Faculty of Computer Science and Engineering
University of Technology - VNUHCM

Virtual Memory

Trinh Van Giang 7140006
Bui Duc Hieu 7140231

November 14, 2014

Content

Outline

Introduction

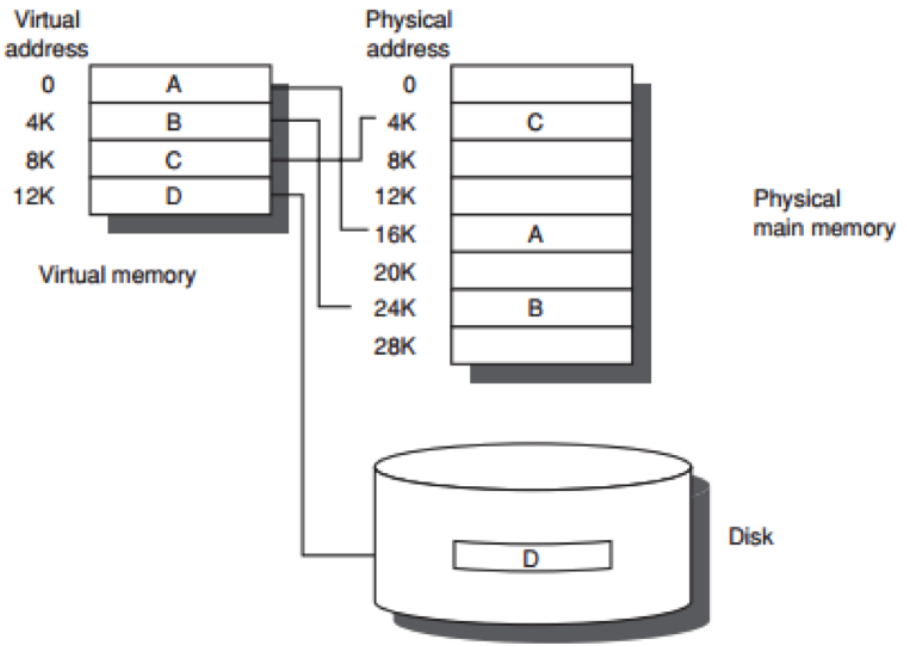
- Computers are running multiple processes with its own address space.
- It's too expensive to create full address space for all process.
- Each process use only small part of its address space.

Outline

Table of Contents

Virtual Memory

- Divides physical memory into blocks and allocates them to different processes.
- Memory management technique is implemented using both hardware and software.
- It maps memory addresses used by virtual addresses into physical addresses.



Benifit

- Freeing applications from having to manage a shared memory space.
- Increasing security due to memory isolation.
- Being able to conceptually use more memory than might be physically available.

Category

Page

- Fixed-size blocks
- 4096 to 8192 bytes

Segment

- Variable-size blocks
- Min: 1 byte
- Max: $2^{16} - 2^{32}$ bytes

Table of Contents

Paged Virtual Memory

- Divide a virtual address space into pages, blocks of contiguous virtual memory addresses.
- Systems with large virtual address ranges or amounts of real memory generally use larger page sizes.

Page table

- Used to translate the virtual addresses seen by the application into physical addresses like MMU.
- Each page table entry holds indexes whether the corresponding page is in real memory or not.
  - **Yes**, page table entry contain the real memory address at which the page is stored.
  - **No**, page fault exception.

Page supervisor

- Creates and manages page tables.
- If page fault exception, paging supervisor
  - Accesses secondary storage.
  - Returns page has virtual address that resulted in the page fault.
  - Updates the page tables to reflect the physical location of the virtual address.
  - Tells the translation mechanism to restart the request.
- If physical memory is full, paging supervisor must free a page page.
  - Use one of page replacement algorithms to determine which page to free.

Table of Contents

Segmented Virtual Memory

- Dividing virtual address spaces into variable-length segments.
- Consisting of a segment number and an offset within the segment.
- Segmentation and paging can be used together by dividing each segment into pages.
- Segmentation that can provide a single-level memory model in which there is no differentiation between process memory and file system consists of only a list of segments.

Table of Contents

Paging versus Segmentation

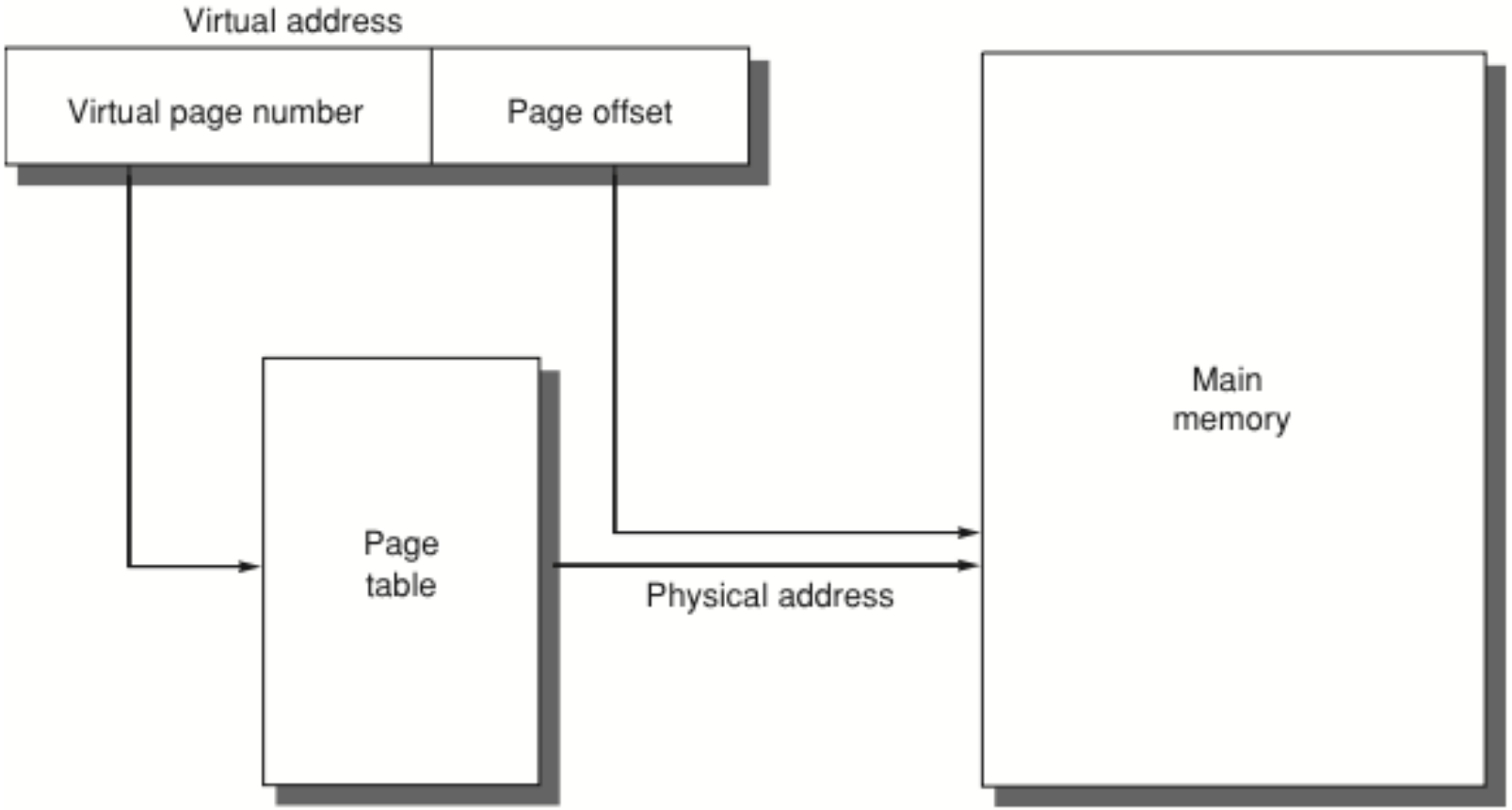
	Page	Segment
Words per address	One	Two (segment and offset)
Programmer visible?	Invisible to application programmer	May be visible to application programmer
Replacing a block	Trivial (all blocks are the same size)	Hard (must find contiguous, variable-size, unused portion of main memory)
Memory use inefficiency	Internal fragmentation (unused portion of page)	External fragmentation (unused pieces of main memory)
Efficient disk traffic	Yes (adjust page size to balance access time and transfer time)	Not always (small segments may transfer just a few bytes)

Outline

Q1: Where Can a Block Be Placed in Main Memory?

- The miss penalty for virtual memory involves access to a rotating magnetic storage device and is therefore quite high.
- Choice of lower miss rates or a ~~simpler placement algorithm~~.
- Operating systems allow blocks to be placed anywhere in main memory.

Q2: How Is a Block Found If It Is in Main Memory?



Q3: Which Block Should Be Replaced on a Virtual Memory Miss?

- Minimizing page faults $\Rightarrow$ replace the least-recently used (LRU)
- To help the operating system estimate LRU, many processors provide a use bit or reference bit, which is logically set whenever a page is accessed.

Q4: What Happens on a Write?

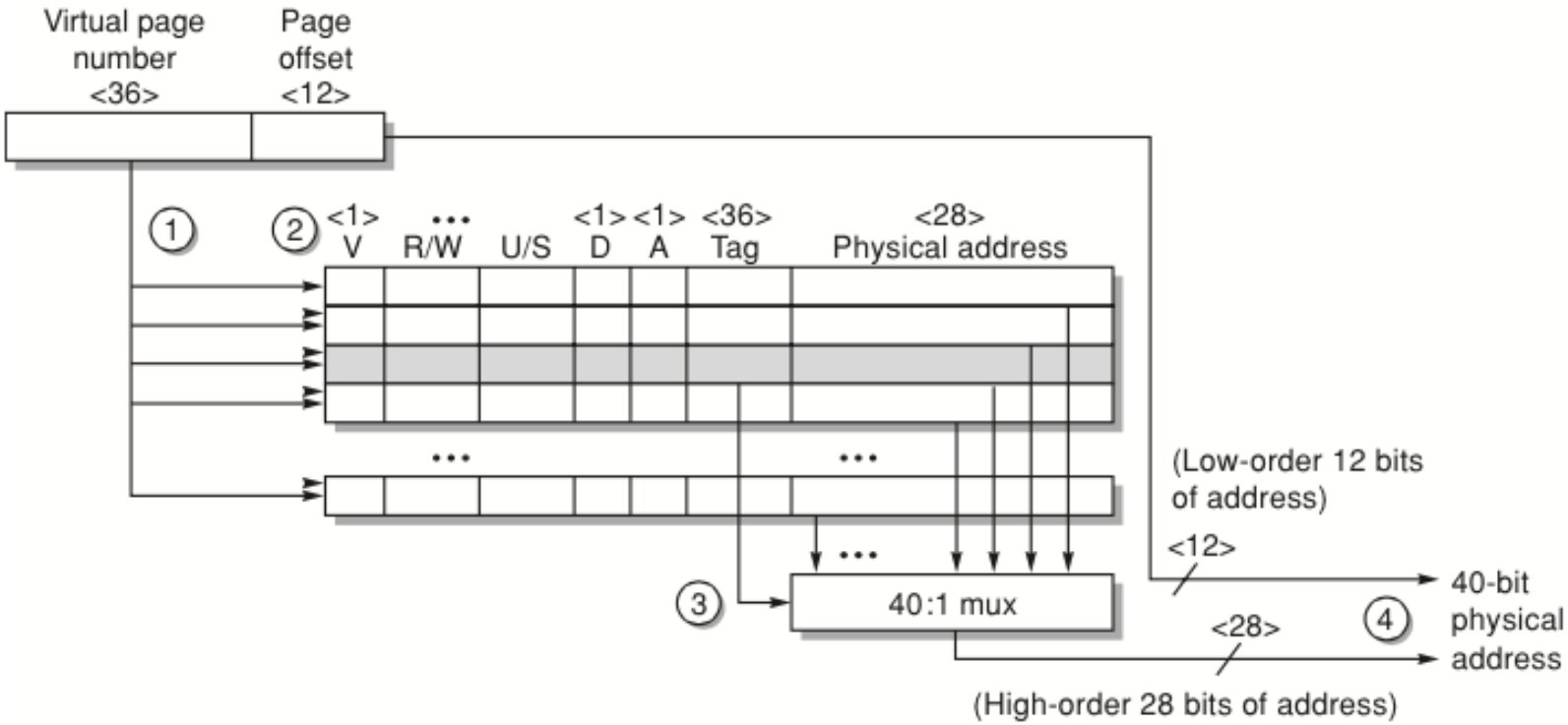
- The level below main memory contains rotating magnetic disks that take millions of clock cycles to access.
  - No one built a virtual memory operating system writes through main memory to disk on every store by the processor.
  - Write strategy is always write back.

Outline

Techiques

- Paging means that every memory access logically takes at least twice as long, with one memory access to obtain physical address and a second access to get data.
- Keeping address translations in a special cache
- Translation lookaside buffer (TLB) or translation buffer (TB)

TLB organization



References

[1] Hennessy, John L., and David A. Patterson *Computer architecture: a quantitative approach.* Elsevier, 2007.

[2] A.S. Tanenbaum *Modern Operating Systems* Second Edition, Prentice Hall, 2001.


Thank you for your attention!