CIS 429/529

Virtual Memory

Prof. Michel A. Kinsky
Address Translation

Virtual Address

TLB Lookup

Page Table Walk

Update TLB

Protection Fault

Protection Check

Physical Address (to cache)

Page Fault (OS loads page)

Segment Fault

Hardware

Hardware or Software

Software

Miss

Hit

The page is

is not in memory

The page is

is in memory

Denied

Permitted

Where?
Page Fault Handler

• When the referenced page is not in DRAM:
  ‣ The missing page is located (or created)
  ‣ It is brought in from disk, and page table is updated
    • Another job may be run on the CPU while the first job waits for the requested page to be read from disk
  ‣ If no free pages are left, a page is swapped out
    • approximate LRU replacement policy
Page Fault Handler

• Since it takes a long time to transfer a page (msecs), page faults are handled completely in software by the OS

  ‣ Untranslated addressing mode is essential to allow kernel to access page tables
Translation for Page Tables

- Can references to page tables cause TLB misses?

- User VA translation causes a TLB miss

- Page table walk: User PTE Base and appropriate bits from VA are used to obtain virtual address VP for page table entry

- Get a TLB miss when we try to translate VP
Translation for Page Tables

- When we get a TLB miss on VP translation, OS adds System PTE Base to bits from VP to find physical address of page table entry for VP
Swapping a Page of a Page Table

- A PTE in primary memory contains primary or secondary memory addresses
- A PTE in secondary memory contains only secondary memory addresses
- A page of a PT can be swapped out only if none of its PTE’s point to pages in the primary memory
Address Translation in CPU Pipeline

- Software handlers need a restartable exception on page fault or protection violation.
- Handling a TLB miss needs a hardware or software mechanism to refill TLB.
Address Translation in CPU Pipeline

- Need mechanisms to cope with the additional latency of a TLB:
  - slow down the clock
  - pipeline the TLB and cache access
  - virtual address caches
  - parallel TLB/cache access
Physical or Virtual Address Caches?

- one-step process in case of a hit (+)
- cache needs to be flushed on a context switch unless address space identifiers (ASIDs) included in tags (-)
- aliasing problems due to the sharing of pages (-)
Aliasing in Virtual-Address Caches

Two virtual pages share one physical page

- General Solution: Disallow aliases to coexist in cache

Virtual cache can have two copies of same physical data. Writes to one copy not visible to reads of other!
Aliasing in Virtual-Address Caches

Software (i.e., OS) solution for direct-mapped cache

1. VAs of shared pages must agree in cache index bits; this ensures all VAs accessing same PA will conflict in direct-mapped cache (early SPARCs)

Diagram:
- Page Table
- Data Pages
- PA
- Tag
- Data

Virtual cache can have two copies of same physical data. Writes to one copy not visible to reads of other!
Concurrent Access to TLB & Cache

- Index L is available without consulting the TLB
  - cache and TLB accesses can begin simultaneously
Concurrent Access to TLB & Cache

- Tag comparison is made after both accesses are completed
  - Cases: \( L + b = k \), \( L + b < k \), \( L + b > k \) what happens here?
Virtual-Index Physical-Tag Caches

- After the PPN is known, $W$ physical tags are compared
- Allows cache size to be greater than $2^{L+b}$ bytes
Virtual Memory Use Today

- Desktops/servers have full demand-paged virtual memory
  - Portability between machines with different memory sizes
  - Protection between multiple users or multiple tasks
  - Share small physical memory among active tasks
  - Simplifies implementation of some OS features
Virtual Memory Use Today

- Vector supercomputers have translation and protection but not demand-paging (Older Crays: base&bound, Japanese & Cray X1: pages)
  - Don’t waste expensive CPU time thrashing to disk (make jobs fit in memory)
  - Mostly run in batch mode (run set of jobs that fits in memory)
  - Difficult to implement restartable vector instructions
Virtual Memory Use Today

• Most embedded processors and DSPs provide physical addressing only
  ▸ Can’t afford area/speed/power budget for virtual memory support
  ▸ Often there is no secondary storage to swap to!
  ▸ Programs custom written for particular memory configuration in product
  ▸ Difficult to implement restartable instructions for exposed architectures