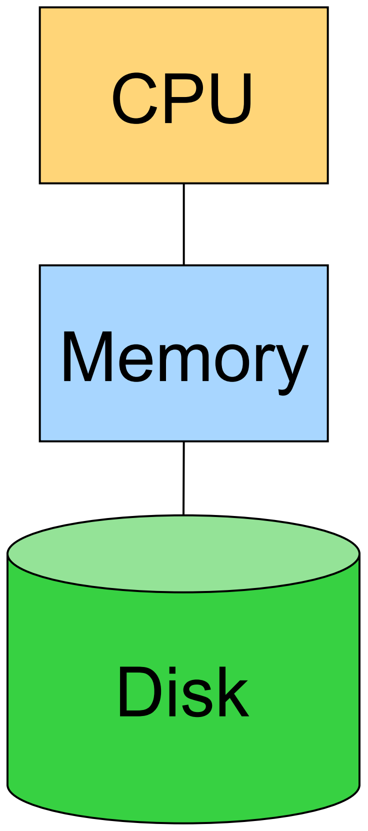
COS 318: Operating Systems

Virtual Memory and Address Translation

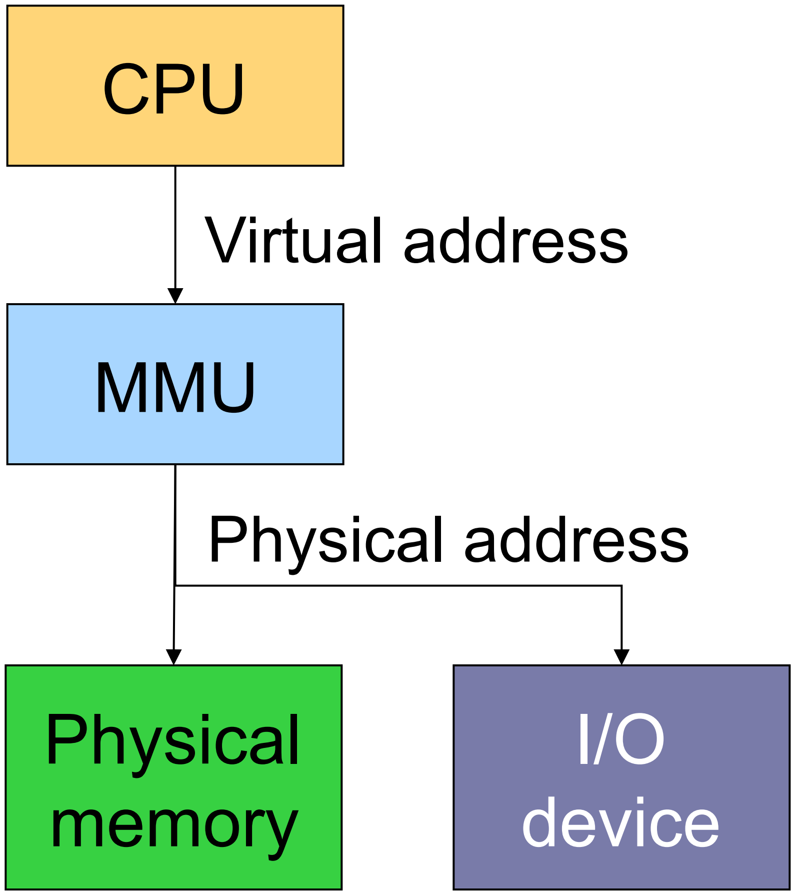
The Big Picture

- DRAM is fast, but relatively expensive
  - $25/GB
  - 20-30ns latency
  - 10-80GB's/sec
- Disk is inexpensive, but slow
  - $0.2-1/GB (100 less expensive)
  - 5-10ms latency (200K-400K times slower)
  - 40-80MB/sec per disk (1,000 times less)
- Our goals
  - Run programs as efficiently as possible
  - Make the system as safe as possible



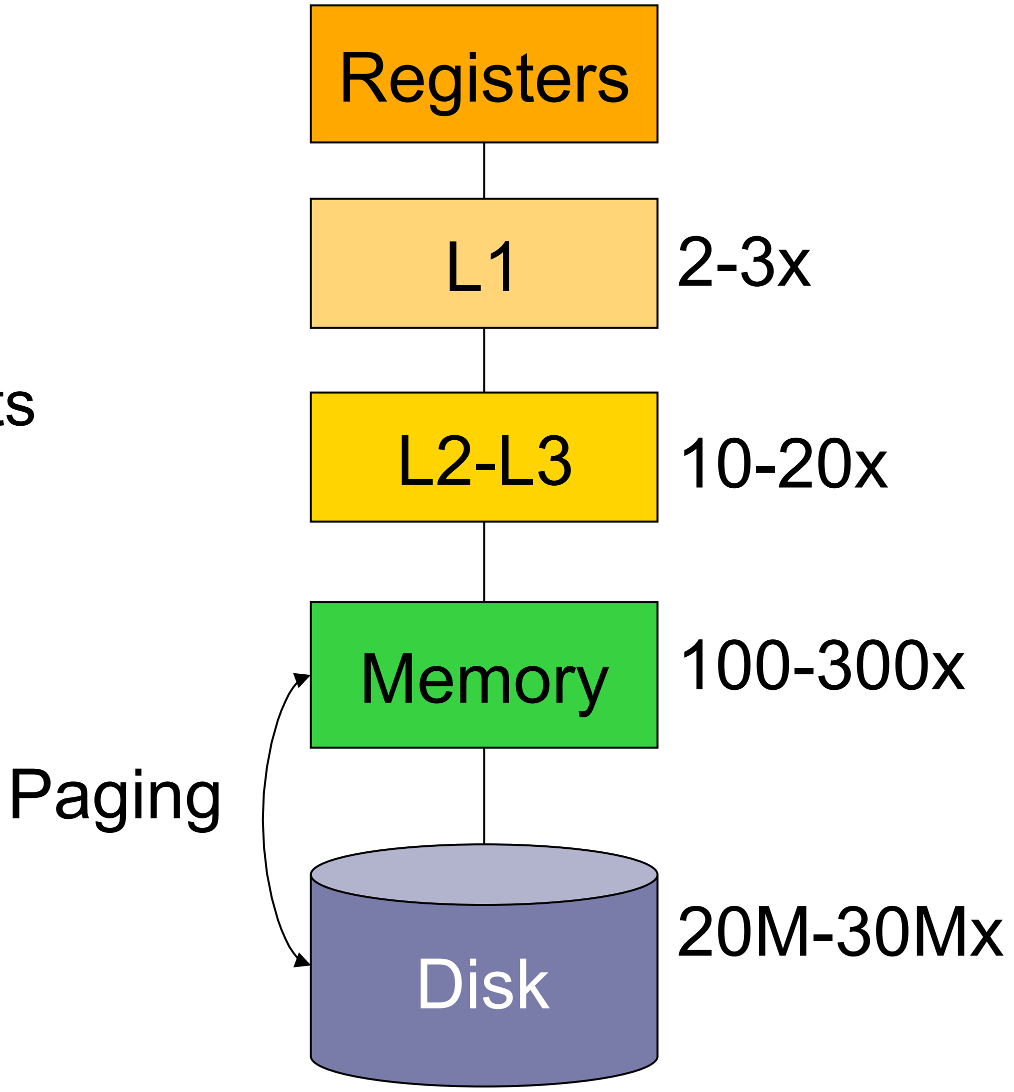
Generic Address Translation

- Memory Management Unit (MMU) translates virtual address into physical address for each load and store
- Software (privileged) controls the translation
- CPU view
  - Virtual addresses
- Each process has its own memory space [0, high]
  - Address space
- Memory or I/O device view
  - Physical addresses



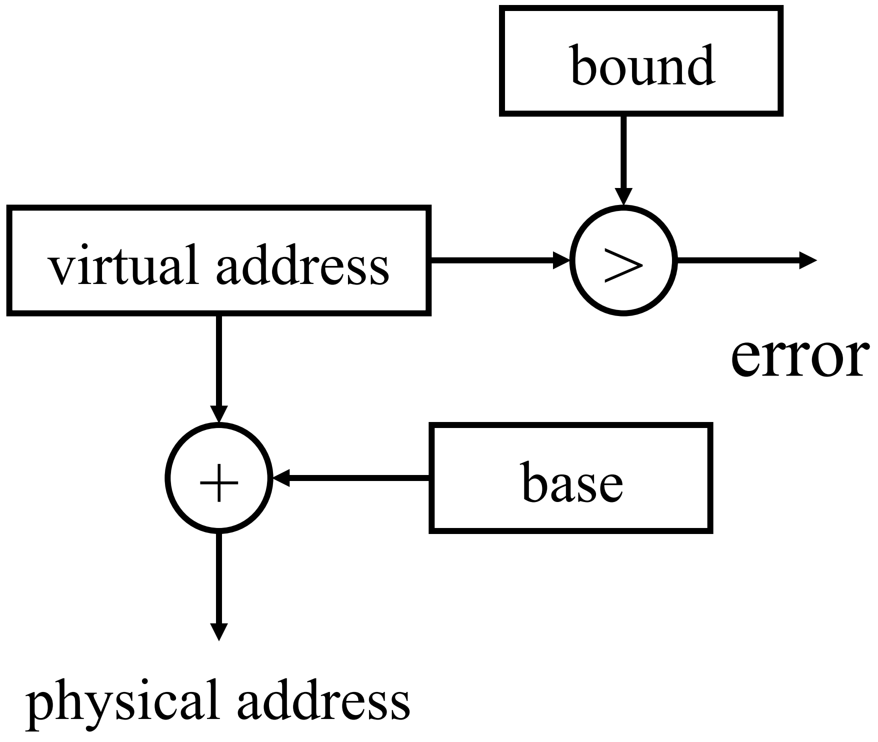
Goals of Translation

- Implicit translation for each memory reference
- A hit should be very fast
- Trigger an exception on a miss
- Protected from user's faults



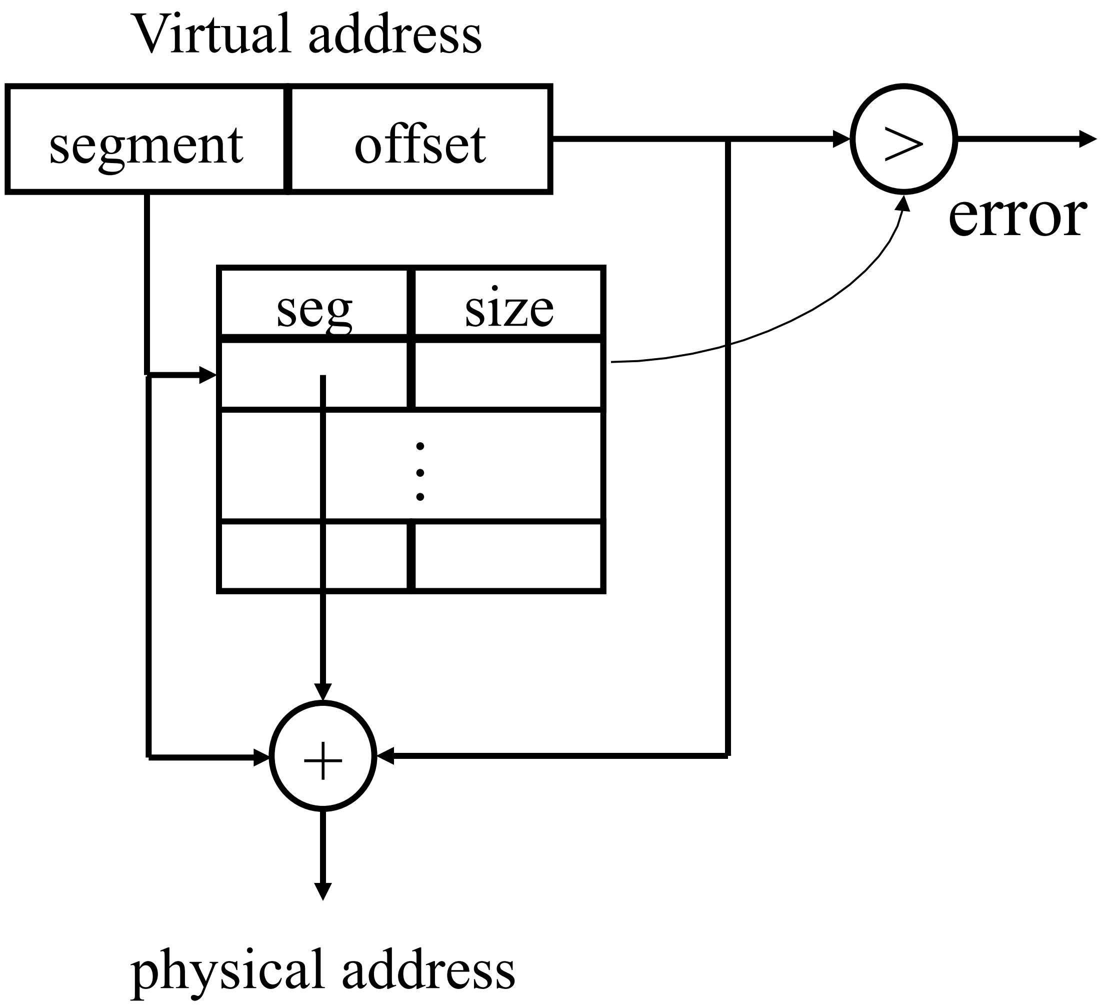
Base and Bound

- Built in Cray-1
- Each process has a pair (base, bound)
- Protection
  - A process can only access physical memory in [base, base+bound]
- On a context switch
  - Save/restore base, bound registers
- Pros
  - Simple
  - Flat and no paging
- Cons
  - Fragmentation
  - Hard to share
  - Difficult to use disks



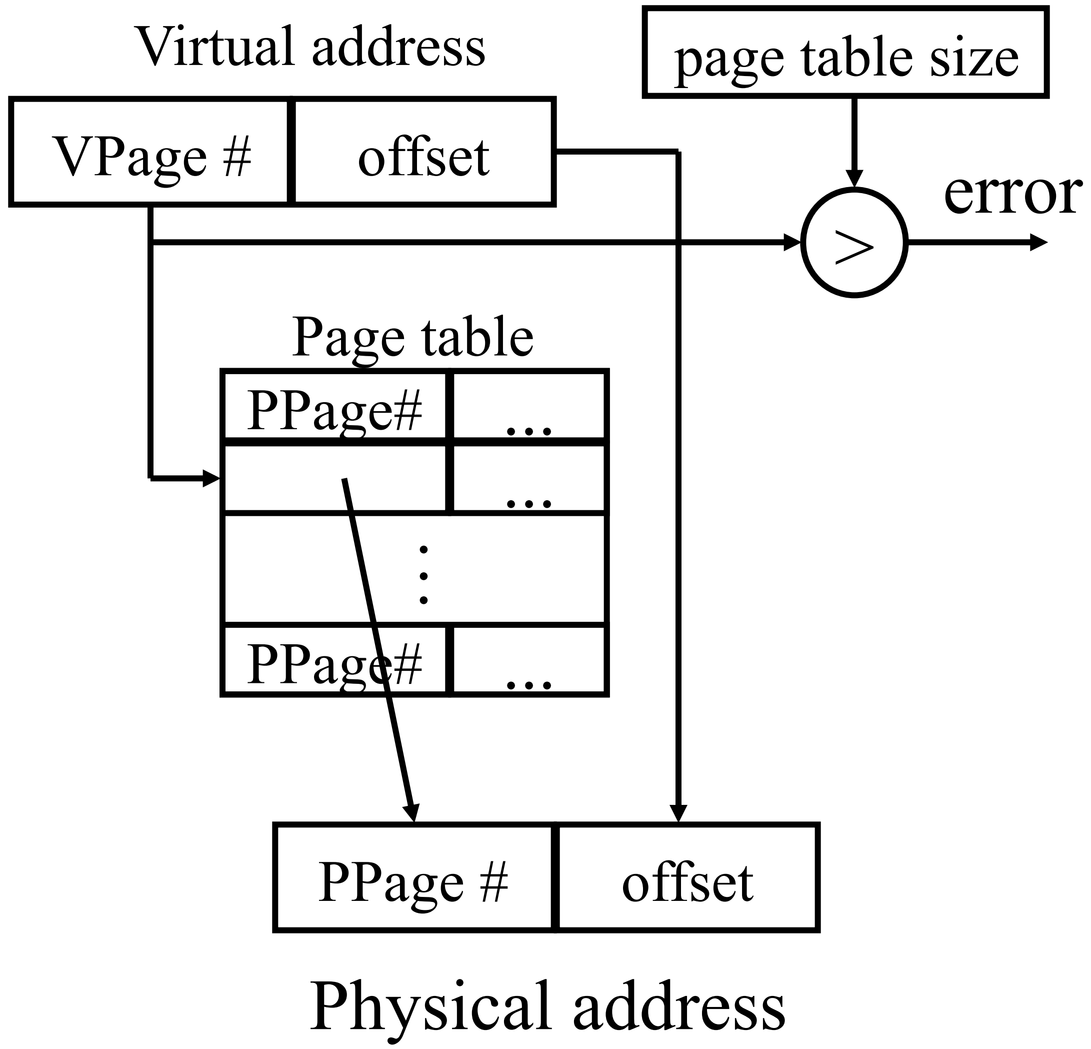
Segmentation

- Each process has a table of (seg, size)
- Treats (seg, size) has a fine-grained (base, bound)
- Protection
  - Each entry has (nil, read, write, exec)
- On a context switch
  - Save/restore the table and a pointer to the table in kernel memory
- Pros
  - Efficient
  - Easy to share
- Cons
  - Complex management
  - Fragmentation within a segment



Paging

- Use a fixed size unit called page instead of segment
- Use a page table to translate
- Various bits in each entry
- Context switch
  - Similar to segmentation
- What should page size be?
- Pros
  - Simple allocation
  - Easy to share
- Cons
  - Big table
  - How to deal with holes?


Virtual address
VPage #
offset
page table size
>
error
Page table
PPage#
...
...
PPage#
...
PPage #
offset
Physical address