Virtual Memory

This lesson discusses the difference between virtual and physical memory. Virtual memory is used by programs and must be translated to physical memory. Since the translation can be slow, a Translation Look-Aside Buffer (TLB) is used.

Why Virtual Memory

Hardware view of memory - the memory is a block of memory with a specific address allocated to each location.

Programmer's View of memory - memory is large arrays with many more addresses than the actual memory. Each program has its own view of the memory.

Virtual memory reconciles the programmer's view of memory with the hardware memory.

Processor's View of Memory

The processor sees the physical memory (actual memory).

- The actual amount of memory is sometimes < 4 GB
- Although the address is 64 bits, there is never this much memory (16 Exabytes/process)
- There is a 1:1 mapping of bytes to words

Program's View of Memory

Programs sees the stack and the heap, with a large gap between the two. The gap is never fully used. The programs see much more memory than the actual memory, so virtual memory is used.

Mapping Virtual → Physical Memory

Physical memory is divided into 4k Byte frames, the virtual memory is divided into 4k Byte pages.
The operating system uses page tables to map the pages to frames. If programs use the same physical memories, then the pages are mapped to the same page.

Where is the Missing Memory

Some of the virtual memory pages are actually stored on a hard disk, rather than in memory.

Virtual to Physical Translation

The virtual address is separated into the virtual page number (the MSB) and the page offset (LSB).
The virtual page is translated (using a page table) to the physical frame number. This physical frame number is combined with the page offset to determine the physical memory.

Size of Flat Page Table

Flat page tables = one entry for every page in the table .

Since some of these pages are never accessed, this table is unnecessarily large.
Page table size = (Virtual Memory / Page Size) * Size of Entry

Multi-Level Page Tables
Multi-level page tables are used for large virtual memories.
Multi-level page tables avoid having page entries for the unused virtual memory addresses.

Multi-Level Page Table Structure
The page number is partitioned into inner and outer page numbers, leading to a hierarchical page table.

Choosing the Page Size
Larger Pages:
- mean smaller page tables -which is good
- mean internal fragmentation - which is bad

Smaller Pages
- large page tables - bad

A compromise between the large and small pages means a page size that is a few KBytes to a few MBytes.

Memory Access Time with V→ P Translation
Additional time must be allowed for the virtual to physical address translation.
The extra steps required for translation are:
- -Computer the physical address of the page table entries (fast)
- -read the page table entry
- -compute the physical address (fast)

The translation can be longer than the memory access time.

Translation Look-Aside Buffer (TLB)
To speed up the translation, a special, small cache is used, called the TLB.

For a TLB miss the operating system (Software TLB Miss Handling) or the processor updates the TLB (Hardware TLB Miss Handling).

TLB Organization
A TLB is similar to a cache.
- TLB tends to be Fully or Highly Associative
- TLB Size = 64 to 512 entries (if more entries are needed, use a two level TLB)