

Tema 9: Jerarquía de memoria

1.1 Memoria caché

- Baja capacidad, alta velocidad
- CPU moderna → diferentes niveles de caché.

{ Hit: dato se encuentra.
Miss: dato no se encuentra.

Fallos:

- Obligorio: el primer acceso a cualquier artículo es una falta.
- Por capacidad: expulsar artículo para hacer sitio a otros.
- Por conflicto: el esquema de mapeo desplaza artículos útiles para desplazar otros artículos → fallos en el futuro.

1.2 Principio de localidad

- localidat espacial: cuando se acceda a un lugar se accede también a los lugares cercanos.
- localidat temporal: accesos a una misma dirección agrupados en el tiempo.

$$\text{Hit Rate} = \frac{\text{Hit Accesses}}{\text{Total accesses.}}$$

1.3 Bloques de caché (tamaño de una potencia de 2)

Tamaño de bloque: número de palabras.

- Pequeño: transferencia más rápida, disminuye hit rate
- Grande: más datos → mejores hit rate → entran datos poco útiles.
- secuencial: mejor aumentar el tamaño del bloque.

1.4 Acceso a la información.

- Acceso de lectura: el procesador demanda información.
 - Aciertos (hits): la información se transfiere desde la caché.
 - Fallos (misses): transferencia de información desde la memoria central.

Técnica direct-load: transferencia a CPU y caché en paralelo.

- Acceso de escritura: con

- Aciertos:

- * Write-Through: la modificación se realiza en ambas memorias.

- * Write-Back: el bloque se marca como modificado, se actualiza y se sustituye.

Fallos:

- Escritura directa en la memoria central. (Asignación sin escritura)
- Carga de bloques en la caché (Asignación de escritura)

1.5 Prefetching

- Ocultar latencia de memoria cuando hay un fallo
- carga bloques en caché antes de ser solicitados.

Enfoques:

- Prefetch on miss: recupera el bloque $i+1$ cuando al acceder a i da fallo.

Útil: evita fallo
Inútil: sustituido antes de ser usado
Perjudicial.

1.6 Organización de la información de la memoria caché

Bits de control:

- Etiqueta: marca identificación en el directorio.
- Datos válidos: indica si la información puede ser utilizada o no.
- Datos modificados: indica si se ha de reescribir la información antes de reemplazar la memoria en la caché

Bits de información: datos o instrucciones.

1.7 Correspondencia: relación entre bloques de caché y bloques de memoria.

- Caché completamente asociativa: Principal $\rightarrow N$ bloques
Cache $\rightarrow M$ bloques
- Hardware caro \uparrow Hit rate
- Muy lento.

- Caché de mapeo: cada bloque tiene una posición fija en la caché

Bloque de caché: (Bloque de memoria) $\text{MOD } (M \text{ bloques caché})$

- Caché asociativa por conjuntos: cada bloque tiene un conjunto predeterminado.

1.8 Algoritmos de reemplazo

- Aleatorio
- FIFO
- (LRU) Menos usado recientemente

2 Diseño de Sistemas de memoria para memoria caché.

2.1 Memorias de gran ancho de palabra:

\downarrow transferencia \uparrow tamaño de palabra Más caro

2.2 Memorias entrelazadas:

Palabras consecutivas \rightarrow diferentes módulos \downarrow latencia

3 Memoria virtual

Memoria principal \rightarrow caché para memoria secundaria

3.2 Direcciones virtuales y físicas: rango de direcciones mayor.

Programas \rightarrow direcciones virtuales Memoria \rightarrow direcciones físicas.

Memoria (mapeo): correspondencia entre direcciones físicas y virtuales.

3.3 Memoria paginada:

- Dirección: n página virtual y desplazamiento de página.
- Rango de direcciones \rightarrow dividido en páginas.
- Marco de página \rightarrow memoria física suficientemente grande para contener una página.

3.4 Memoria segmentada:

Rango de direcciones: se divide en segmentos de tamaño variable

3.5 Políticas de sustitución.

Seleccionar que página reemplazar y cual recuperar

Algoritmos $\begin{cases} \text{FIFO} \\ \text{LRU} \end{cases}$

3.6 Protección de la información.

Un programa no debe de poder modificar datos o instrucciones de otro.