

-

Problemas con los que se enfrenta DMT

Son numerosos los problemas que debe de resolver DMT y en general un Multitasker para que el nuevo entorno multitarea funcione correctamente y no decepcione al usuario final. Podemos hacer dos clasificaciones de los problemas a los que se enfrenta, problemas respecto al hardware y problemas respecto al software.

• Problemas respecto al hardware

Cuando varias tareas se ejecutan concurrentemente en una misma máquina surgen diversos problemas a la hora de interactuar con el hardware, ya que muchas tareas intentaran acceder al mismo tiempo a un componente hardware y esto, en la mayoría de los casos, produciría un error fatal.

• El problema de la pantalla

Supongamos que tenemos dos tareas, tarea A y tarea B, ejecutándose de forma concurrente en un Multitasker y en la tarea A tenemos cargado nuestro editor de textos favorito, donde estamos escribiendo un documento. Si en la tarea B tenemos cargado un programa que nos muestra todos los archivos que tenemos en el disco duro, seguramente ambas tareas accederán a la pantalla continuamente escribiendo las teclas pulsadas por el usuario, en la tarea A, y escribiendo nombres de archivos, en la tarea B. Lo más seguro es que el usuario no desee esta situación ya que él deseará ver por pantalla sólo el documento que está escribiendo en la tarea A y que la tarea B siga ejecutándose, pero que no acceda a la pantalla para no ensuciar lo que él está escribiendo.

Para solucionar este problema DMT sólo permite que la tarea que está en primer plano pueda acceder libremente a la pantalla. Para el resto de tareas que están en segundo plano, se ha de detectar cuando una de estas tareas intentan leer/escribir en la pantalla y se ha de redireccionar este acceso de lectura/escritura a una pantalla virtual que poseerá cada tarea en segundo plano. Cuando el usuario desee pasar a primer plano una tarea que está en segundo plano, para ver que está sucediendo en dicha tarea, DMT se encargará de cambiar el contenido de la pantalla física por el contenido de la pantalla virtual de la tarea a la que se va a pasar a primer plano.

Hasta ahora se ha solucionado qué es lo que hay que hacer cuando una tarea en segundo plano intenta leer/escribir en la pantalla pero... ¿Qué pasa si una tarea en segundo plano modifica la resolución de la pantalla?. Seguramente el usuario no deseará que una tarea en segundo plano le cambie la resolución de la pantalla, el modo de vídeo o incluso la paleta de colores de la tarea en primer plano. Este problema surge cuando una tarea en segundo plano modifica los registros internos de la tarjeta de vídeo, por lo que DMT se ha de encargar de detectar cualquier acceso a los registros de la tarjeta de vídeo por tareas en segundo plano y guardar todos estos accesos en unos registros virtuales de cada tarea, que DMT utilizará para cargar los registros de la tarjeta de vídeo cuando se pase a primer plano una de estas tareas.

Algunos Multitaskers como DESQView y Windows permite la visualización de varias tareas por pantalla en forma de ventanas independientes. Para poder realizar esta función habrá que redireccionar la interrupción de vídeo de la BIOS a una rutina propia que no imprima en pantalla, sino que guarde todos los accesos en un buffer que tendrá asignado cada tarea y que utilizará para recalculas las posiciones de pantalla dentro de la ventana que le corresponde. En el caso de que una tarea acceda a la pantalla sin utilizar las funciones de la BIOS habrá que detectar cualquier acceso a la RAM de vídeo y recalculas la posición del acceso para que vaya a para dentro de los límites de su propia ventana. Esta opción no está disponible en DMT.

En el capítulo 8 Así se ha implementación DMT se describe detalladamente como se ha solucionado el

problema de la pantalla.

- **Un sólo teclado para varias tareas**

Otro problema a tener en cuenta es el que nos ofrece el teclado en el nuevo entorno multitarea, ya que se ejecutarán varias tareas a la vez y seguramente desearemos que sólo una tarea, la que está en primer plano, tenga el control total de éste. Supongamos que tenemos nuevamente dos tareas ejecutándose, tarea A y tarea B, y que la tarea A se encuentra en primer plano, por lo que desearemos que cada vez que se pulse una tecla por parte del usuario vaya a parar a esta tarea. Si la CPU está atendiendo a la tarea A y el usuario pulsa una tecla se producirá una interrupción de teclado que irá a parar a la tarea que está siendo atendida, en este caso la tarea A. Si ahora la CPU atiende a la tarea B, que está en segundo plano, y el usuario vuelve a pulsar una tecla se producirá nuevamente una interrupción de teclado que irá a parar a la tarea B, lo cual no es deseable debido a que el usuario no verá reflejada la tecla pulsada en la tarea A y tendrá que pulsar nuevamente la misma tecla para que sea recogida por la tarea A.

El problema anterior surge porque la CPU no puede acceder al espacio de direcciones de una tarea que no sea la que está atendiendo actualmente, por lo que no puede procesar la rutina de interrupción de teclado sobre la tarea que desee.

En el capítulo 8 se describe como se ha solucionado este problema y que otras soluciones más eficientes pero menos deseables se pueden llevar también a cabo.

- **Problemas con el ratón**

El ratón ofrece el mismo problema que el teclado, ya que si el usuario está pulsando un botón del ratón o simplemente lo está moviendo se producirán diversas interrupciones de ratón que no siempre irán a parar a la tarea en primer plano, por lo que el usuario no verá moverse el puntero del ratón por pantalla de una forma suave y precisa.

El ratón ofrece otro problema y es precisamente cuando varias tareas hacen uso de él. Supongamos que tenemos la tarea A en primer plano y tenemos el puntero del ratón en la coordenada (XA,YA), ahora pasamos a primer plano la tarea B y movemos el puntero del ratón a la posición (XB,YB), si ahora nuevamente pasamos a primer plano la tarea A desearemos que el puntero aparezca en la posición que tenía la tarea A antes de cambiar a segundo plano, es decir, queremos que el puntero aparezca en la coordenada (XA,YA) y no en (XB,YB). Además, el usuario final deseará que no sólo se restaure la posición del ratón entre cambios de tareas a primer plano, sino que se restauren todas las características que poseía como tamaño del cursor, velocidad, etc.

En el capítulo 8 se describe cómo se ha solucionado el anterior problema.

- **Serialización en las operaciones de Disco**

El disco es un componente bastante delicado bajo DMT ya que si el procesador produce una conmutación de tarea cuando se estaba realizando una operación de disco, se puede llegar a pérdidas graves de datos en el disco. El problema anterior surge porque las operaciones básicas de disco ofrecidas por la BIOS están realizadas para un entorno monotarea. Imaginemos que tenemos una tarea que quiere copiar 1000 bytes que hay en la posición de memoria 2A00:0000 a disco, para realizar tal operación llamará a una función de la BIOS para que realice la transferencia de memoria a disco. La función de la BIOS indica al controlador de disco que copie los 1000 bytes desde la posición de memoria anterior al disco, si justo cuando se está realizando la transferencia se produce una conmutación de tarea por parte de la CPU para atender a otra tarea, el controlador de disco seguirá leyendo bytes desde la posición de memoria anterior pero ahora desde la memoria de la nueva tarea, con lo que probablemente se escribirá basura en el disco.

Debido al problema que surge con el disco, debemos de disponer de un mecanismo que asegure que las operaciones de disco no se puedan interrumpir, lo que supondrá ciertas pérdidas de tiempo que se podrían aprovechar para atender a otras tareas.

- **Problemas con el DMA (Acceso Directo a Memoria)**

Para conseguir los objetivos que se propone DMT, tener varios programas DOS en memoria, necesitamos un mecanismo de paginado de memoria para dar la ilusión a cada uno de estos programas que se ejecutan en el primer Mbyte de memoria física, es decir, dar la ilusión a cada programa que se ejecuta sólo en memoria bajo el control de MS-DOS. Si una tarea accede a una posición de memoria determinada, por ejemplo la dirección 4563:1002, el mecanismo de paginado la convertirá en otra dirección que referenciará a la dirección física final, por ejemplo 1000:0233. Se puede decir que las tareas utilizan *direcciones lógicas* de memoria y tras el mecanismo de paginado se convierte en una *dirección física* de memoria.

El uso del DMA, para accesos a discos, disquetes, tarjeta sonido..., supone un gran problema respecto al mecanismo de paginado. Esto es debido a que el DMA no comprueba si la paginación está activada o no y cree que todas las direcciones que le pasan las diversas tareas son físicas y no lógicas. Esto supone un gran inconveniente ya que el DMA leerá y escribirá en posiciones de memoria que seguramente se corresponderán a otra tarea o peor incluso, escribirá en la memoria utilizada por DMT y producirá, por tanto, una caída del sistema.

Para solucionar el problema del DMA deberemos de detectar todo acceso a este *chip* y convertir las direcciones lógicas en físicas antes de pasárselas al DMA, permitiendo, ahora sí, la utilización de este chip bajo el nuevo entorno multitarea.

En [TISCHER] y [STALKER] se puede encontrar más información sobre el DMA.

En el capítulo 8 se describe cómo se puede detectar cada acceso al DMA y cómo se convierten las direcciones lógicas en físicas antes de acceder a él.

- **Problemas respecto al software**

En esta sección vamos a describir qué inconvenientes nos encontramos para soportar al MS-DOS y sus programas bajo DMT y qué gran inconveniente existe para lograr el máximo rendimiento de la CPU bajo MS-DOS.

- **Protección entre los programas DOS**

El MS-DOS es un sistema operativo monotarea que no ofrece ninguna protección frente a los programas que ejecuta, así cualquier programa ejecutado bajo él puede acceder a cualquier dirección de memoria, puede interceptar cualquier interrupción y manejarla a su gusto, puede manipular cualquier puerto de E/S, puede cambiar las estructuras internas del DOS, etc. Toda esta libertad que se le ofrece a los programas DOS es debido a que no utiliza los mecanismos de protección que ofrecen los microprocesadores más modernos, ya que desea mantener la compatibilidad con sus antecesores. Ante esta situación, DMT debe de intentar seguir dando toda esta libertad a cada programa DOS para que su funcionamiento sea el que tenía previsto, pero eso sí, detectar cuando una tarea puede afectar a la integridad del sistema global y eliminarla de la memoria si es necesario.

Como cada tarea posee una copia entera del MS-DOS, se permite que realice cualquier tipo de barbaridad sobre su copia del DOS, pero no sobre la zona de memoria correspondiente a otra tarea o a la memoria utilizada por DMT. Para lograr este objetivo debemos de dividir la memoria en particiones independientes de forma que una tarea sólo pueda ver su correspondiente partición sin poder afectar al resto de las tareas y

exista, por tanto, una pacífica coexistencia de todas las tareas en memoria.

En el capítulo 4 se explican los mecanismos que ofrece el 80386 y sucesores para lograr la protección entre las diversas tareas ejecutadas en memoria.

- **Incapacidad de lograr el máximo rendimiento de la CPU**

Para lograr que la CPU esté trabajando del 90 al 100 por cien en el nuevo entorno multitarea debemos de aprovechar los tiempos de espera que se producen al acceder a un dispositivo de E/S. Cuando una tarea está esperando que se pulse una tecla por parte del usuario, que el disco empiece a enviar datos, etc. la CPU se queda esperando tales eventos bajo diversos *bucles*, lo que resulta una pérdida de ciclos de reloj que podrían ser aprovechados para atender a otra tarea y lograr un mayor rendimiento por parte de la CPU.

Para que un Multitasker pueda lograr este objetivo, máximo el rendimiento de la CPU, necesita saber que es lo que está realizando el MS-DOS en un momento dado, es decir, deberá de saber cuando el MS-DOS está en un bucle de espera para un acceso a disco, esperando la pulsación de una tecla, etc. y así aprovechar el Multitasker para cambiar de tarea y evitar esas innecesarias esperas. El problema que existe para lograr tal objetivo es que el MS-DOS no posee ningún mecanismo que indique que es lo que está realizando en un momento dado, con lo que el Multitasker no puede enterarse si la CPU se encuentra en uno de los innecesarios bucles de espera y realizar una posible conmutación de tarea.

La solución para convertir al MS-DOS en multitarea sería la de emular todos los servicios que ofrece y adaptarlos para el nuevo entorno multitarea. El principal problema sobre esto es que emular todos los servicios del DOS requeriría un arduo trabajo, ya que habría una gran dependencia de la máquina y para realizar esa labor mejor se opta por realizar un nuevo sistema operativo con todas las características multitarea.

Basic Input Output System

Ver capítulo 4

Problemas con los que se enfrenta DMT 11

6