Memory Paging

CS 256/456
Dept. of Computer Science, University of Rochester

Paging: Address Translation Scheme

A logical address is divided into:
- Page number (p) – used as an index into a page table which contains base address of each page in physical memory.
- Page offset (d) – the offset address with each page/frame.

Use TLB to speed up the address translation.

Paging MMU With TLB

Effective Access Time

- Assume
  - TLB Lookup = 1 ns
  - Memory cycle time is 100 ns
- Hit ratio (α) – percentage of times that a page number is found in the TLB.
- Effective memory Access Time (EAT)
  \[ EAT = 101\times\alpha + 201\times(1 - \alpha) \]
Layout of A Page Table Entry

- Physical page frame address
- No logical page number
- Other bits for various page properties

Memory Access Setting in Page Table

- Parts of the logical address space may not be mapped
  - Valid-invalid bit attached to each entry in the page table.
  - indicating whether the associated page is in the process’ logical address space, and is thus a legal page.
- Some pages are read-only, or can’t contain executable code
  - access bits in page table to reflect these.
- Software exception if attempting to access an invalid page, or to perform disallowed actions

Process Creation: Copy-on-Write

- Basic idea:
  - fork() semantics says the child process has duplicate copy of the parent’s address space
  - child process often calls exec() right after fork()
  - Copy-on-Write (COW) allows both parent and child processes to initially share the same pages in memory.
- Implementation:
  - shared pages are marked readonly after fork().
  - if either process modifies a shared page, a page fault occurs and then the page is copied.
  - the other process (who later faults on write) discovers it is the only owner; so it doesn’t copy again.

Page Table Structure

- Problem with a flat linear page table
  - assume a page table entry is 4-byte; page size is 4KB; the 32-bit address space is 4GB large
  - how big is the flat linear page table?
- Solutions:
  - Hierarchical Page Tables
    - break the logical page number into multiple levels
- Metrics:
  - Space consumption and lookup speed
Two-Level Page Table

- A logical address (on 32-bit machine with 4K page size) is divided into:
  - a page offset consisting of 12 bits.
  - a page number consisting of 20 bits; further divided into:
    - a 10-bit level-2 page number.
    - a 10-bit level-1 page number.
- Thus, a logical address looks like:

<table>
<thead>
<tr>
<th>Level</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page number</td>
<td>20</td>
</tr>
<tr>
<td>Level-1 page number</td>
<td>10</td>
</tr>
<tr>
<td>Level-2 page number</td>
<td>10</td>
</tr>
<tr>
<td>Page offset</td>
<td>12</td>
</tr>
</tbody>
</table>

Address translation scheme:

<table>
<thead>
<tr>
<th>Logical address</th>
<th>Page number</th>
<th>Page offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>( p_2 )</td>
<td>( d )</td>
</tr>
</tbody>
</table>

Two-Level Page Table: An Example

- Space consumption
- Lookup speed

Deal With 64-bit Address Space

- Two-level page tables for 64-bit address space
  - more levels are needed
- Inverted page tables
  - One entry for each real page of memory.
  - Entry consists of the process id and virtual address of the page stored in that real memory location.

- Problems:
  - search takes too long
  - difficult to share memory

Hashed Page Tables

- The virtual page number is hashed into a page table. This page table contains a chain of elements hashing to the same location.
- Virtual page numbers are compared in this chain searching for a match. If a match is found, the corresponding physical frame is extracted.
Page Size Selection

- Issues concerning page size
  - fragmentation
  - page table size
  - TLB reach

- **TLB Reach** - the amount of memory accessible from the TLB.
  - TLB Reach = (TLB Size) X (Page Size)
- Large TLB reach means fewer TLB misses

- Multiple page sizes:
  - This allows applications that require larger page sizes the opportunity to use them without an increase in fragmentation.