

## Lecture 5: Virtual Memory Management

2227, Spring 2006

1

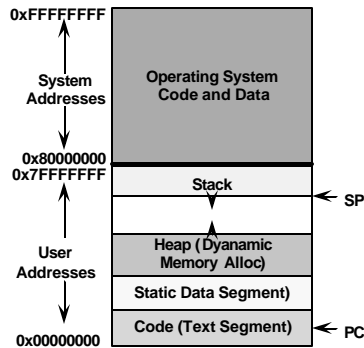
## Memory Management Requirements

- Relocation
  - Programmers don't know what physical memory will be available when their programs run
  - → need some type of address translation
- Protection
  - A process's memory should be protected from *unwanted* access by other processes, both intentional and accidental
  - → Requires hardware support
- Sharing
  - Need ways to specify and control what sharing is allowed
- Logical/Physical Organization
  - Map between program structures and linear array of bytes
  - Manage transfers between disk and main memory

2227, Spring 2006

2

## Virtual address space



- process address space (A.S.) layout
  - logical or *virtual* A.S.
- CPU generates logical addresses in this space as program executes
  - Called *virtual addresses*
- Memory system must see physical (real) addresses
  - Translation is done by *memory management unit* (MMU)
  - Physical memory must be allocated for each virtual location used by the program

2227, Spring 2006

3

## Paging

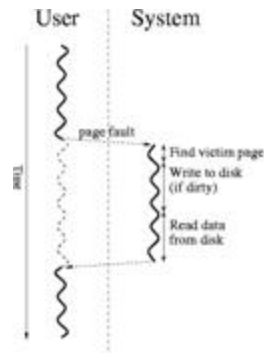
- Partition memory into equal, fixed-size chunks
  - called *page frames* or simply *frames*
- Divide processes' memory into chunks of the same size
  - These are called *pages*
- Any page can be assigned to any free page frame
  - No external fragmentation
  - Minimal internal fragmentation
- First seen in CTSS circa 1961
- *Demand paging* (automatic transfer to/from backing store) first used in the Atlas computer
  - Described in a 2-page CACM article, 1961

2227, Spring 2006

4

## Atlas virtual memory

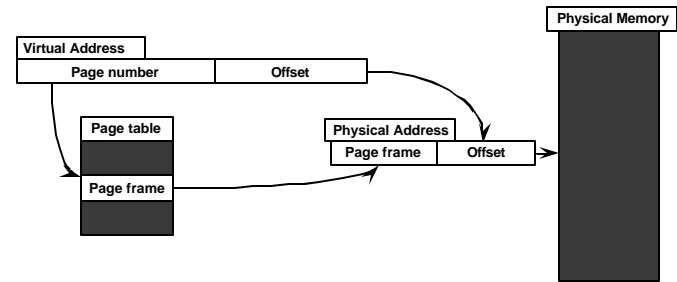
- Inverted page table (entry per physical page, records what virtual page is stored there)
  - Only 2048 entries, stored in registers, searched in parallel
- Missing pages fetched on demand from drum into core
  - Victim also selected on demand



2227, Spring 2006

5

## “Typical” Address Translation

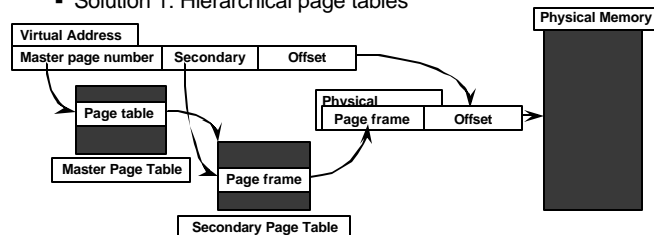


2227, Spring 2006

6

## Page tables – space limitations

- Memory required for page table can be large
  - Need one PTE per page
  - 32 bit virtual address space w/ 4K pages =  $2^{20}$  PTEs
  - 4 bytes/PTE = 4MB/page table
  - 25 processes = 100MB just for page tables!
  - Solution 1: Hierarchical page tables



2227, Spring 2006

7

## 64-bit address spaces

- Suppose we just extended the hierarchical page tables with more levels
  - 4K pages → 52 bits for page numbers
  - Maximum 1024 entries per level → 6 levels
    - Too much overhead
  - 16K pages → 48 bits for page numbers
  - Maximum 4096 entries per level → 4 levels
    - Better, but still a lot
- Paper 1: “A new page table for 64-bit address spaces”, Talluri, Hill & Khalidi, SOSP '95
  - Introduces *clustered page tables*, building on the concept of hashed page tables

2227, Spring 2006

8

## Page tables – time overhead

- Each virtual memory reference requires multiple physical memory references to complete
  - 1 per level in hierarchical tables + actual data access
- Solution: cache recently used translations in MMU
  - Translation lookaside buffer (TLB)
  - Fully associative cache (all entries looked up in parallel)
  - Indexed by virtual page numbers
  - entries are PTEs (entries from page tables)
  - With PTE + offset, can directly calculate physical address

2227, Spring 2006

9

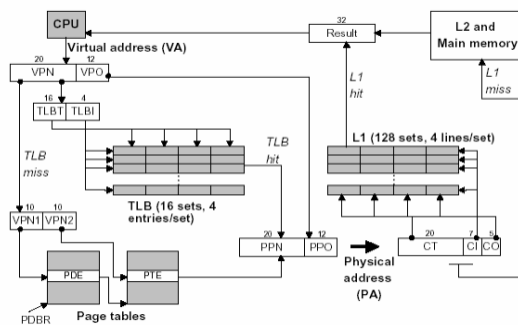
## TLB performance

- TLB hit rates critical to performance
  - *TLB reach* == fraction of the virtual address space covered by the TLB
  - Depends on page size, number of TLB entries
- TLB size is fixed (typically small, 2048 entries or less)
- Many modern MMU's allow multiple page sizes
- Paper 2: “Practical, transparent operating system support for superpages”, Navarro, Iyer, Druschel & Cox, OSDI'02

2227, Spring 2006

10

## Pentium Address Translation



2227, Spring 2006

11

## Other big issues in memory management...

- Placement policy
  - Given multiple free page frames, which one should be allocated?
- Replacement policy
  - Tons of work in the 60's and 70's and since...
- Fetch policy
  - On-demand or prefetch (which ones? How many? Timing?)
- Balancing real memory used for virtual memory vs. file cache

2227, Spring 2006

12