## Virtual Memory

***Summary:*** Virtual Memory is similar to caching, but instead of caching data from a disk, it caches data from RAM. This is done by using a Page Table and a Translation Lookaside Buffer (TLB).

***Definitions:***

- **Page Table** A table that maps virtual page numbers to physical page numbers.
- **Translation Lookaside Buffer (TLB)** A cache for the page table. It stores the most recently used page table entries.
- **Page Fault** When a page table entry is “Disk”, and the page is not in the TLB, a page fault occurs.
- **Page Hit** When a page table entry is not “Disk”, and the page is not in the TLB, a page hit occurs.
- **TLB Hit** When the tag of a page table entry is in the TLB, a TLB hit occurs.

***Example:***
**Given Information:**

- 4KiB page size = 4096 bytes
- Entries
  - 12068
  - 10756
  - 10746
  - 35527
  - 4229
  - 2446
  - 12064
  - 2447
  - 44510

**Initial Page Table:**

| Index | Valid | Physical Page Number |
|---|---|---|
| 0 | 0 | Disk |
| 1 | 0 | Disk |
| 2 | 1 | 5 |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | Disk |
| 7 | 1 | 4 |
| 8 | 0 | Disk |
| 9 | 0 | Disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 1 | 7 |
| 13 | 1 | 8 |
| 14 | 0 | Disk |
| 15 | 0 | Disk |
| 16 | 0 | Disk |
| 17 | 0 | Disk |
| 18 | 1 | 2 |
| 19 | 0 | Disk |
| 20 | 1 | 10 |
| 21 | 1 | 14 |

**Initial TLB:**

| Valid | Tag | Physical Page Number | LRU |
|---|---|---|---|
| 1 | 20 | 10 | 0 |
| 1 | 2 | 5 | 1 |
| 1 | 3 | 6 | 2 |
| 0 | 0 | 15 | 3 |

1. Determine the indices of the page table entries that are accessed for each address by dividing each address by the page size in bytes (with no remainder)

| Address | Page Number |
|---|---|
| 12068 | 2 |
| 10756 | 2 |
| 10746 | 2 |
| 35527 | 8 |
| 4229 | 1 |
| 2446 | 0 |
| 12064 | 2 |
| 2447 | 0 |
| 44510 | 10 |

2. Determine the physical page number for each address by looking up the page table entry in the TLB

   a. If the tag exists in the TLB, then the physical page number is the one in the TLB entry (proceed to step 3) [TLB Hit]

   b. If the tag does not exist in the TLB, then you will have to query the page table to find the physical page number

   - If the page table entry is valid, then the physical page number is the one in the page table entry (proceed to step 3) [Page Hit]

   - If the page table entry is "Disk", then you take the largest physical page number present in the Page Table, multiply it by 2, and then set the physical page number of the page table entry to that value (proceed to step 3) [Page Fault]

3. Update the TLB by inserting the tag and physical page number into the TLB entry that has the highest LRU value, and setting that entry's LRU value to 0. Then increment every other value by 1 (you should only ever have 0,1,2, and 3 in the LRUs)

**Final Page Table:**

| Index | Valid | Physical Page Number |
|---|---|---|
| 0 | 1 | 112 |
| 1 | 1 | 56 |
| 2 | 1 | 5 |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | Disk |
| 7 | 1 | 4 |
| 8 | 0 | 28 |
| 9 | 0 | Disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 1 | 7 |
| 13 | 1 | 8 |
| 14 | 0 | Disk |
| 15 | 0 | Disk |
| 16 | 0 | Disk |
| 17 | 0 | Disk |
| 18 | 1 | 2 |
| 19 | 0 | Disk |
| 20 | 1 | 10 |
| 21 | 1 | 14 |

**Final TLB:**

| Valid | Tag | Physical Page Number | LRU |
|---|---|---|---|
| 1 | 0 | 112 | 1 |
| 1 | 2 | 5 | 2 |
| 1 | 1 | 56 | 3 |
| 1 | 10 | 3 | 0 |

**Final Results:**

| Address | Page Number | Result |
|---|---|---|
| 12068 | 2 | TLB Hit |
| 10756 | 2 | TLB Hit |
| 10746 | 2 | TLB Hit |
| 35527 | 8 | Page Fault |
| 4229 | 1 | Page Fault |
| 2446 | 0 | Page Fault |
| 12064 | 2 | TLB Hit |
| 2447 | 0 | TLB Hit |
| 44510 | 10 | Page Hit |

***Tricky Things to Remember:***

- Page Table Index Starts With 0
- Remember to update your valid bit when you update your tables
- Remember to update your LRU when you update your TLB