

Calcule el espacio de memoria necesario de una tabla de página de un nivel para un espacio de direcciones de 16bits, 32bits, 48bits y 64bits. Asuma que cada entrada de página es del mismo tamaño que el espacio de direcciones (Ej. 16bits = 2bytes). Tamaño de página de 8kB.

$$n^{\circ} \text{ pag} = \frac{\text{Tamaño}_{total}}{\text{Tamaño}_{pagina}}$$

$$\text{Tamaño}_{tabla} = n^{\circ} \text{ entradas} \cdot \text{Tamaño}_{entrada}$$

16bits:

Tamaño total: 2^{16}

Tamaño página: 2^{13}

n° de páginas: $2^{16} / 2^{13} = 2^3 = 8$ páginas

Tamaño de tabla: $2^3 \cdot 2B = 16B$ por tabla

32bits:

Tamaño total: 2^{32}

Tamaño página: 2^{13}

n° de páginas: $2^{32} / 2^{13} = 2^{19}$ páginas

Tamaño de tabla: $2^{19} \cdot 4B = 2MB$ por tabla

48bits:

Tamaño total: 2^{48}

Tamaño página: 2^{13}

n° de páginas: $2^{48} / 2^{13} = 2^{35}$ páginas

Tamaño de tabla: $2^{35} \cdot 6B = 192GB$ por tabla

64bits:

Tamaño total: 2^{64}

Tamaño página: 2^{13}

n° de páginas: $2^{64} / 2^{13} = 2^{51}$ páginas

Tamaño de tabla: $2^{51} \cdot 8B = 2PetaBytes$ (2048 TeraBytes) por tabla

¿Qué pasa si se utilizan tablas de página de segundo nivel?

¿Cuál es el consumo de memoria de cada tabla de página?

¿Porqué requieren menos espacio?

Repetir los cálculos para espacio de direcciones de 32bits, 2 niveles de página, tamaño de página de 4kB.

Tamaño de página de 4kB -> offset de 12 bits

N° de entradas: 2 niveles -> (32-12) bits / 2 -> 10 bits por índice => 2^{10} entradas.

$$\text{Tamaño}_{tabla} = n^{\circ} \text{ entradas} \cdot \text{Tamaño}_{entrada}$$

tamaño por tabla: $2^{10} \cdot 4B = 4kB$ por tabla.

n° de tablas de página de nivel 2 = n° de entradas de tabla de nivel 1 = 2^{10}

Tamaño total = Tamaño tabla nivel 1 + Σ Tamaño tablas de nivel 2 = 4,004MB

Si con un nivel de tabla de página se usa con páginas de 4kB se utilizan 4MB, ¿porqué es más eficiente utilizar 2 niveles en vez de sólo uno?

Un procesador utiliza un sistema de memoria virtual con paginación, en tres posibles configuraciones:

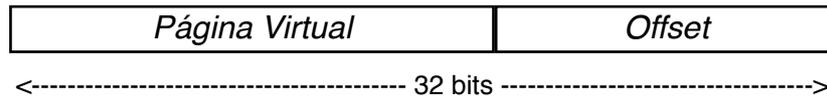
T1: Tabla de páginas de 1 nivel.

T2: Tabla de páginas de 2 niveles, con igual cantidad de entradas en las tablas de nivel 1 y 2.

T3: Tabla de páginas invertida.

Las direcciones virtuales son de 32 bits, el tamaño de la página es de 4 KBytes y el tamaño de la memoria física (real) es de 256 MBytes.

a) ¿Cuántos bits se utilizan para expresar el offset dentro de una página, el número de página virtual y el número de página real (marco de página)?



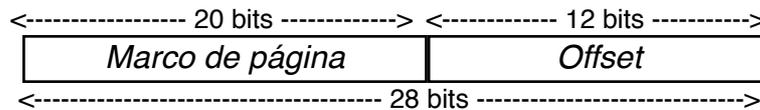
Offset = n° de entradas de página

$$n^{\circ} \text{ entradas} = \frac{\text{Tamaño}_{\text{página}}}{\text{Tamaño}_{\text{entrada}}}$$

4 kB = 2^{12} bytes.

Luego, se utilizan 12 bits para el offset.

Dirección REAL:



$$n^{\circ} \text{ marcos de página} = \frac{\text{Tamaño}_{\text{memoria}}}{\text{Tamaño}_{\text{página}}}$$

$256 \text{ MBytes} / 4\text{kB} = 2^{28} / 2^{12} = 64 * 2^{10}$ marcos de página.

Hay 2^{16} marcos de página.

Luego, se utilizan 16 bits para el número de página real.

Se utilizan los bits restantes del offset como página virtual: 20 bits.

Dirección REAL:



b) Calcule la cantidad de Bytes necesarios para cada entrada en la tabla de páginas en cada una de las tres alternativas anteriores. Asuma que los bits de validez y protección se almacenan en 1 Byte, e ignore el espacio para almacenar la dirección de bloque donde reside la página en disco en caso de encontrarse en memoria. Su resultado debe expresarse en múltiplos enteros de 1 Byte. Para el caso de T2, debe indicar el tamaño de la entrada para la tabla de primer nivel y para la tabla de segundo nivel.

Para T1:

1 Byte de protección y validez

2 Bytes para indicar marco de página real (16 bits)

Total: $1+2 = 3$ Bytes por entrada

Para T2:

Nivel 1: 1 Byte de validez

2 Bytes por marco de página real

Total: 3 Bytes por entrada

Nivel 2: 1 Byte de validez y protección

2 Bytes por marco de página real

Total: 3 Bytes por entrada

Para T3:

1 Byte de protección y validez

2 Bytes para indicar marco de página real (16 bits)

3 Bytes para indicar página virtual (20bits)

Total: $1 + 2 + 3 = 6$ Bytes por entrada

c) Calcule la cantidad de páginas en memoria real necesarias para almacenar la tabla de páginas en cada una de las tres alternativas. Para T2, asuma que un 20% de las tablas de segundo nivel se encuentran presentes en memoria real, y recuerde que las tablas se almacenan en páginas de memoria individuales.

Para T1:

Tamaño por entrada * entradas = 3 Bytes * $2^{20} = 3$ MB

Para T2:

Nivel 1: Tamaño por entrada * entradas: 3Bytes * $2^{10} = 3$ kB.

Nivel 2: Tamaño por entrada * entradas: 3Bytes * $2^{10} = 3$ kB.

Total: Tamaño tabla Nivel 1 + $0.2 * 2^{10} * \text{Tamaño tabla Nivel 2} = 3\text{kB} + 0.2 * 2^{10} * 3\text{kB} \approx 208$ kB

Para T3:

La memoria está llena, luego hay 2^{16} entradas

Tamaño por entrada * entradas = 6Bytes * $2^{16} = 384$ kB

d) Considere que el sistema anterior tiene un TLB con una tasa de fallos del 1%, y un tiempo de acierto de 1 ciclo. La traducción se inicia simultáneamente en el TLB y en las tablas de páginas. Cada acceso a memoria real toma 20 ciclos por palabra de 4 Bytes. Calcule el tiempo promedio (en ciclos de reloj) necesarios para traducir una dirección virtual a una dirección real (asumiendo que no hay fallos de página). Asuma (sólo para esta parte del ejercicio) que una entrada en cualquiera de las tablas T1, T2 y T3 cabe en una palabra de 4 Bytes. Asuma también que una búsqueda asociativa en la tabla de páginas invertida requiere de 3 accesos a memoria en promedio.

$$T_{traduccion} = T_{TLB} + \% fallo_{TLB} \cdot T_{accesomem}$$

Para T1:

Se produce un acceso a tabla de página,

Total: 20 ciclos

$$T_{Trad-T1} = 0.99 \cdot 1 + 0.01 \cdot 20 = 1,01 \text{ ciclos}$$

Para T2:

Se produce un accesos para la tabla de nivel 1, otro para la tabla de nivel 2.

Total: 40 ciclos.

$$T_{Trad-T2} = 0.99 \cdot 1 + 0.01 \cdot 40 = 1,03 \text{ ciclos}$$

Para T3:

Se producen 3 accesos para la búsqueda asociativa de páginas invertida.

Total: 60 ciclos

$$T_{Trad-T3} = 0.99 \cdot 1 + 0.01 \cdot 60 = 1,05 \text{ ciclos}$$

Ejercicio en Clases:

$t_{CPU} = 0.5ns$

Bloque L1 = 64 bytes, latencia I-L1 = 1 ciclo, latencia D-L1 = 2 ciclos

Tasa fallos L1 = 1% (instrucciones), 4% (datos)

Bloque L2 = 128 bytes, latencia L2 = 4 ciclos + 1/transacción

Tasa fallos local L2 = 10%

Bus L1 -L2: 64 bits; Bus DRAM -L2: 64 bits

DRAM: DIMM PC6400 DDR2 5-5-5-15-T1

30% de load/store

Calcule el tiempo para transferir un bloque de datos de DRAM a L2 (en ciclos de CPU)

Al producirse un fallo en L2, se deben transferir, desde memoria, 128 bytes.

Como el bus es de 8bytes, se deben realizar $128/8 = 16$ transferencias.

La memoria tipo DDR2, puede transmitir, ristas de máximo 8 elementos ("Largo de ristra DDR: 2,4,8; DDR2: 4, 8; DDR3: 8 (posible inhibir4)") por lo que es necesario transmitir 2 ristas.

Velocidad de memoria:

PC6400, 6400 MBytes/seg => $6400/8 = 800$ Mtransferencias/seg

DDR2 envía en canto de subida y de bajada, luego la frecuencia del bus:

$$800/2 = 400MHz$$

Período del Bus:

$$T_{RAM} = 1/(400 MHz) = 2,5 [ns] = 5 * T_{CPU}$$

Para el tiempo de transferencia de datos de la memoria de la memoria al caché, si se deben realizar 16 transferencias, en 2 ristas:

$$T_{RAM-L2} = \text{Tasa de comandos} + t_{RCD} + n^{\circ}_{ristras} * (t_{CAS} + T_{\text{amaño-Ristra}/2} * t_{BUS})$$

$$T_{RAM-L2} = \text{Tasa de comandos} + t_{RCD} + 2 * (t_{CAS} + 4 * t_{BUS})$$

Para la memoria

$$\begin{array}{c} \text{DIMM PC6400 DDR2} \\ 5 - 5 - 5 - 15 - T1 \\ t_{CAS} - t_{RCD} - t_{RP} - t_{RAS} - \text{CommandRate} \end{array}$$

$$T_{RAM-L2} = 5 * t_{CPU} + 5 * (5 * t_{CPU}) + 2 * (5 * (5 * t_{CPU}) + 4 * (5 * t_{CPU}))$$

$$T_{RAM-L2} = 120 \text{ ciclos}$$

Calcule el tiempo necesario para transferir un bloque de datos de L2 a L1 (en ciclos de CPU), asumiendo que el bloque se encuentra en L2.

Al producirse fallo en L1, se deben transmitir 64 bytes. Como el bus es de 8 bytes, se deben realizar 8 transferencias de L2 a L1.

$$T_{\text{acierto-L2}} = 4 \text{ ciclos} + 1/\text{transacción} = 4 * t_{CPU} + 8 * 1 * t_{CPU} = 12 * t_{CPU}$$

Calcule el tiempo necesario para transferir un bloque de L2 a L1, considerando que el bloque puede encontrarse en L2 o DRAM.

$$T_{L2-L1} = T_{\text{acierto}} + \text{tasa}_{\text{falla-L2}} * T_{\text{RAM-L2}}$$

$$T_{L2-L1} = 12 * t_{\text{CPU}} + 0,1 * 120$$

$$T_{L2-L1} = 24 * t_{\text{CPU}}$$

Calcule el tiempo medio de acceso a memoria (AMAT) del sistema de memoria

$$\text{AMAT} = (\text{AMAT}_{\text{fetch}} + 0,3 * \text{AMAT}_{\text{datos}}) / 1,30$$

$$\text{AMAT}_{\text{fetch}} = (1 + 0,01 * 24) = 1,24 \text{ ciclos}$$

$$\text{AMAT}_{\text{datos}} = (2 + 0,04 * 24) = 2,96 \text{ ciclos}$$

$$\text{AMAT} = (1,24 + 0,3 * 2,96) / 1,30 = 1,637 \text{ ciclos}$$

Si el cache L1 tuviese una arquitectura unificada, tendría una latencia de 2 ciclos y una tasa de fallos del 1.5% para instrucciones y datos. ¿Cuál sería el AMAT en este caso?

$$\text{AMAT} = (\text{AMAT}_{\text{fetch}} + \text{AMAT}_{\text{datos}}) / 1,30$$

$$\text{AMAT}_{\text{fetch}} = (2 + 0,015 * 24) = 2,36 \text{ ciclos}$$

$$\text{AMAT}_{\text{datos}} = 1 + \text{AMAT}_{\text{fetch}} = 3,36 \text{ ciclos}$$

Se debe agregar un ciclo por el conflicto estructural que se produce en la caché unificada al hacer un load/store



$$\text{AMAT} = (2,36 + 0,3 * 3,36) / 1,30 = 2,59 \text{ ciclos}$$

Se tiene en un sistema:

Un procesador con Caché L1 de estructura Harvard.

Fallo de instrucciones: 5%

Fallo de datos: 10%

Cantidad de load/store = 15%

Tiempo de acierto: 1 ciclo.

Cache L2 unificado.

Tasa de fallos: 2%

Tiempo de acierto: 3 ciclos.

Memoria DRAM:

Tasa de fallos: 0,01%

Tiempo de acceso: 100 ciclos

Tiempo de acceso a disco duro: 1.000.000 ciclos

Calcule el tiempo de acceso promedio a un dato.

$$AMAT_{RAM} = 100 * 0,9999 + 0,0001 * 1.000.000 = 199,9 \text{ ciclos}$$

$$AMAT_{L2} = 3 * 0,98 + 199,99 * 0,02 = 4,9798 \text{ ciclos}$$

$$AMAT_{L1-IF} = 1 * 0,95 + 0,05 * 4,9798 = 1,19899 \text{ ciclos}$$

$$AMAT_{L1-MEM} = 1 * 0,9 + 0,1 * 4,9798 = 1,39798 \text{ ciclos}$$

$$AMAT_{Total} = (AMAT_{L1-IF} + 0,15 * AMAT_{L1-MEM}) / 1,15 \approx 1,225 \text{ ciclos}$$