

Pauta Control 6

Profesor: Yadran Eterovic

Pregunta 1 (3 puntos)

En un computador x86 de 16 bits se desea implementar un mecanismo que permita al sistema operativo terminar programas que llevan mucho tiempo en ejecución, e informarle al usuario de esto mediante un mensaje en la pantalla. Para esto, se considera la siguiente tabla que indica los vectores de interrupción:

Dirección del vector	Tipo	Función asociada
00-01	Excepción	<i>Exception handlers</i>
02	Excepción	Usada para errores críticos del sistema, no enmascarable
03-07	Excepción	
08	IRQ0	<i>Timer del sistema</i>
09	IRQ1	Puerto PS/2: Teclado
0A	IRQ2	Conectada al PIC esclavo
0B	IRQ3	Puerto serial
0C	IRQ4	Puerto serial
0D	IRQ5	Puerto paralelo
0E	IRQ6	<i>Floppy disk</i>
0F	IRQ7	Puerto paralelo
10	Int. de <i>Software</i>	Funciones de video
11-6F	Int. de <i>Software</i>	Funciones varias
70	IRQ8	<i>Real time clock (RTC)</i>
71-73	IRQ9-11	No tienen asociación estándar, libre uso
74	IRQ12	Puerto PS/2: <i>Mouse</i>
75	IRQ13	Coprocesador matemático
76	IRQ14	Controlador de disco 1
77	IRQ15	Controlador de disco 2
78-FF	Int. de <i>Software</i>	Funciones varias

¿Qué dispositivos de I/O de los descritos en la tabla utilizaría para implementar el mecanismo? Indique qué función cumpliría cada uno.

Solución:

En primer lugar se necesita del *Timer* del sistema para contar el tiempo que lleva un programa en ejecución y dentro de su *ISR* poner la lógica para terminar los programas que llevan mucho tiempos en ejecución.

Además se requiere de la pantalla, que se utiliza mediante el trap de funciones de video.

Asignación de puntaje:

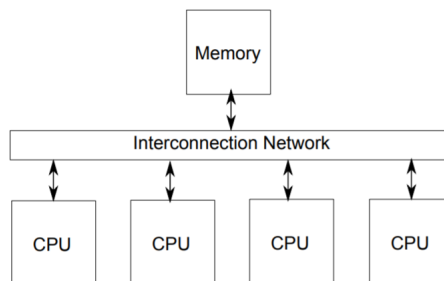
- 1.5pts por indicar función y dispositivo de forma correcta

1. Pregunta 2 (3 puntos)

- (a) Enuncia de manera clara y precisa el problema de coherencia de cache.
- (b) Da un ejemplo de aplicación que muestre claramente la ocurrencia del problema.
- (c) Describe un mecanismo simple pero eficiente para resolver el problema.

Solución:

- (a) El problema de coherencia de cache se enuncia como la posibilidad de que la información de una misma variable pueda quedar almacenada de manera inconsistente en el sistema (esto en un contexto de multiprocesador con memoria compartida).
- (b) Suponga múltiples procesadores con una memoria compartida como se muestra en el diagrama:



Digamos que todas las CPU tienen política de escritura *Write-Back*, y la CPU de más a la izquierda lee y luego escribe en la dirección 10 de la Memoria donde se ubica una variable global para cualquier programa. Al mismo la CPU de más a la izquierda escribe en la misma dirección pero un dato diferente. En este caso para una misma variable las dos memorias cache que almacenan el mismo dato pero con valores distintos, lo que genera una inconsistencia en todo el sistema, lo que puede significar en operaciones erróneas para el resto de las CPU.

- (c) Un mecanismo simple y eficiente para resolver el problema es el protocolo MESI. En este sistema cada bloque de memoria puede estar en un estado Modificado (M), Exclusivo (E), Compartido (S) o Invalido (I). Luego se tiene lo siguiente (0,2 por cada punto):
 - Cuando la CPU parte, todas las entradas de su caché están en I.
 - La primera vez que se lee la memoria, la línea referenciada se lleva a la caché de la CPU que hizo la lectura y se la marca como E.
 - Si otra CPU lee la misma línea (a su caché), avisa una copia y ambas son marcadas como S. Por otro lado si esta CPU escribe en esta línea, las otras CPUs descartan sus copias y su propia copia queda en estado M (si una línea está en estado E cuando es escrita, no es necesario invalidar las otras cachés).
 - Si una tercera CPU lee la línea, la CPU anterior (que tiene la línea actualizada) le pide que espere mientras actualiza la memoria, cuando la memoria queda actualizada, esta tercera CPU lee la línea y ambas CPUs marcan la línea como S. Si cualquiera de estas CPUs escribe la línea nuevamente, invalida la copia en la caché de la otra y deja la propia como M.

- Si ahora la primera CPU escribe una palabra de esta línea (en la memoria) la CPU que tiene la línea en estado M le pide que espere mientras la escribe en la memoria al terminar, esta CPU marca su línea como I. La primera CPU, entonces, va a escribir una línea que no está en ninguna caché.

NOTA: Si se explica el protocolo MSI de los apuntes también obtiene puntaje completo.

Asignacion de puntaje:

- 1 punto por cada letra correcta.