6.823 Computer System Architecture
Virtual Memory Implementation

http://csg.csail.mit.edu/6.823/

Hierarchical Page Table Supporting Variable-Sized Pages

Small fixed-sized pages (e.g. 4 KB) reduce internal fragmentation and the page fault penalty compared to large fixed-sized pages. However, when we run an application with a large working set, they may degrade a processor's performance by incurring a number of TLB misses because of their small *TLB reach*. Therefore, researchers have proposed to support variable-sized pages to increase the TLB reach without losing the benefits of small fixed-sized pages. Many modern processor families (e.g. UltraSparc, PA-RISC, MIPS) and operating systems (e.g. Sun Solaris, SGI IRIX) support this feature.

In this handout, we present an example implementation of variable-sized pages, supporting only two page sizes: 4 KB and 4 MB. Assume that the system uses 44-bit virtual addresses and 40-bit physical addresses. 4KB pages are mapped using a three-level hierarchical page table. 4MB pages are mapped using the first two levels of the same page table. An L2 Page Table Entry (PTE) contains information which indicates if it points to an L3 table or a 4MB data page. All PTEs are 8 Bytes. The following figure summarizes the page table structure and indicates the sizes of the page tables and data pages (not drawn to scale):

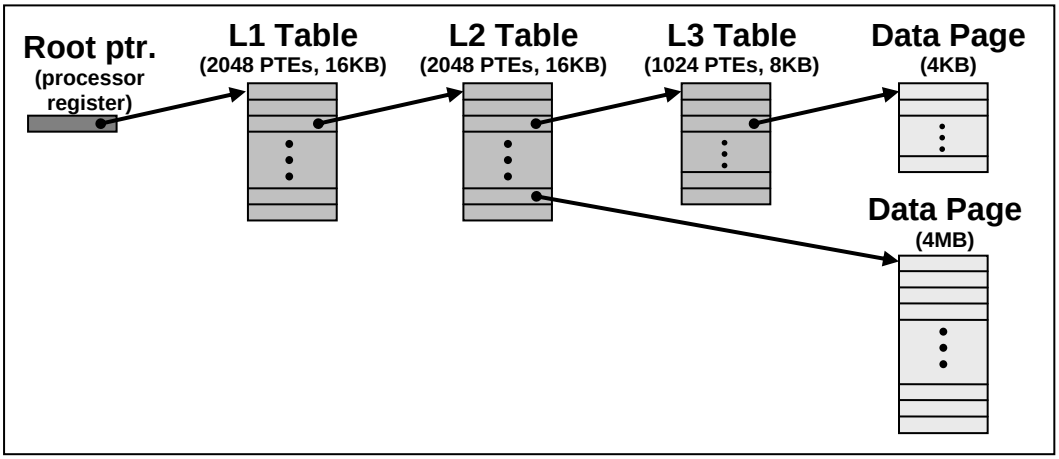

Figure H8-A. Example implementation of variable-sized pages.

Page Table Entries and Translation Lookaside Buffers

Each Page Table Entry (PTE) will map a virtual page number (VPN) to a physical page number (PPN). In addition to the page number translation, each page table entry also contains some permission/status bits.

Bit Name	Bit Definition
PPN / DBN	Physical Page Number / Disk Block Number
V (valid)	1 if the page table entry is valid, 0 otherwise
R (resident)	1 if the page is resident in memory, 0 otherwise
W (writable)	1 if the page is writable, 0 otherwise
U (used)	1 if the page has been accessed *recently*, 0 otherwise
M (modified)	1 if the page has been modified, 0 otherwise
S (supervisor)	1 if the page is only accessible in supervisor mode, 0 otherwise

Each entry in the Translation Lookaside Buffer (TLB) has a tag that is matched against the VPN and a TLB Entry Valid bit (note, the TLB Entry Valid bit is not the V bit shown in the table above). The TLB Entry Valid bit will be set if the TLB entry is valid. Each TLB entry also contains all the fields from the page table that are listed above.

A TLB miss (VPN does not match any of the tags for entries that have the TLB Entry Valid bit set) causes an exception. On a TLB miss kernel software will load the page table entry into the TLB and will restart the memory access. (Kernel software can modify anything in the TLB that it likes and always runs in supervisor mode). If the entry being replaced was valid, then the kernel will also write the TLB entry that is being replaced back to the page table.

Hardware will set the used bit whenever a TLB hit to the corresponding entry occurs. Similarly, the modified bit (in the TLB entry) will be set when a store to the page happens.

All exceptions that come from the TLB (hit or miss) are handled by software. For example, the possible exceptions are as follows:

TLB Miss:	VPN does not match any of tags for entries that have the TLB Entry Valid bit set.
Page Table Entry Invalid:	Trying to access a virtual page that has no mapping to a physical address.
Write Fault (Store only):	Trying to modify a read-only page (W is 0).
Protection Violation:	Trying to access a protected (supervisor) page while in user mode.
Page Fault:	Page is not resident.

(Unless noted, exceptions can occur for both loads and stores.)