20F-COMSCIM151B-1 Homework 6

NEVIN LIANG

TOTAL POINTS

48.25 / 50

QUESTION 1

Dimensions 25 pts

1.1 Offset width 3 / 3

✓ **+ 3 pts Correct**

+ 0 pts Incorrect

1.2 Virtual page number width 3 / 3

✓ **+ 3 pts Correct**

+ 0 pts Incorrect

1.3 Physical page number width 3 / 3

✓ **+ 3 pts Correct**

+ 0 pts Incorrect

1.4 TLB index width 3 / 3

✓ **+ 3 pts Correct**

+ 0 pts Incorrect

1.5 TLB tag width 3 / 3

✓ **+ 3 pts Correct**

+ 0 pts Incorrect

1.6 Number of TLB entries 3 / 3

✓ **+ 3 pts Correct**

+ 1.5 pts Approximate

+ 0 pts Incorrect

1.7 TLB miss rate for C code 7 / 7

✓ **- 0 pts Correct**

- 2 pts Wrong final answer

- 4 pts Confusion between associativity and block size

- 3 pts Assumed decimal prefix

- 7 pts Incorrect

QUESTION 2

Access sequence 25 pts

2.1 Small page 10 / 10

✓ **- 0 pts Correct**

- 3 pts Bad LRU

- 2 pts One wrong tag in TLB

- 5 pts Wrong tags in TLB

- 4 pts Missing column in bottom table

- 4 pts Wrong TLB misses/hits

- 2 pts One wrong TLB miss/hit

- 1 pts Wrong physical page in TLB

- 10 pts Incorrect

2.2 Large page 10 / 10

✓ **- 0 pts Correct**

- 4 pts Page missing in TLB

- 5 pts Wrong TLB tags

- 3 pts Wrong physical page numbers in TLB

- 4 pts Wrong page table hits/misses

- 6 pts Wrong TLB hits/misses

- 3 pts Wrong page table

- 10 pts Incorrect

2.3 Comparison 3.25 / 5

Advantages

✓ **+ 1 pts Fewer TLB/page table misses if decent locality**

✓ **+ 1 pts Smaller page table**

+ 0.5 pts Larger disk bandwidth on page faults

+ 0.5 pts Faster TLB

+ 0.5 pts Smaller TLB

+ 0.5 pts Faster translation in OS

- 1 pts Not an advantage

+ 0 pts Blank

Disadvantages

✓ **+ 1.25 pts** **Waste of memory if many small processes**

+ 1.25 pts Slower page fault resolution

+ 1.25 pts More page faults if many small processes

+ 1 pts Waste of disk bandwidth if small processes

- 1 pts Not a valid disadvantage

QUESTION 3

3 Late Submission Penalty 0 / 0

✓ **- 0 pts** **On time**

- 12.5 pts Late submission

Nevin Liang

Homework Set - 6

Session: Fall 2020 **Instructor: Prof. P. Gupta**

Total Points: 50 **Due Date: November 19**

Instructions

1. **There is only one section in this homework set.**

1. Assume we have a virtual memory system with the following details:
   - 1 GB Physical Address Space
   - 8 GB Virtual Address Space
   - 1 KB page size
   - A TLB with 4 sets that is 8-way associative with LRU replacement.

   (a) How many bits will be used for page offset? [3 points]

   $\log_2 1024 = 10$

10 bits

   (b) How many bits will be used for Virtual Page Number? [3 points]

   $8 \cdot 2^{30} = 2^{33}$ bytes 33 bits - 10 = 23

23 bits

   (c) How many bits will be used for Physical Page Number? [3 points]

   $1 \cdot 2^{30} = 2^{30}$ bytes 30-10=20

20 bits

(d) How many bits will be used for TLB Index? [3 points]

$\log_2 4 = 2$

2 bits

(e) How many bits will be used for TLB tag? [3 points]

23-2 = 21 bits

21 bits

(f) How many entries would be there in the page table? [3 points]

~~8 * 4 = 32~~

2^{23} ~~32~~ entries

(g) We run the following code with an empty TLB. Calculate the TLB miss rate for data (ignore instruction fetches). Assume **i** and **sum** are stored in registers and **cool** is page-aligned. Assume int datatype in C uses 4 bytes of storage. [7 points]

```
#define LEAP 8
int cool[512];
... // Some code that assigns values into the array cool
... // Now flush the TLB. Start counting TLB miss rate from here.
int sum;
for (int i = 0; i < 512; i += LEAP) {
  sum += cool[i];
}
```

What is the TLB miss rate:

.25 KB

$.25 = \frac{1}{4} \cdot 2^{10} = 2^8$

$\frac{64}{256}$

$\frac{64}{256} = \frac{512}{256} = \boxed{2}$ 2/64

1/32

2. Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following table is a stream of virtual addresses as seen on a system. Assume 4 KB pages, a four-entry fully associative TLB, and a true LRU replacement. If pages must be brought in from disk, increment the next largest page number. [20 Points]

4095	31272	15789	15000	7193	4096	8912

TLB

Valid	Tag	Physical Page Number
1	11	12
1	7	4
1	3	6
0	4	9

Assume that the first row of the TLB initially contains the least recently used page number, the second row initially contains the second least recently used page, and the third row initially contains the most recently used page.

Page Table

Valid	Physical Page, or in Disk
1	5
0	Disk
0	Disk
1	6
1	9
1	11
0	Disk
1	4
0	Disk
0	Disk
1	3
1	12

A. Given the address stream in the table and shown the initial state of the TLB and page table, show the final state of the system. Also, list for each reference if it's a hit in the TLB, a hit in the page table, or a page fault. [10 points]

Initial State TLB			Final State TLB		
Valid	Tag	Physical Page Number	Valid	Tag	Physical Page Number
1	11	12	1	1	13
1	7	4	1	7	4
1	3	6	1	3	6
0	4	9	1	~~9~~ 2	~~9~~ 14

Initial State Page Table		Final State Page Table	
Valid	Page or Disk	Valid	Page or Disk
1	5	1	5
0	Disk	1	13
0	Disk	1	14
1	6		
1	9		
1	11		
0	Disk		
1	4		
0	Disk		
0	Disk		
1	3		
1	12		

Address	Binary Address Page Offset (12) Virtual Page Number	TLB	Page table	Page fault
4095	0111111111111		hit	
31272	111101000101000	hit		
15789	11110110101101	hit		
15000	11101010011000	hit		
7193	1110000011001			fault
4096	1000000000000	hit		
8912	10001011010000			fault

B. Repeat this exercise, but this time using 16KB pages instead of 4KB pages.

Initial State TLB			Final State TLB		
Valid	Tag	Physical Page Number	Valid	Tag	Physical Page Number
1	11	12	1	1	13
1	7	4			
1	3	6			
0	4	9	1	0	5

Initial State Page Table		Final State Page Table	
Valid	Page or Disk	Valid	Page or Disk
1	5	1	5
0	Disk	1	13
0	Disk		
1	6		
1	9		
1	11		
0	Disk		
1	4		
0	Disk		
0	Disk		
1	3		
1	12		

Address	Binary Address Page Offset (14) Virtual Page Number	TLB	Page table	Page fault
4095	000111111111111		hit	
31272	111101000101000			fault
15789	011110110101101	hit		
15000	011101010011000	hit		
7193	001110000011001	hit		
4096	001000000000000	hit		
8912	010001011010000	hit		

C. What would be some of the advantages of having a large page size? What are the disadvantages? [2.5 + 2.5 points]

Advantages

- Smaller page table
- fewer faults in the page table
  - Disk instead of page
- the overhead for r/w of pages is less

Disadvantages

- internal fragmentation
- worse locality of reference &.