Computer Architecture

Fall, 2019

Week 17

2019.12.30

組別：__________ 簽名：____

[group3] (對抗賽)

1. Please fill the blank below:

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	

A:

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

[group2] (對抗賽)

2. 請選出錯誤的選項，並更正:

   (A) Fetching missing items from disk only on page fault is called demand load policy.

   (B) Page table which stores placement information is in disk.

   (C) Hardware must detect page fault and it can also handle the faults.

   (D) In virtual memory system, page's write policy uses write-through to write data.

   (E) There are twice memory accesses for each address translation.

Ans:

BCD

(A) True

(B) Page table store in main memory.

(C) Handle the faults in software instead of hardware, because handling time is small compared to disk access.

(D) Use write-back, because disk writes take millions of cycles.

(E) True

[group5]

3. If page fault happen, should we use hardware or software to handle it, why?

A: We can use hardware to detect page fault and use software to handle it, since using hardware still need millions of cycle to handle it, and if we use software, we can use some smart algorism to solve it.

[group11] (對抗賽)

4. The following statement, please choose which are false, and why it is false.
   (A) There is only one memory reference for each memory request.
   (B) To reduce page fault rate, prefer LRU replacement. A page with reference bit=0 means that has not been used recently.
   (C) Disk can use write-through and write-back.
   (D) It is possible for cache is hit, TLB is miss and page table is miss.

ANS:

(A)There are two memory reference for each memory request.

(C)Disk can use write-back. Write through is impractical.

(D) It is impossible for cache is hit, TLB is miss and page table is miss. Not in cache if page not in memory.

[group7] (對抗賽)

5. True or False? Explain why.

A. Cache Miss, TLB Hit and Page table Hit is an impossible combination Because the Entry must be in Cache before it is in TLB.

B. To implement LRU in an 8-way associative scheme, it suffices to keep one Reference bit.

C. As write-through is too inefficient for virtual memory system, write-back is the only acceptable policy for writing in TLB.

ans:

A. False, this combination could happen if the page might have been brought to memory and discarded or replaced

B. False, LRU can only be implemented with 1-bit Reference for 2-way associative memory

C. True, writing too disk is too expensive, we only write on page replacement if the page has been written (indicated by Dirty Bit).

[group9] (對抗賽)

6. Answer the following questions related to TLB (Translation Lookaside Buffer).

(a) Why TLB can improve system performance?

(b) Why TLB might need to be flushed after context switch?

Ans.

a. 因為查詢 TLB 比查詢 page table 還要來的省時間，一旦在 TLB 查到 page 與 frame 之間的對應，可以節省一次查詢 page 的時間。因此只要 TLB hit ratio 夠好，便能提升系統效能。

b. 因為不同 process 之間的 TLB 是共用的。如果不刷新 TLB，那麼便會錯誤地讀取到其它 process 的資料

[group6] (對抗賽)

7. A virtual memory system has the following characteristics:

- The TLB is a fully associative cache
- The TLB has 32 entries
- The page size is 1024 bytes
- Virtual addresses are 32 bits wide
- Physical addresses are 31 bits wide

Determine the total number of bits in the TLB. Assume each entry of the TLB has the valid bit, the dirty bit, the tag, and the physical page number.

Ans.

The page size is 1024 bytes → page offset: 10 bits

Physical addresses are 31 bits wide → physical page number: $31 - 10 = 21$ (bits)

Virtual addresses are 32 bits wide → tag: $32 - 10 = 22$ (bits)

The TLB has 32 entries → total number of bits in the TLB =

$$32 \times \left(1_{valid} + 1_{dirty} + 22_{tag} + 21_{\text{physical page number}}\right) = 1440$$