VM Simulator Design - Team 1

1.Parameter

* Page table: two-level
* Constant:
  + Physical memory size: 16G
  + Address space size: 64 bit
  + // Disk size
  + Page size: 4k
* variables:
  + TLB size: 128 entries

2.Policy

* TLB eviction policy (policy performance depends on workload)
  + LRU
  + LFU
  + Random
  + FIFO
* TLB flushing policy
  + flush all entries
  + Selective flushing (with ASID): which should be flushed and which remain?
  + Lazy Flushing
* Fully associative / Set associative TLB? 全相连/组相连映射
* Swapping
  + LFU
  + FIFO
  + randomFragmentation (optional)
* One level TLB? Two level TLB?
* Unified TLB? Instruction TLB & Data TLB?

3. TLB entry design

* Protection bit or R/W bit:
* Shared?
* Dirty bit
* COW
* Page number len
* Physical frame len
* Present bit, 1 in physical memory, 0 on disk
* Valid bit, 1 valid, 0 not valid
* Use bit for LRU approximation
* Separated huge page entries?

4. Performance measurement interface

Mock CPU cycles:

* Tlb hit 1
* Memory access 20
* Mode switch (kernel & user mode) 100
* Disk access 1000

Other:

* TLB hit rate
* average memory access time (AMAT) versus TLB size for different page sizes

represents the cost of accessing TLB (how long?)

represents the cost of accessing memory (100 nanosec)

represents the cost of accessing disk (10 millisec)

represents TLB miss rate (1 - hit rate)

represent memory miss rate

more?

5. Test case input format

Process\_id action size/pos

1 allocate 100

1 access 50

1 free 100

2 switch c

2 allocate 200

2 access 10

2 access 20

What kind of processes should be designed? E.g. game, microservices, machine learning, database, etc.

**Implementation Documentation**

**Page table**

Class PageTable

| * entries * … |
| --- |
| * getPFN( )   … |

**TLBs**

Class TLB1

| * entries //fully associative * size: 64   … |
| --- |
| * insert() * replace() //policy * flush() //all flushed when process switches   … |

Class TLB2

| * entries //set associative * size: 1024   … |
| --- |
| * insert() * replace() // policy * flush() //selective flushed when process switches: policy   … |

**Bitwise operation**:

<https://baike.baidu.com/item/%E4%BD%8D%E8%BF%90%E7%AE%97/6888804#:~:text=%E4%BD%8D%E8%BF%90%E7%AE%97%E5%B0%B1%E6%98%AF%E7%9B%B4%E6%8E%A5%E5%AF%B9,%E7%A9%BA%E4%BD%8D%E9%83%BD%E5%BD%930%E5%A4%84%E7%90%86%EF%BC%89%E3%80%82> (中文版, 位运算)

<https://www.geeksforgeeks.org/bitwise-operators-in-c-cpp/> (Bitwise operators in C/C++)

**Fully and set associative**:

<https://en.wikipedia.org/wiki/Cache_placement_policies>

Class BusTLB

| * tlb1 * tlb2 * page\_table   … |
| --- |
| This bus might handle  1. moving TLB entries between tlb1 and tlb2  2. moving the entries from page table to tlb1 or tlb2 |

**RAM and Disk**

Class RAM

| * entries * size   … |
| --- |
| * insert()   … |

Class Disk(Swap space)

| * entries * size * … |
| --- |
| * insert()   … |

Class BusSwap

| * ram * disk   … |
| --- |
| This bus might handle the swapping between Ram and Disk   * swap\_out // swapping policy * swap\_in   … |

**Processes, scheduler and generator**

Class Process

| * id * page\_table   … |
| --- |
| * allocate() * access() * free()   … |

Class Scheduler

| * processes   … |
| --- |
| Handle the switching of processes |

Class Generator

| * processes   … |
| --- |
| Model the workloads |