Computer Architecture

Fall, 2024
Week 15
2024.12.9

組別：_____ 簽名：_____

[group 13]

1. Which of the following statements about sources of cache misses are correct?

A. Compulsory misses occur because the block is being accessed for the first time.
B. Capacity misses happen because the cache size is finite, and a replaced block is accessed again later.
C. Conflict misses occur even in a fully associative cache of the same total size.
D. Conflict misses arise due to competition for entries in a set in a non-fully associative cache.

Ans:

A、B、D

[group 6]

2.
(a) Memory 中常見的三種 miss 來源與其原因該如何配對?
(1): Compulsory misses A: Due to competition for entries in a set
(2): Capacity misses B: First access to a block
(3): Conflict misses C: Due to finite cache size

(b)T or F
1. We detect the page faults in software and handle them in hardware.
2. To reduce page faults, we use direct mapped replacement
3. In LRU replacement ,we cleared the reference bits in PTE to zero periodically.

Ans:

(a) : (1)B, (2)C, (3)A
(b):
1.F , We detect the page faults in hardware and handle the page faults in software.
2.F , use Fully associative replacement
3.T

[group 3]

3. 虛擬位址佔 16 bit，實體位址佔 12 bit， page 大小為 256 Bytes。根據底下 page table ，將虛擬位址 0x05AC 翻譯成實體位址

Page #	frame #
1	3
2	10
3	5
4	7
5	15
6	2
7	8
8	4

Ans:

page 佔 8 bit，從虛擬位址高位 8 bit 取得 page # = 0x05，翻譯後得到 frame # = 0xF。最終得到實體位址 0xFAC

[group 5]

4. 完成下面的 Challenge in Memory Hierarchy 表格

Design change	Effects on miss rate	Possible effects
size ↑		access time ↑
associativity ↑		access time ↑
block size ↑		miss penalty ↑

Ans:

Design change	Effects on miss rate	Possible effects
size ↑	capacity miss ↓	access time ↑
associativity ↑	conflict miss ↓	access time ↑
block size ↑	spatial locality ↑	miss penalty ↑

[group 11]

5. Determine whether the following statements are true or false. If false, provide an explanation.

1. Page faults occur when the required page is not in physical memory.
2. Demand paging loads all pages of a process into memory at once.
3. FIFO (First-In-First-Out) is a page replacement policy that replaces the oldest page in memory.
4. A process can continue execution even when its required page is not in memory

Ans:

1. True
2. False

Demand paging only loads pages into memory as they are accessed, reducing memory usage and startup latency. This policy is specifically called the “demand load policy.”

3. True
4. False.

When a page fault occurs, the process cannot proceed until the operating system loads the required page into memory. Execution resumes only after the fault is handled

[group 9]

6. True/False:

A. Virtual memory allows each program to have a private virtual address space that is completely independent of physical memory.
B. Page faults occur when the required page is found in the TLB but not in the page table.
C. Using larger page sizes in virtual memory can help reduce the page fault rate but may increase internal fragmentation.
D. The Translation Lookaside Buffer (TLB) is used to speed up virtual-to-physical address translation by caching page table entries.

Ans:

A. True. Virtual memory provides isolation and abstraction by mapping virtual addresses to physical addresses, allowing programs to operate in their own private address spaces.
B. False. A page fault occurs only when the page is not in physical memory, regardless of its presence in the TLB or page table.
C. True. Larger page sizes reduce the number of page faults by amortizing the cost of accessing a page, but they may also leave unused memory within a page, leading to internal fragmentation.
D. True. The TLB is a small, fast cache that stores recent page table entries to reduce the overhead of accessing the page table in main memory.

[group 10]

7. Assume virtual memory addresses are 32 bits and every page is 4KB. Every page entry needs 4 bytes space to store data. Questions: How many page entries and total size of page table?

Ans:

page entries:
Because every page is 4KB, page offset is 12 bit. It means the virtual page number is 32-12 bit long. So there are 2^20 page entries. Total size of page table = page entries * 4 bytes = 4MB.

[group 2]

8. Complete the table.

Cache	TLB	Page Table	Possible? Conditions?
Miss	Hit	Hit	
Hit	Miss	Hit	
Miss	Miss	Hit	
Miss	Miss	Miss	
Miss	Hit	Miss	
Hit	Hit	Miss	
Hit	Miss	Miss	

Ans:

Cache	TLB	Page Table	Possible? Conditions?
Miss	Hit	Hit	Yes; but page table never checked if TLB hits
Hit	Miss	Hit	TLB miss, but entry found in page table; after retry, data in cache
Miss	Miss	Hit	TLB miss, but entry found in page table; after retry, data miss in cache
Miss	Miss	Miss	TLB miss and is followed by a page fault; after retry, data miss in cache
Miss	Hit	Miss	impossible; not in TLB if page not in memory
Hit	Hit	Miss	impossible; not in TLB if page not in memory
Hit	Miss	Miss	impossible; not in cache if page not in memory

[group 4]

9. True or False

(a) TLB miss must be handle in hardware.

(b) TLB hit , page table must hit.

(c) Write-through is the most efficient policy for virtual memory.

Ans:

(a) false , can also be handled in software
(b) true
(c) false , write-back is more efficient

[group 7]

10. Which statement below is correct?
(a) A page table is an array of PTEs.
(b) A frame is the same size as a page.
(c) TLB is located inside the cache.
(d) There is a chance when TLB hits, a page fault occurs.

Ans:

(a) & (b) are correct statements.

[group]

11. Where is TLB(Translation lookaside buffer) ? Why do we need it?

Ans:

Fast Cache in CPU, because access to the page table has good locality, we can take advantage of it to reduce memory access time while translating addresses.