

Tema 6



Soporte de Sistema Operativo para Multimedia

Soporte de Sistema Operativo para Multimedia



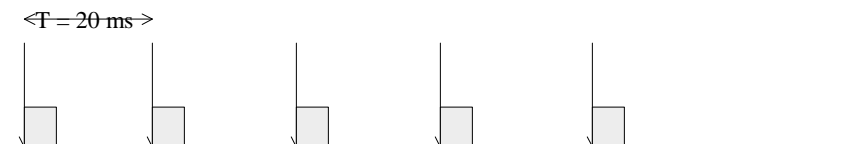
- Proceso de Información Multimedia
- Gestión de Procesos
- Técnicas de Planificación de CPU para Multimedia
- Sistemas de Ficheros para Multimedia
- Políticas de Planificación de disco para Multimedia
- Transferencias de Datos

Proceso de Información Multimedia (I)

- Satisfacer los requisitos de tiempo real que imponen las aplicaciones multimedia conlleva la adaptación de los sistemas operativos a este fin.
- Se combinan los siguientes factores:
 - Tareas periódicas (e.g. Audio, video).
 - Frecuencia alta (e.g. Video digital).
 - Baja latencia (e.g. Interactividad, tiempo de respuesta).

Proceso de Información Multimedia (II)

- Ejemplo: un audio stream 8 KHz PCM se procesa en paquetes de muestras capturadas cada 20 ms que se almacenan en un buffer temporal.
- Una tarea que procesase este audio stream ha de ejecutarse con un periodo de 20 ms.



Proceso de Información Multimedia (III)

- Ejemplo: un video stream MPEG-2 a 30 fps, exige la decodificación de un nuevo frame cada 33 ms.
- Si bien el patrón de ejecución periódica se mantiene, el tiempo de decodificación es variable al tratarse de distintos tipos de frame (I, P, B).



Proceso de Información Multimedia (IV)

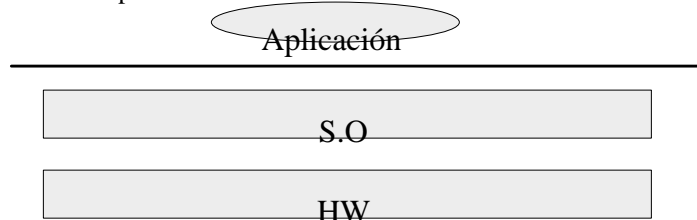
- Limitaciones de los sistemas existentes:
 - Aplicaciones multimedia ejecutadas en sistemas multiprogramados (e.g. Unix, Windows NT):
 - errores temporales y pérdidas de datos cuando hay actividad concurrente
 - estos sistemas operativos inciden en la distribución homogénea y equitativa de recursos entre las distintas aplicaciones, por lo que no tienen en cuenta aquellas aplicaciones que poseen requisitos temporales tales como las aplicaciones multimedia.

Gestión de Procesos (I)

- En las aplicaciones multimedia los cambios de contexto son muy frecuentes (debido a la alta frecuencia del proceso de reproducción y a la existencia en general de diversos flujos simultáneos).
- Se requiere una gestión de procesos ligera para minimizar cargas administrativas en el sistema operativo (*threads*, LWP, ULT's)
 - => los modelos basados en *threads* son preferibles a los basados en procesos.

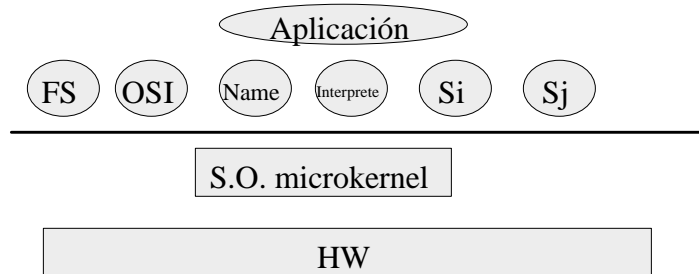
Gestión de Procesos (II)

- El S.O. impone una alta sobrecarga por la interacción requerida aplicación/núcleo: necesidad de invocar llamadas al sistema para realizar operaciones de E/S (lectura videoclips del sistema de ficheros, presentación, transferencias por la red, etc.)
- Núcleo monolítico
 - Todos los servicios del sistema operativo se ejecutan en un espacio de direcciones común => numerosas llamadas al sistema y cruces al dominio de protección del núcleo



Gestión de Procesos (III)

- Núcleo *microkernel*
 - El núcleo sólo implementa servicios básicos. El resto de los servicios se implementan en servidores que se ejecutan a nivel de aplicación
=> no es necesario cruzar el dominio de protección del núcleo, disminuyendo el overhead.



Técnicas de Planificación de la CPU para MM (I)

- Clasificación de las aplicaciones:
 - Aplicaciones de tiempo real “duras”.
 - han de satisfacer los plazos de ejecución en todo caso con un comportamiento determinista (e.g. Proceso periódico estrictamente sin retrasos).
 - Ejemplo: Tele-surgery.
 - Aplicaciones de tiempo real “blandas”.
 - requieren garantías de ejecución en términos estadísticos (no importa si existen pérdidas de plazos ocasionales).
 - Ejemplo: aplicaciones multimedia con una QoS garantizada.
 - Aplicaciones “Best-effort”.
 - requieren un tiempo medio de respuesta bajo (e.g. Aplicaciones multimedia actuales tales como un mpeg_player)

Técnicas de Planificación de la CPU para MM (II)

- Los sistemas multimedia han de soportar diversos tipos de aplicaciones simultáneamente:
 - los sistemas operativos utilizan diferentes algoritmos de planificación.
 - Con requisitos de ejecución “duros”
 - EDF, RateMonotonic, ...
 - Con requisitos de ejecución “blandos”
 - Fair Scheduling, Lottery, Reserves, ...
 - En modo “best-effort”
 - Time-sharing.

Técnicas de Planificación de la CPU para MM (III)

- Time-Sharing.
 - El tiempo de CPU se divide en rodajas o slots, que se van asignando por turnos a las tareas existentes en el sistema. Se consigue una distribución equitativa de la CPU.
 - Unix utiliza un esquema Round-Robin con pesos asignados por utilización previa de la CPU.
 - Existe una cola de procesos multinivel de modo que se ejecutan los procesos en el nivel más alto.
 - Los procesos van subiendo de nivel conforme llevan más tiempo sin ser atendidos por la CPU.

=>No válida en general para procesos multimedia pues no tiene en cuenta sus requisitos de ejecución.

Técnicas de Planificación de la CPU para MM (IV)

- Rate-Monotonic
 - Para planificar tareas periódicas, asigna la CPU al proceso cuyo periodo de ejecución sea menor, por medio de una prioridad fija.
 - Fácil de implementar y soportada por muchos sistemas operativos (Unix, Solaris, Windows NT, ...).
 - Técnica ampliamente aceptada en el mundo del Sistemas de Tiempo Real.
- EDF (Earliest Deadline First)
 - Cada tarea tiene un plazo de ejecución determinado (e.g. Para un proceso multimedia el final del periodo de *playback* actual).
 - Se asigna la CPU a aquel proceso cuyo plazo de ejecución esté más próximo.
 - Existe un test de admisión por el que se puede conocer a priori, dada la carga en el sistema, si pueden perderse o no plazos de ejecución.

Técnicas de Planificación de la CPU para MM (V)

- Fair Scheduling/Proportional Share Resource Allocation (se ha demostrado útil para appl. MM.)
 - Se asignan pesos a las tareas en función de su importancia en el sistema (*ticketing*)
 - Los procesos multimedia reciben CPU en función del rate (Bits/seg) de datos que tienen que procesar.
 - El quantum es ajustable para conseguir una granularidad más fina.
- Reservas de Procesador (flexible en su utilización).
 - El programador especifica una reserva de tiempo de cómputo para un proceso (e.g. 20 ms cada 100ms).
 - Dependiendo de la reserva que se efectúe el comportamiento de la aplicación y las garantías obtenidas serán diferentes (best-effort -> “blandas” -> “duras”).

Sistemas de Ficheros para Multimedia (I)

- Los sistemas de ficheros multimedia han de soportar tamaños de ficheros muy grandes debido al gran volumen de los datos.
- Selección del tamaño de bloque del sistema de ficheros (fs block):
 - Si es pequeño se aprovecha bien el espacio de disco (fragmentación baja), pero la tasa de transferencia disminuye.
 - Si es grande la tasa de transferencia aumenta pero existe mayor fragmentación.

Sistemas de Ficheros para Multimedia (II)

- Con objeto de minimizar los tiempos de acceso a disco se han desarrollado distintas políticas de alojamiento (*data placement*):
 - *Alojamiento Contiguo*: todos los bloques de un mismo fichero se alojan de forma consecutiva en el disco => mejora el tiempo de acceso al fichero, si bien causa mayor fragmentación en el disco.
 - *Alojamiento localmente contiguo*: los ficheros se dividen en secciones cada una de las cuales se aloja de forma contigua => mejora el tiempo de acceso aunque hay algunas búsquedas intermedias y disminuye la fragmentación.
 - *Alojamiento con restricciones*: los bloques de un fichero se alojan de forma aleatoria (previene fragmentación) pero entre ellos se define una distancia máxima en número de tracks.
 - *Alojamiento en bloques de duración fija*: los bloques se corresponden con los datos para reproducir un intervalo de tiempo concreto (e.g. 1 seg.)

Políticas de planificación de disco para SSMM (I)

- Políticas clásicas:
 - Shortest-Seek-Time-First (SSTF)
 - gran rendimiento en términos de throughput, puede haber *starvation*
 - SCAN
 - rendimiento inferior pero aún alto, evita el *starvation*
- Políticas para SSMM:
 - EDF-SCAN
 - las peticiones con el mismo plazo se atienden en barrido.
 - EDF-JIT
 - las peticiones con plazos se sirven con EDF y el tiempo de holgura se aprovecha para barrer en dirección a la petición con plazos.

Políticas de planificación de disco para SSMM (II)

- GSS - Group Sweeping Strategy (Barrido por Grupos)
 - Las peticiones se dividen en grupos que se atienden en round-robin, cada grupo se barre con SCAN.
- Planificación por Rondas
 - Se define un periodo de recuperación de disco T de duración fija (e.g. 1 seg.), los ficheros multimedia se dividen en bloques cuya reproducción coincide con el periodo de reproducción (en función del bitrate del stream el tamaño de los bloques diferirá para cada uno de ellos).
 - En cada periodo T se leen de disco los bloques correspondientes a la reproducción en T para cada uno de los streams siguiendo la política SCAN.
 - Test de admisión: se puede aceptar la entrada de un nuevo cliente si se cumple la siguiente condición:
 - $\text{Tiempolecturadisco}(\text{los bloques de } n \text{ streams}) \leq T$.

Transferencias de Datos (I)

- La grabación y reproducción de audio/video clips implica una alta interacción entre la aplicación y los dispositivos de E/S, debido a la gran cantidad de datos a transferir.
 - Los datos han de copiarse del espacio de direcciones de usuario al núcleo y viceversa.
- Los manejadores de dispositivos (drivers), residen en el núcleo e incluyen:
 - captura de audio/video (convertidores A/D, compresión, ...)
 - controladores de disco (SCSI-I, SCSI-II, Fast SCSI, Wide SCSI, IDE, ...).
 - Interfaces de red (e.g. Tarjetas Ethernet, RDSI, ...)
 - Bufferes de almacenamiento temporal.

Transferencias de Datos (II)

- La transferencia de datos propiamente dicha implica:
 - control y sincronización: inicio, sincronización productor/consumidor, control del flujo de datos.
 - Especificación del buffer: dónde se copian los datos en el espacio de direcciones de usuario.
 - Transferencia de datos: bien copia o bien un “remapping” de los bufferes de memoria.
- => necesidad de disminuir el overhead de transferencias de datos evitando copias innecesarias entre buffers (e.g. Del buffer del sistema operativo al buffer a nivel de aplicación).

Transferencias de Datos (III)

- Distintos Modos de Transferencia de Datos:
 - Hardware *streaming*.
 - Hw especializado que se encarga de la transferencia de datos entre los dispositivos.
 - DMA-DMA *streaming*
 - no requiere hardware especializado
 - el procesador no “toca” los datos.
 - OS *streaming*.
 - Los datos son manipulados en el espacio de direcciones del núcleo.
 - User-level *streaming*:
 - los datos son manipulados en el espacio de direcciones de usuario (por lo que se requiere una copia de datos)
 - implica cruzar la barrera de protección del núcleo.
 - => mayor overhead al requerir la intervención de la aplicación.