: t1 (6,2) - t2 (2,1) → orden de prioridad: t2 > t1  
 $U = 2/6 + 1/2 = 0,833 > 0,828$  → no se garantizan los plazos.



Como se puede apreciar, no se cumple la condición del factor de utilización del procesador, pero igualmente se garantizan las tareas en sus plazos específicos.

**10) ¿Cómo funciona *Earliest Deadline First*? ¿En qué sentido es óptimo y cuál es su prueba de schedulability?**

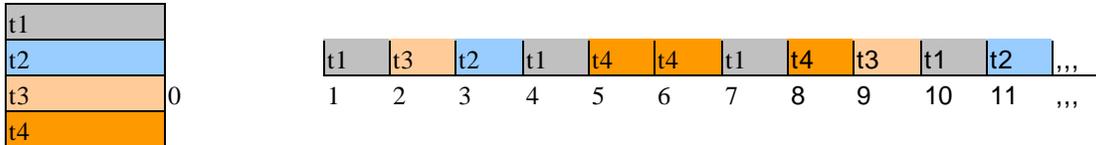
Asigna prioridades dinámicamente, se basa en asignar mayor prioridad a las tareas que tienen el plazo (deadline) más cercano siempre. EDF es óptimo, ya que cualquier secuencia planificada con otro algoritmo EDF también lo hace. Optimiza el uso del CPU.

Prueba de Schedulability:

Ejemplo:

4 tareas: t1 (3,1) - t2 (6,1) - t3 (5,1) - t4 (10,3) → (Período, Tiempo de Ejecución)

→ orden de prioridad: t1 > t3 > t2 > t4

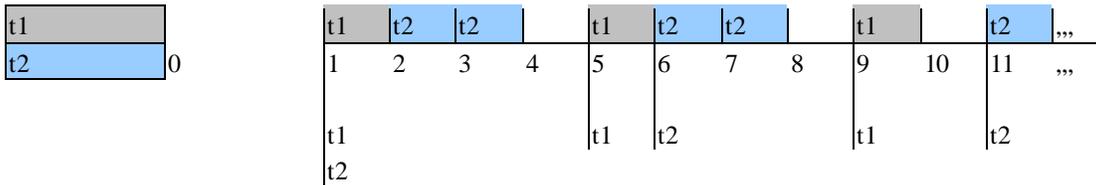


Como puede verse las tareas se garantizan.

**11) Planifique por EDF las tareas de 9.**

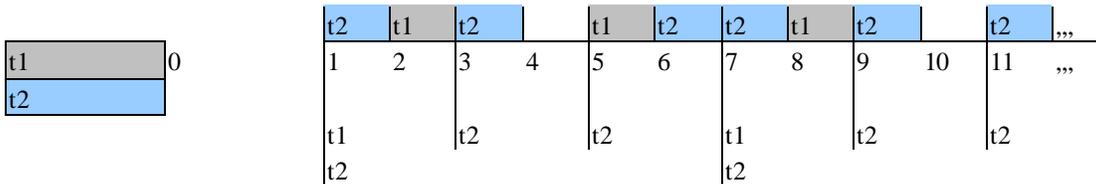
Para el primero:

t1 (4,1) - t2 (5,2)



Para el segundo:

t1 (6,1) - t2 (2,1)



Se garantizan las tareas para ambos casos.

**12) De un ejemplo de tareas que RM no planifique y si EDF.**

El ejemplo número 1 del punto 9 (y el ejemplo del punto 10) es una tarea que EDF planifica y RM no.

**13) ¿Cómo se tratan las tareas esporádicas?**

Las tareas esporádicas son tareas que se sabe que van a aparecer en un intervalo de tiempo determinado, pero no se sabe específicamente en qué momento. Estas tareas se tratan igual que las tareas periódicas en el peor caso.