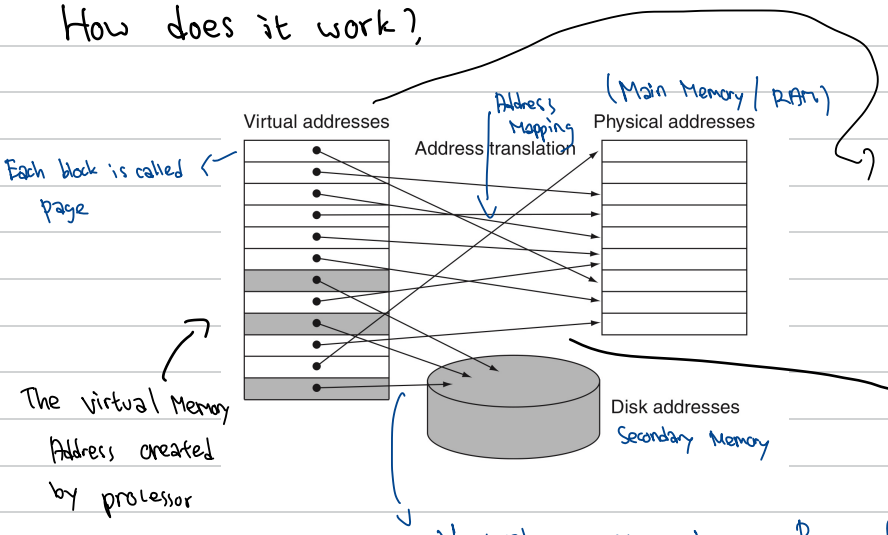
Name: 洪 理川
ID : B113040056

5.7 ~ 5.8

Virtual Memory

Definition : A Memory Management technique that makes main memory (RAM) to have larger memory by using secondary memory.

How does it work?



Each block is called Page

The virtual memory address manages the 2 levels of memory hierarchy, main memory and secondary storage.

The virtual Memory Address created by processor

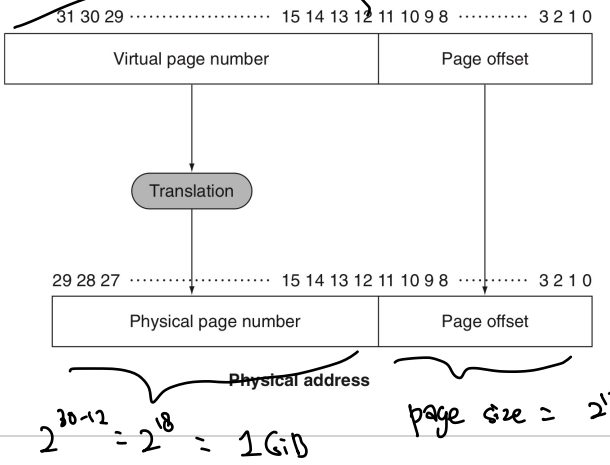
Virtual Memory will provide relocation, so relocation allow to load program anywhere

Virtual memory miss = Page fault

How does Address Mapping Work?

$2^{32-12} = 2^{20} = 4$ GiB

Virtual address = consist of virtual page number & page offset



Usually Virtual pages > Physical Pages for basis in illusion of virtual Memory

$2^{30-12} = 2^{18} = 1$ GiB

page size = $2^{12} = 4$ KiB

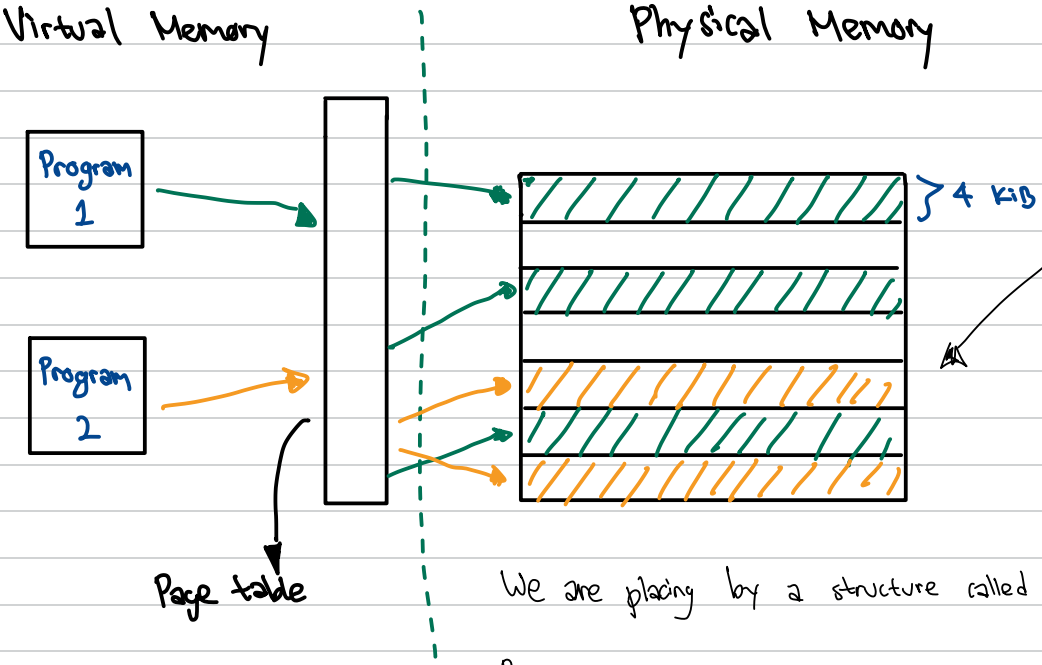
Key desicions in Designing Virtual Memory systems

Background: Page fault → takes million of clock cycles to process

- Pages need to be large
- Pages fault can be handle by software
- Virtual Memory chooses write back instead of write-through

Since disk transfer time is small compared to its access time.

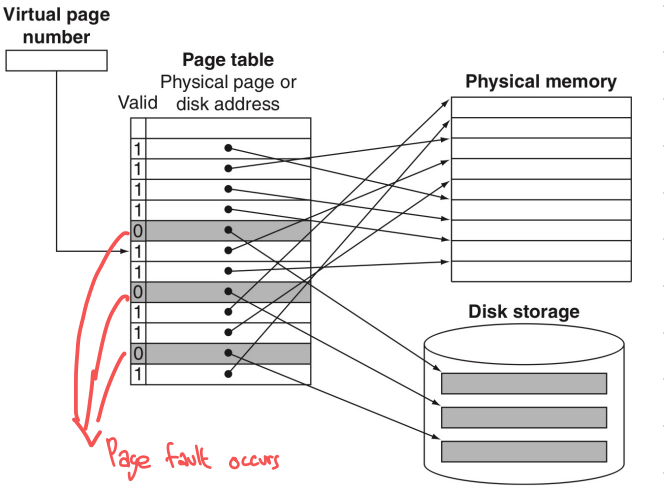
Placing a Page and Finding it Again



We are placing by a structure called page table,

Page tables is indexed by the page number to discover the corresponding page number.

Page faults



Page fault → OS gets the control

↓

Find Available pages

↳ Need keep track of the location in secondary memory of each page in virtual address space.

Using LRU - Least recently used scheme

Also create space "swap space" for page replacement

Besides that, Also record each virtual page is stored on disk.

5 techniques for reducing the amount of storage in page table

A. Keep limit of the register → Address space need in just 1 direction

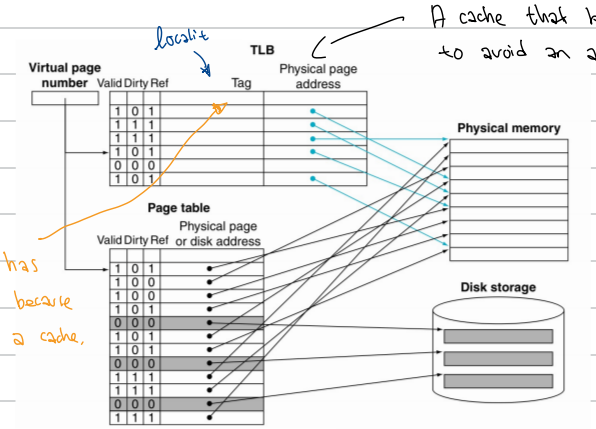
B. Create 2 seperate page tables & 2 seperate limit → can grow from 2 direction

C. Hashing the virtual address, making (page table == physical page) sized

D. Page tables to paged → page tables placed in virtual address space.

E. Multiple level of page tables

TLB: Translation-lookaside buffer



TLB has a tag because it is a cache.

A cache that keeps track of recently used address mappings to try to avoid an access to the page table

- Multi-level page tables works fine.
- Miss in TLB → Page fault
  → TLB miss

How to handle TLB Misses & Page faults

- How can TLB Misses: no entry in the TLB Matches a virtual address

exception mechanism → Transfering Control → Resuming execution

To interrupt the active process

To the Operating System

Need to be declared if the TLB miss / page fault happens.

If it's the Virtual Address caused page fault.

3 steps will be taken:

1. Look up page table → find location of references page.
2. Choose a physical page to replace
3. Read referenced page to the physical page

Common Framework for Memory Hierarchy

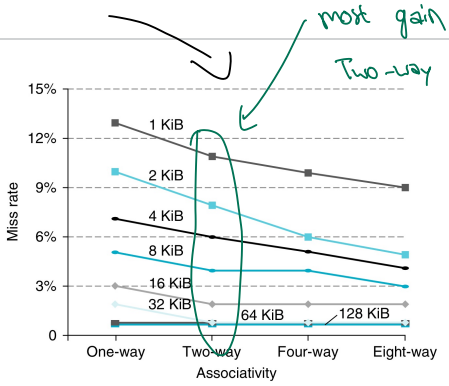
Block placement scheme we learn until now

Scheme name	Number of sets	Blocks per set
Direct mapped	Number of blocks in cache	1
Set associative	Number of blocks in the cache / Associativity	Associativity (typically 2–16)
Fully associative	1	Number of blocks in the cache

chose depend on the MISS vs COST of implementing

Ex: L2 cache → higher associativity (hit times x critical)

↑ degree → ↓ miss rate



Virtual Memory → Full Associativity

- Misses are very expensive
- Can easily index, no extra hardware
- Allow software to use sophisticated schemes

2 Replacement Blocks Techniques

- Random: Randomly choose, use some hardware
- Least Recently Used (LRU) = replace block that hasn't been used for the longer time.
  ↳ used in larger associativity

2 techniques on Write

Write-through

used in cache

Definition: Information written to cache & lower level memory hierarchy (RAM for a cache)

Advantages:
- Easier to implement
- Miss are cheaper

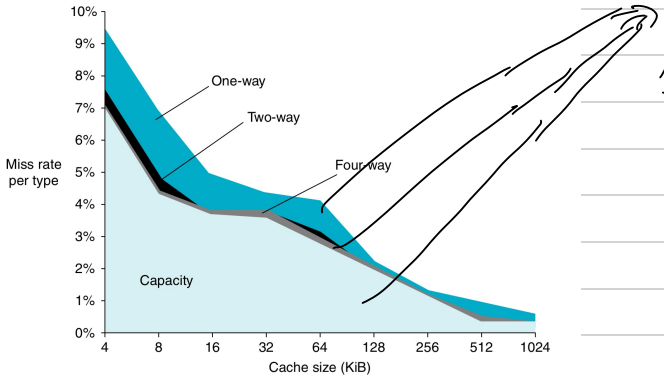
Write-back

used in Virtual memory

Definition: The information is written to cached only.
The information written to lower level memory hierarchy only when it's replaced

Advantages:
- Effective use of high-bandwidth transfer
- Multiple writes → 1 writes
- Individual words can be written by the processor

Miss Rate Sources



3 Sources of Miss Rate

1. Compulsory misses (cold-start misses):
   cause: first access to a block that isn't in the cache.
2. Capacity misses
   cause: cache can't contain all the blocks
3. Conflict misses
   cause: multiple blocks compete for the same set

Design Consideration

Design change	Effect on miss rate	Possible negative performance effect
Increases cache size	Decreases capacity misses	May increase access time
Increases associativity	Decreases miss rate due to conflict misses	May increase access time
Increases block size	Decreases miss rate for a wide range of block sizes due to spatial locality	Increases miss penalty. Very large block could increase miss rate

Block placement scheme