ADDRESS TRANSLATION AND TLB

Mahdi Nazm Bojnordi
Assistant Professor
School of Computing
University of Utah
Overview

- Announcement
  - Homework 4 submission deadline: Nov. 6th

- This lecture
  - Virtual memory
  - Page tables and address translation
  - Translation look-aside buffer (TLB)
Recall: Memory Hierarchy

- Lower levels provide greater capacity longer time
  - Does the program fit in main memory?
  - What if running multiple programs?

Greater Capacity

- Cache: Capacity: 8MB, Time: ~20 ns
- Main Memory: Capacity: 8GB, Time: ~250 ns
- Secondary Memory: Capacity: 500GB, Time: ~10 ms
Virtual Memory

- Use the main memory as a "cache" for secondary memory
  - Placement policy?

```c
for(i=0; i<100;++i) {
    a[i]++;
}
```
Virtual Memory

- Use the main memory as a “cache” for secondary memory
  - Placement policy?
- Allow efficient and safe sharing the physical main memory among multiple programs
  - Replacement policy?

```c
for(i=0; i<100; ++i) {
a[i]++; 
}
for(i=0; i<200; ++i) {
a[i]=a[i]+i;
}
```
Virtual Memory Systems

- Provides illusion of very large memory
  - Address space of each program larger than the physical main memory
- Memory management unit (MMU)
  - Between main and secondary mem.
  - Address translation
    - Virtual address space used by the program
    - Physical address space is provided by the physical main memory

Secondary Memory
Virtual Address Space
Virtual Memory Systems

- Provides illusion of very large memory
  - Address space of each program larger than the physical main memory
- Memory management unit (MMU)
  - Between main and secondary mem.
  - Address translation
    - Virtual address space used by the program
    - Physical address space is provided by the physical main memory
Virtual Address

- Every virtual address is translated to a physical address with the help of hardware
- Data granularity
Virtual Address

- Every virtual address is translated to a physical address with the help of hardware.
- Data granularity

<table>
<thead>
<tr>
<th>31</th>
<th>Virtual Address</th>
<th>0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Physical Address</th>
<th>0</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Physical Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Frame 0</td>
</tr>
<tr>
<td>Page Frame 1</td>
</tr>
<tr>
<td>Page Frame 2</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Page Frame N-1</td>
</tr>
</tbody>
</table>
Virtual Address

- Every virtual address is translated to a physical address with the help of hardware
- Data granularity
Virtual Address

- Every virtual address is translated to a physical address with the help of hardware
- Data granularity
Virtual Address

- Every virtual address is translated to a physical address with the help of hardware
- Data granularity

How big is the page table?
Address Translation Issues

- Where to store the table?
  - Too big for on-chip cache
  - Should be maintained in the main memory
Address Translation Issues

- Where to store the table?
  - Too big for on-chip cache
  - Should be maintained in the main memory

- What to do on a page table miss (page fault)?
  - No valid frame assigned to the virtual page
  - OS copies the page from disk to page frame
Address Translation Issues

- Where to store the table?
  - Too big for on-chip cache
  - Should be maintained in the main memory

- What to do on a page table miss (page fault)?
  - No valid frame assigned to the virtual page
  - OS copies the page from disk to page frame

- What is the cost of address translation?
  - Additional accesses to main memory per every access
  - Optimizations?
Address Translation Cost

- Page walk: look up the physical address in the page table

How many pages for storing the page table?

```
Virtual Address

20  12

base→

Page Table

Page frame No  12

Physical Address
```
Multi-Level Page Table

- The virtual (logical) address space is broken down into multiple pages
  - Example: 4KB pages
Translation Lookaside Buffer

- Exploit locality to reduce address translation time
  - Keep the translation in a buffer for future references
Translation Lookaside Buffer

- Exploit locality to reduce address translation time
  - Keep the translation in a buffer for future references
Translation Lookaside Buffer

- Just like any other cache, the TLB can be organized as fully associative, set associative, or direct mapped.
- TLB access is typically faster than cache access:
  - Because TLBs are much smaller than caches.
  - TLBs are typically not more than 128 to 256 entries even on high-end machines.

<table>
<thead>
<tr>
<th>V</th>
<th>Virtual Page #</th>
<th>Physical Page #</th>
<th>Dirty</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Content addressable memory (CAM)
- Unlike RAM, data in address out

RAM: Read Operation
- Address in
- Decoder
- Data in
- Data out

CAM: Search Operation
- Data in
- address
- match

What if multiple rows match?
CAM Based TLB

- Content addressable memory (CAM)
  - Unlike RAM, data in address out

- CAM based TLB
  - Both CAM and RAM are used