A Real Problem

- What if you wanted to run a program that needs more memory than you have?

HARD DRIVE TO STORE MEMORY YOU AREN'T (CURRENTLY) USING.

VIRTUAL MEMORY

Virtual Memory (and Indirection)

- Finally, we get to Virtual Memory!
  - We' ll talk about the motivations for virtual memory
  - We' ll talk about how it is implemented
  - Lastly, we' ll talk about how to make virtual memory fast: Translation Lookaside Buffers (TLBs).

A Real Problem

- What if you wanted to run a program that needs more memory than you have?
  - You could store the whole program on disk, and use memory as a cache for the data on disk. This is one feature of virtual memory.
  - Before virtual memory, programmers had to manually manage loading "overlays" (chunks of instructions & data) off disk before they were used. This is an incredibly tedious, not to mention error-prone, process.

More Real Problems

- Running multiple programs at the same time brings up more problems.

1. Even if each program fits in memory, running 10 programs might not.

2. Multiple programs may want to store something at the same address.

3. How do we protect one program's data from being read or written by another program?

More Real Problems

- Running multiple programs at the same time brings up more problems.

1. Even if each program fits in memory, running 10 programs might not.
   - This is really the same problem as on the previous slide.

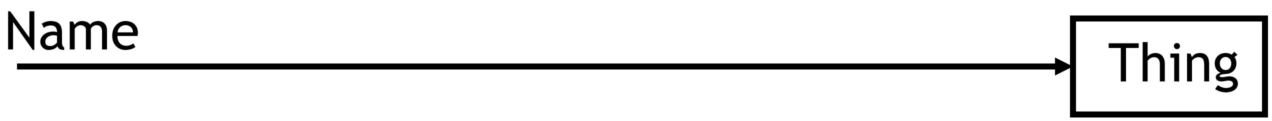
2. Multiple programs may want to store something at the same address.
   - I.e., what if both Program A and B want to use address 0x10000000 as the base of their stack?
   - It is impractical (if not impossible) to compile every pair of programs that could get executed together to use distinct sets of addresses.

3. How do we protect one program's data from being read or written by another program?

Indirection

- "Any problem in CS can be solved by adding a level of indirection"

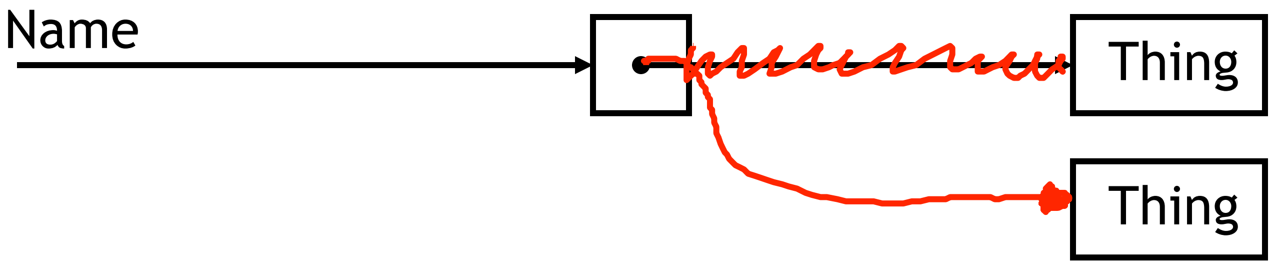
- Without Indirection



Name

Thing

- With Indirection



Name

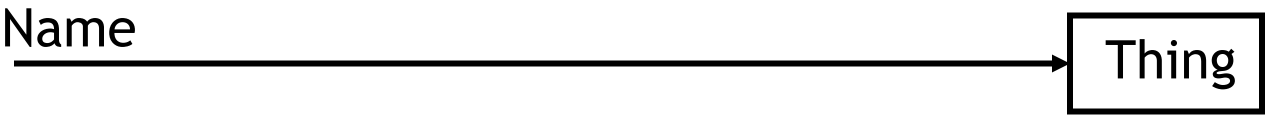
Thing

Thing

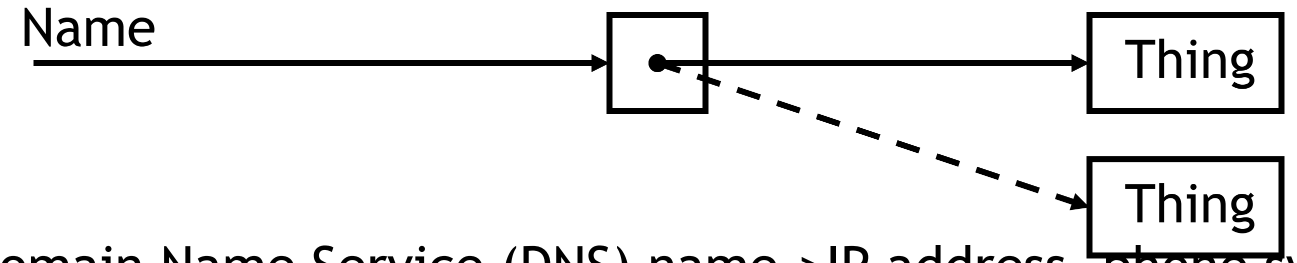
Indirection

- **Indirection:** Indirection is the ability to reference something using a name, reference, or container instead the value itself. A flexible mapping between a name and a thing allows changing the thing without notifying holders of the name.

- Without Indirection



- With Indirection

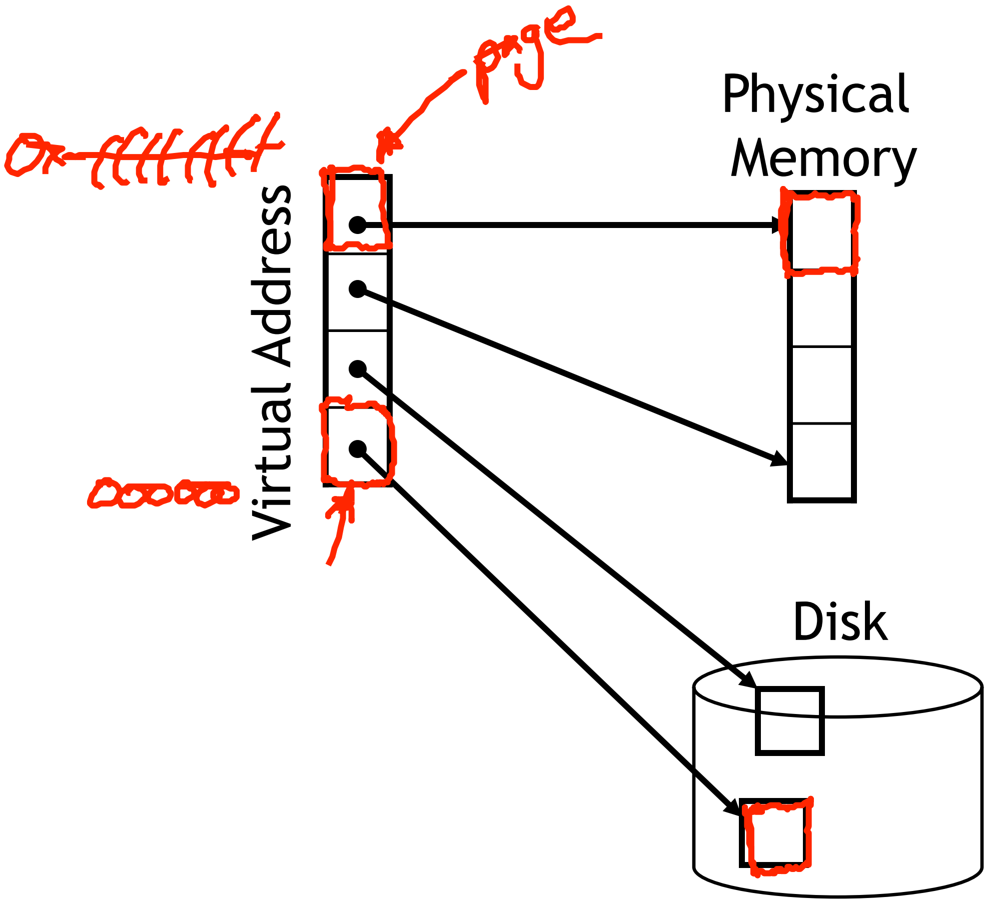


- **Examples:**

  Pointers, Domain Name Service (DNS) name->IP address, phone system (e.g., cell phone number portability), snail mail (e.g., mail forwarding), 911 (routed to local office), DHCP, color maps, call centers that route calls to available operators, etc.

Virtual Memory

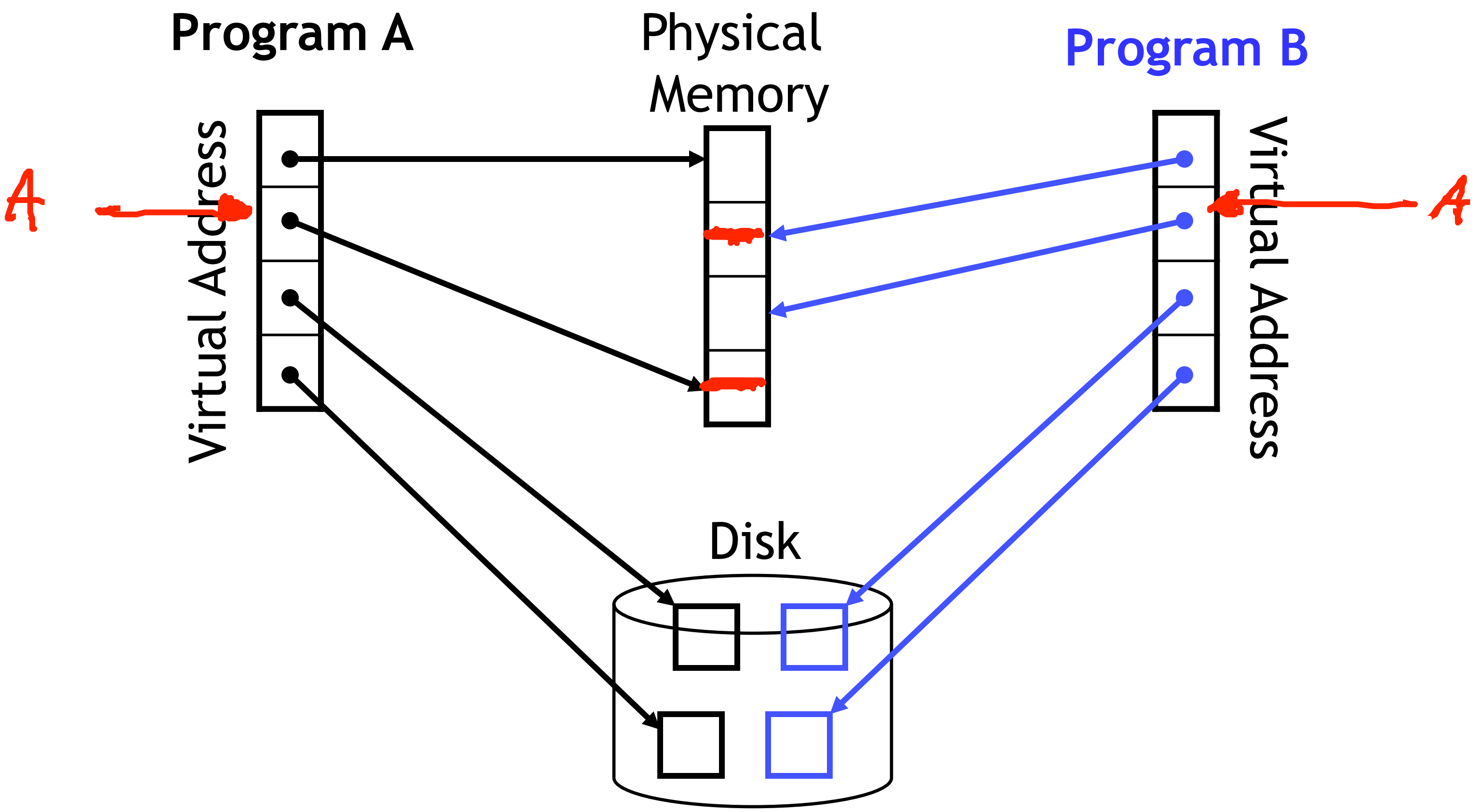
- We translate “virtual addresses” used by the program to “physical addresses” that represent places in the machine’s “physical” memory.
  - The word “translate” denotes a level of indirection



A virtual address can be mapped to either physical memory or disk.

Virtual Memory

- Because different processes will have different mappings from virtual to physical addresses, two programs can freely use the same virtual address.
- By allocating distinct regions of physical memory to A and B, they are prevented from reading/writing each others data.



Caching revisited

- Once the translation infrastructure is in place, the problem boils down to caching.
  - We want the size of disk, but the performance of memory.

- The design of virtual memory systems is really motivated by the high cost of accessing disk.
  - While memory latency is ~100 times that of cache, disk latency is ~100,000 times that of memory.
    - i.e., the miss penalty is a real whopper.

GHz 10^9

32B, 64B 10ns 10^2

- Hence, we try to minimize the miss rate:
  - VM "pages" are much larger than cache blocks. Why?
  - A fully associative policy is used.
    - With approximate LRU

pages 4kB, 8kB

Spatial locality

no conflicts

Just capacity

24

- Should a write-through or write-back policy be used?

D E