内核笔记

段武杰

2016-09-09

目录

内存寻址

1.1 没有高速缓存的转换过程

在没有高速缓存的情况下虚拟地址的转换过程：

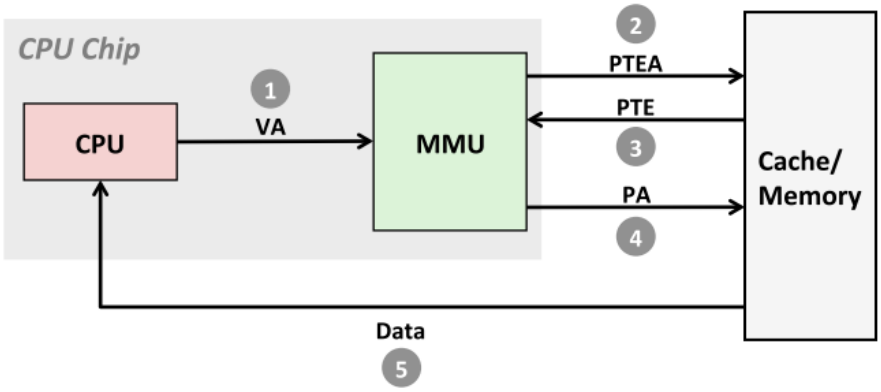

Figure 1.1: 没有高速缓存对数据的访问

1. Processor sends virtual address to MMU
2. MMU fetches PTE from page table in memory(Access memory once)
3. MMU sends physical address to memory(Access memory twice)
4. Memory send sdata word to processor

因此在没有高速缓存的情况下，CPU 读取数据会进行 2 次内存访问，其中 1 次读取页表，1 次读取数据。

1.2 有高速缓存的转换过程

在有高速缓存的情况下虚拟地址的转换过程：

1. 处理器将虚拟地址发送给 MMU
2. MMU 根据 PTEA 从 cache 中读取 pte，如果命中 cache 则使用转换出的物理地址读取内存数据。
3. 如果没有命中 cache，则重内存中取得 pte 并更新 cache，再重内存中读取数据 (最多读取 2 次内存)

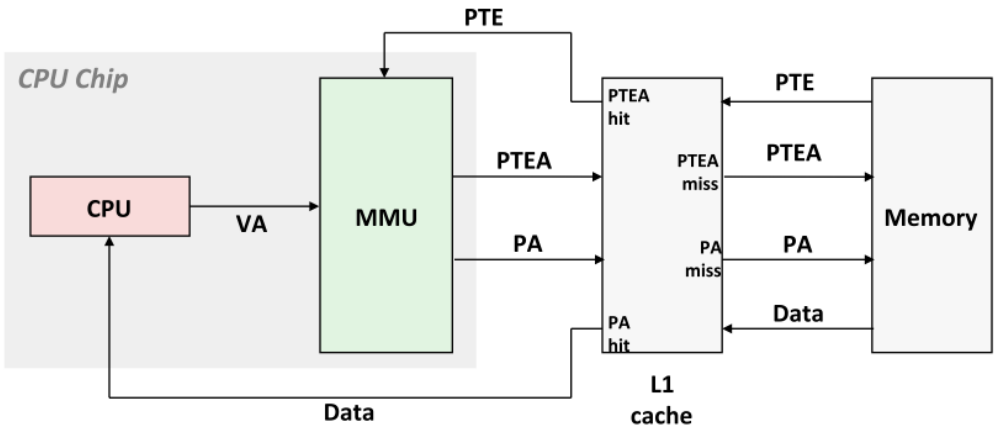

Figure 1.2: 有高速缓存时对数据的访问

由于局部性原理，cache 命中的概率很高，这样就减少了对内存的直接访问，从而提高了访问速度。

1.3 Linux 中的分页

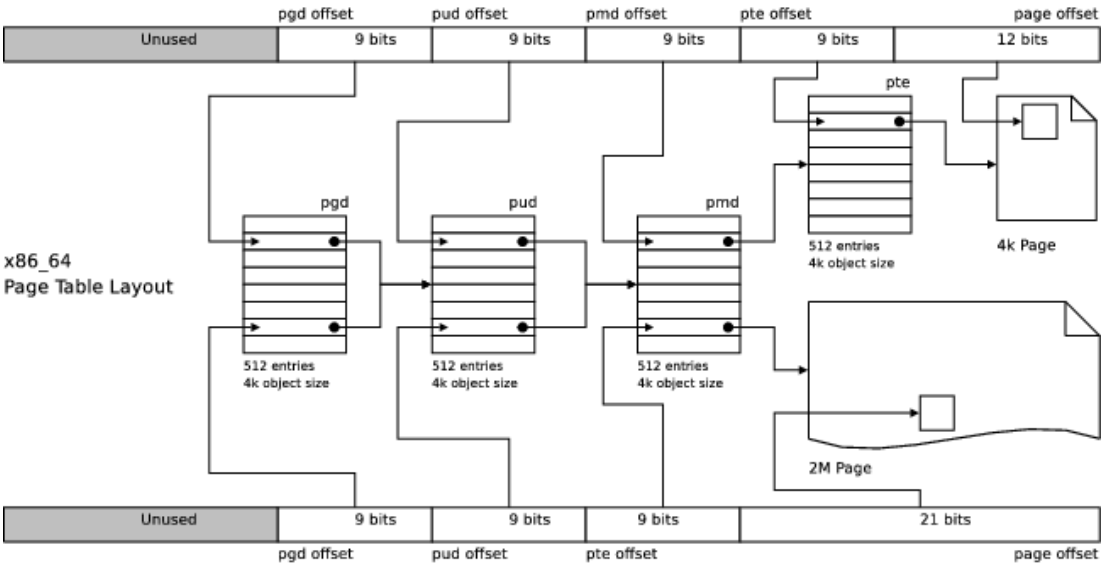

Figure 1.3: 虚拟地址转换 [?]

- PGD: 页全局目录 (Page Global Directory)
- PUD: 页上级目录 (Page Upper Directory)
- PMD: 页中间目录 (Page Middle Directory)
- PT: 页表 (Page Table)
- PTE: 页表项 (Page table entry)

页全局目录包含若干页上级目录的地址，页上级目录包含若干页中间目录的地址，而页中间目录又包含若干页表的地址。每一个页表项指向一个页框。线程地址被分成 5 部分。

1.4 PTE 结构

Core i7 的页表结构:

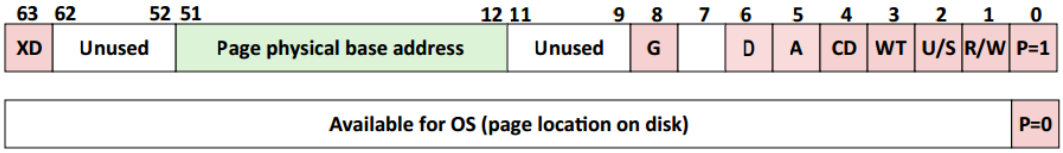

Figure 1.4: Core i7 Level 4 Page Table Entries

- P: Child page is present in memory (1)or not(0)
- R/W: Read-only or read-write access permission for child page
- U/S: User or supervisor mode access
- WT: Write-through or write-back cache policy for this page
- CD: Cache disabled (1) or enabled (0)
- A: Reference bit (set by MMU on reads and writes, cleared by sohware)
- D: Dirty bit (set by MMU on writes, cleared by sohware)
- G: Global page (don’t evict from TLB on task switch)
- Page physical base address: 40 most significant bits of physical page address (forces pages to be 4KB aligned)

参考文献

[1] http://linux-mm.org/PageTableStructure. 4