Computer Organization & Architecture

3-7 Virtual Memory and Paging

Wang Guohua

School of Software Engineering

Contents of this lecture

- Virtual Memory
  - Why? What?
- One of the implementations of Virtual Memory — Paging

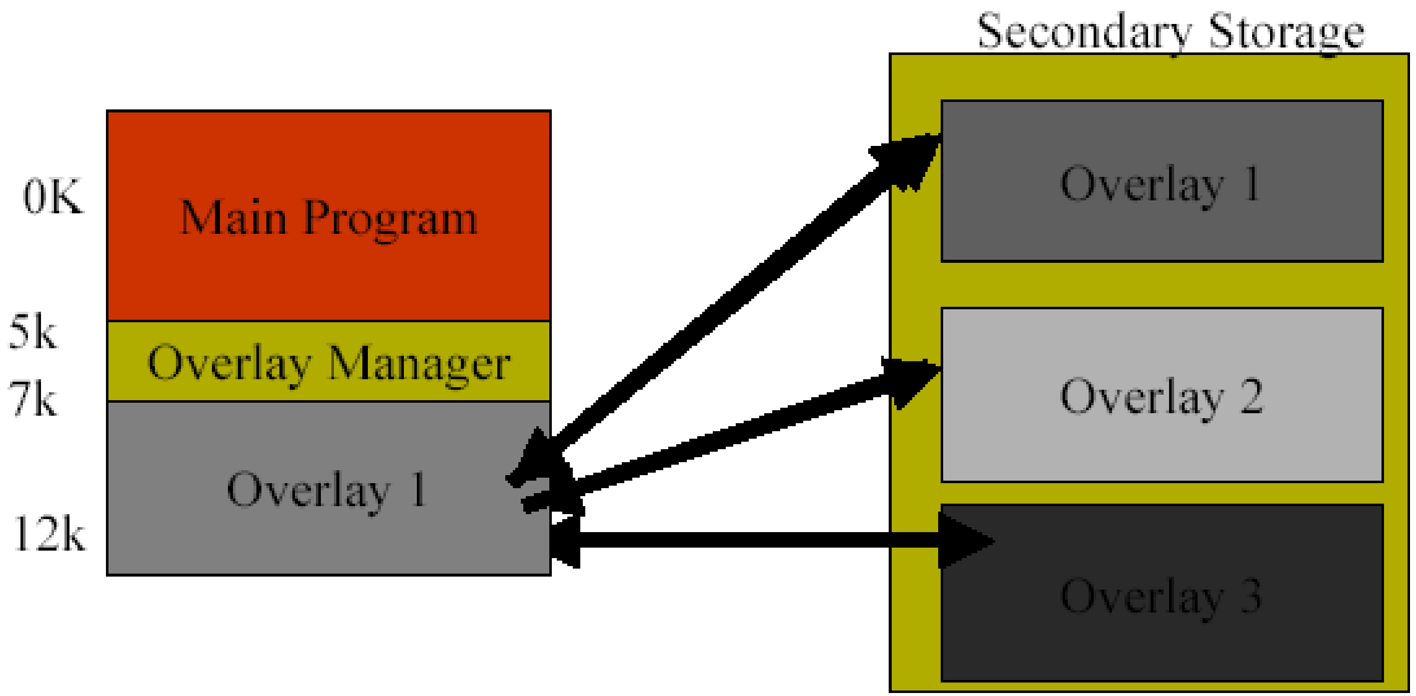
Overview

- Two Problems to Solve
  - What happens if the main memory (DRAM) is not sufficient for the execution of a program? Use disk? How?
  - What happens if one wishes to run several programs (processes) “at once”? How to load them in memory?

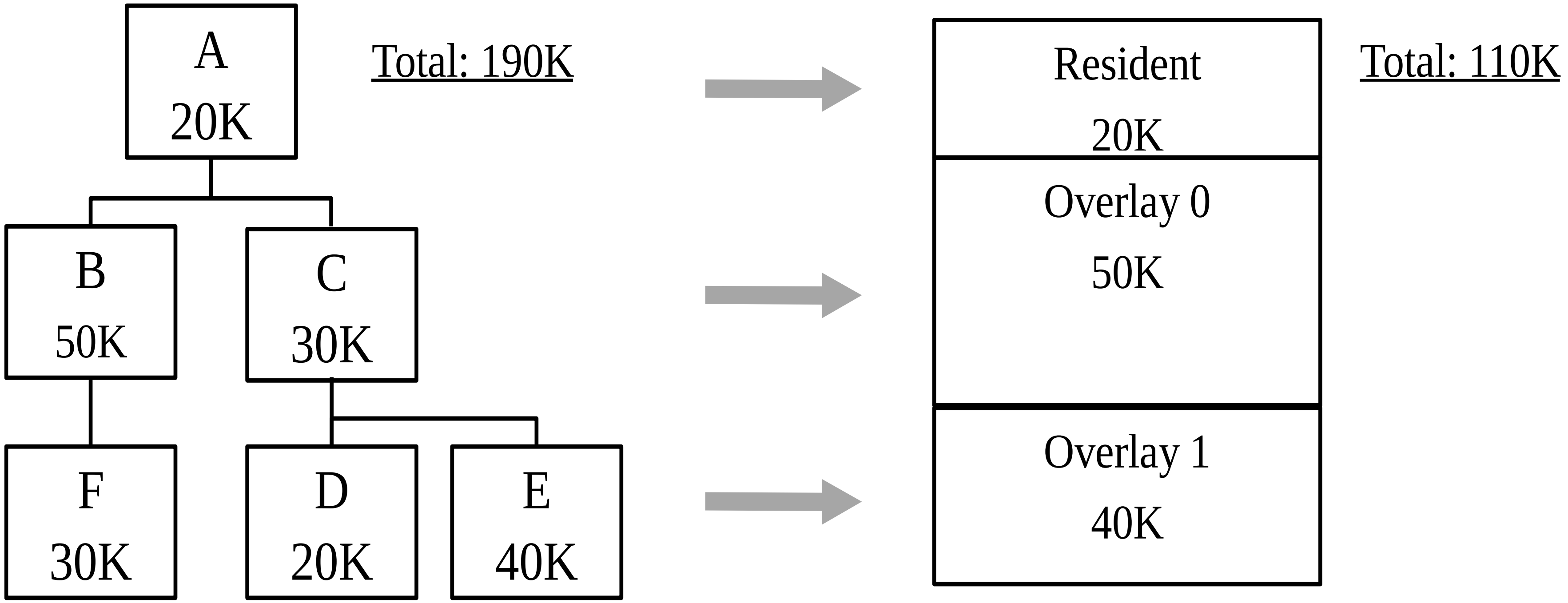
What was there before Virtual Memory?

- Overlaying
  - Programmers had to divide the program into several segments, and then use the principal of locality to choose segments to keep in memory during program execution.
  - Tedious to do by hand.
  - Difficult to determine overlay set, especially when considering multiprogramming.

What was there before Virtual Memory?



What was there before Virtual Memory?

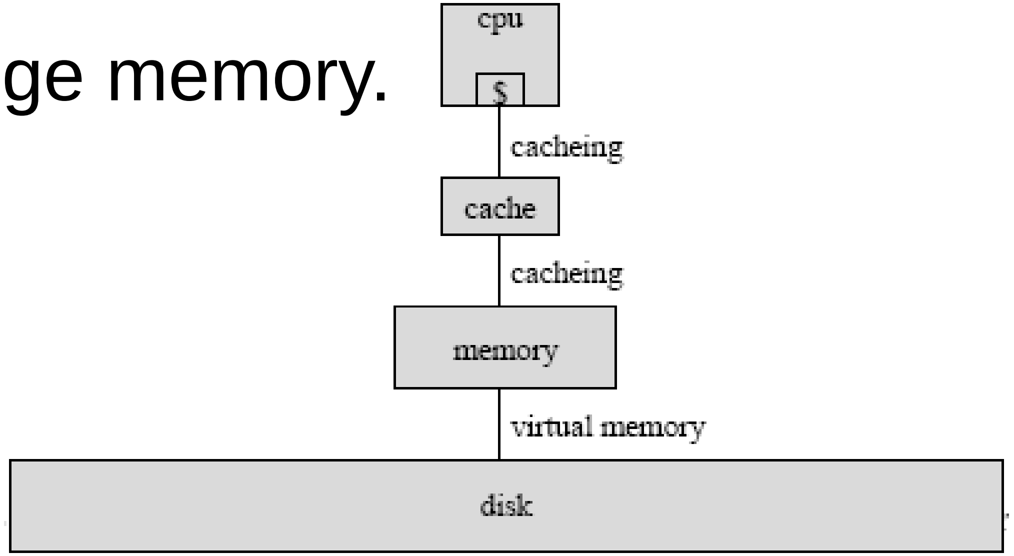


An alternative approach： (100K)

- A(20K)： 20K；
- B(50K)、D(20K)、E(40K)： 50K；
- C(30K)、F(30K)： 30K；

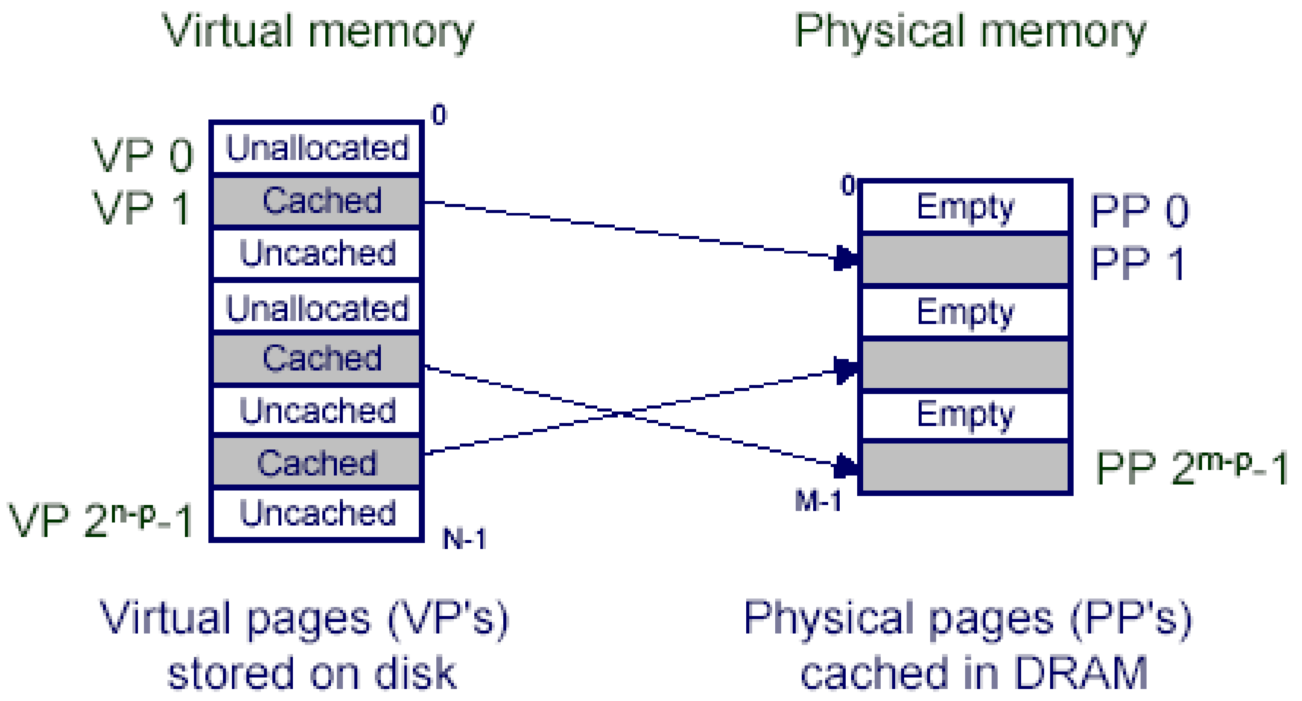
Virtual Memory (1)

- What is Virtual Memory
  - <u>Not</u> a physical device but an abstract concept.
  - Technique (an abstraction) for using disks to extend the apparent size of **physical memory** beyond it's actual physical size.
  - Goal: every process thinks it has a huge memory.



Virtual Memory (2)

- Virtual Memory as a Tool for Caching
  - Virtual memory is an array of N contiguous bytes stored on disk.
  - The contents of the array on disk are cached in physical memory.

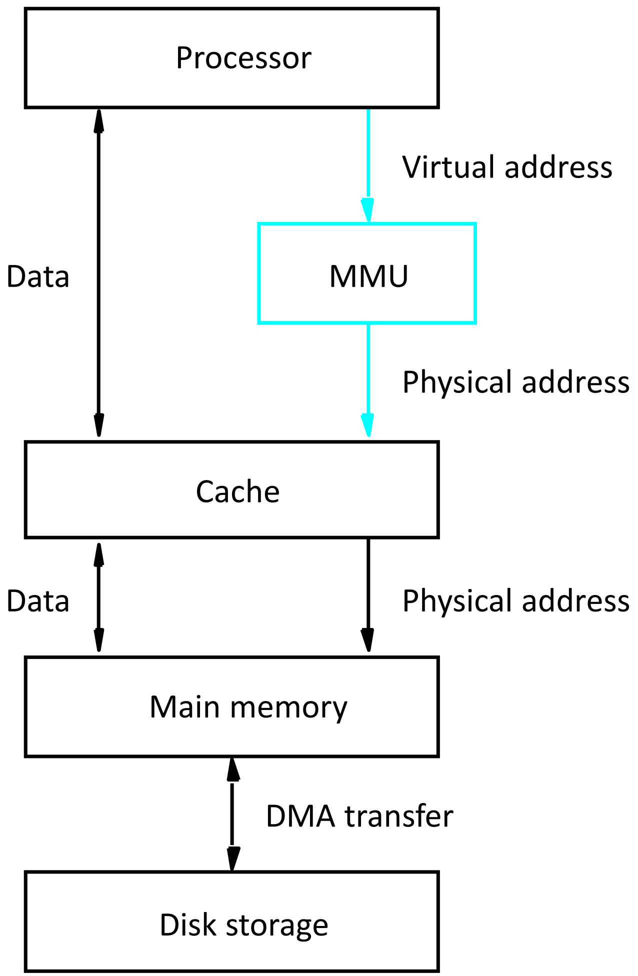


Virtual Memory (3)

- Relevant Terms
  - Physical Memory: memory actually available in the computer.
  - Logical or virtual Memory: memory that the OS allows a program to believe it has.
  - Physical Address: real location in physical memory; identifies actual storage.
  - Logical or virtual Address: conventional addressing used by a program which the OS must translate into a physical address.

Virtual Memory (4)

- Typical Organization of Virtual Memory



  - MMU (Memory Management Unit): The hardware converts virtual addresses into physical addresses via an OS-managed lookup table (page table).

Virtual Memory (5)

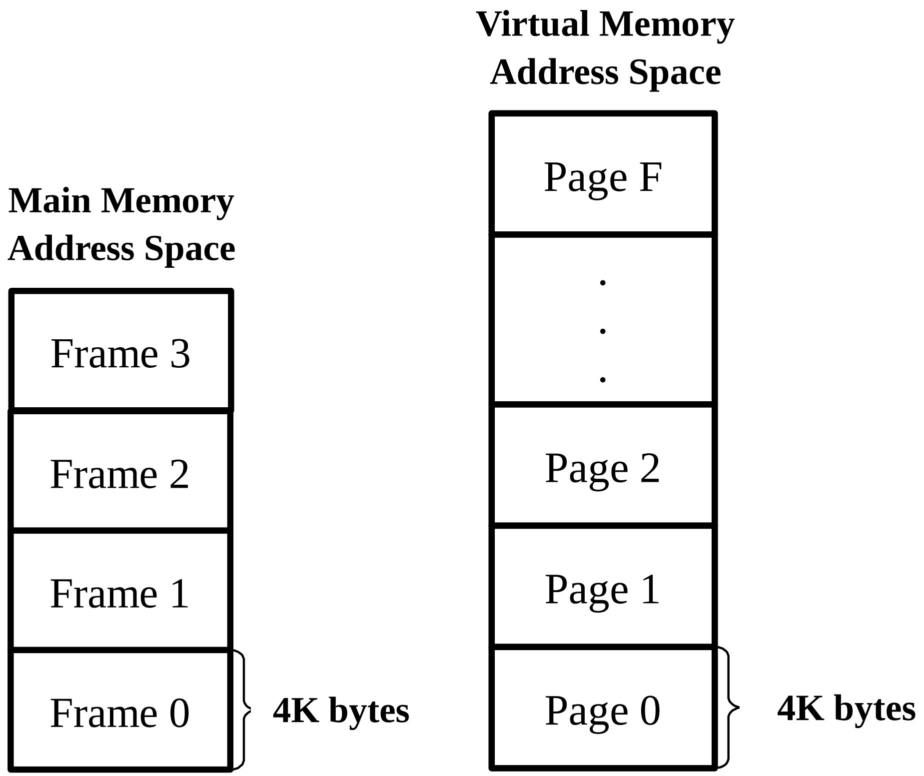
- Benefits of Virtual Memory
  - Large address space: Virtual address to Physical Address mapping.
  - Simplify relocation: the same program can run in any location in the physical memory.
  - Reduce time to start a program: not all code and data need to be in the physical memory before a program can start.
  - Relieve programmers from burden of overlays: VM automatically manages the two levels of memory hierarchy represented by main memory and secondary storage.

Implementation of Virtual Memory

- Implementation Methods
  - Paging
  - Segmentation
  - Segmentation with paging

Paging (1)

- Organization of Virtual Memory and Main Memory
  - Page: a block of consecutive words
  - Page Frame: an area in the main memory that can hold one page



Paging (2)

- Note
  - Pages constitute the basic unit of information that is moved between the main memory and the disk.
  - Pages commonly range from 2K to 16K bytes in length.
  - Pages should not be too small or too large.

Paging (3)

- Address Structure
  - Virtual Address (2 fields)
    - Virtual page number (high-order bits)
    - Offset (low-order bits): Specify the location of a particular byte (or word) within a page
  - Physical Address (2 fields)
    - Physical page number (high-order bits)
    - Offset (low-order bits)

Paging (4)

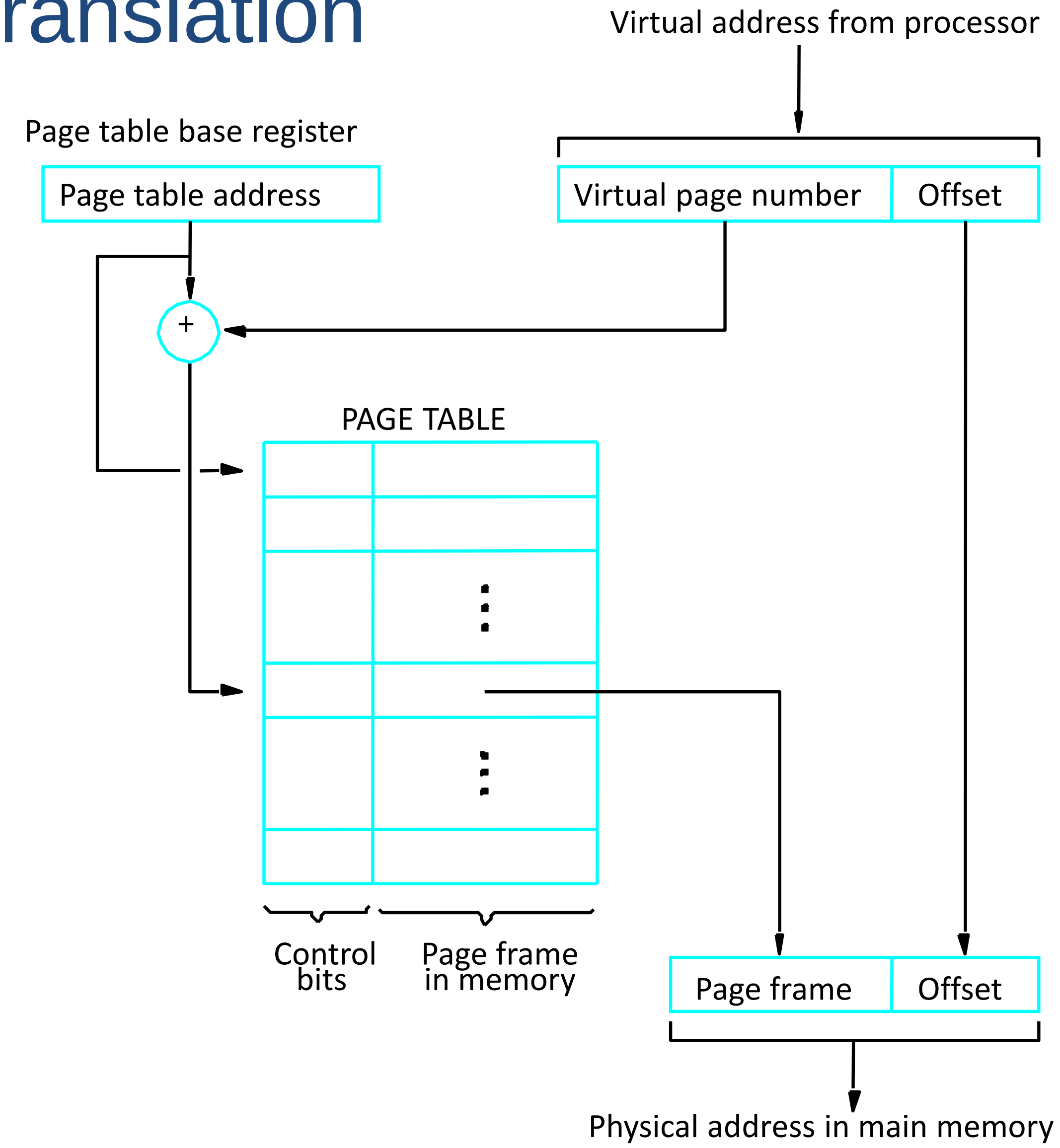
- Page Table
  - A page table is an array of page table entries (PTEs) that maps virtual pages to physical pages.
  - The virtual page number is used as an index into the page table to find the entry for that virtual page.
  - The page table is kept in the main memory.
  - *Page Table Base Register*: points to the page table's location in memory.

Paging (5)

- Page Table Entry
  - *Page frame number*: Physical page number OR pointer to secondary storage
  - *Valid bit (Presence bit)*: Indicate whether the page is actually loaded in the main memory
  - *Modify bit*: Indicate whether the page has been modified during its residency in the main memory
  - *Use bit (Reference bit)*: Indicate whether the page has been used recently
  - *Access Control bit*: Read, write, execute, etc.

Paging (6)

- Address Translation



Paging (7)

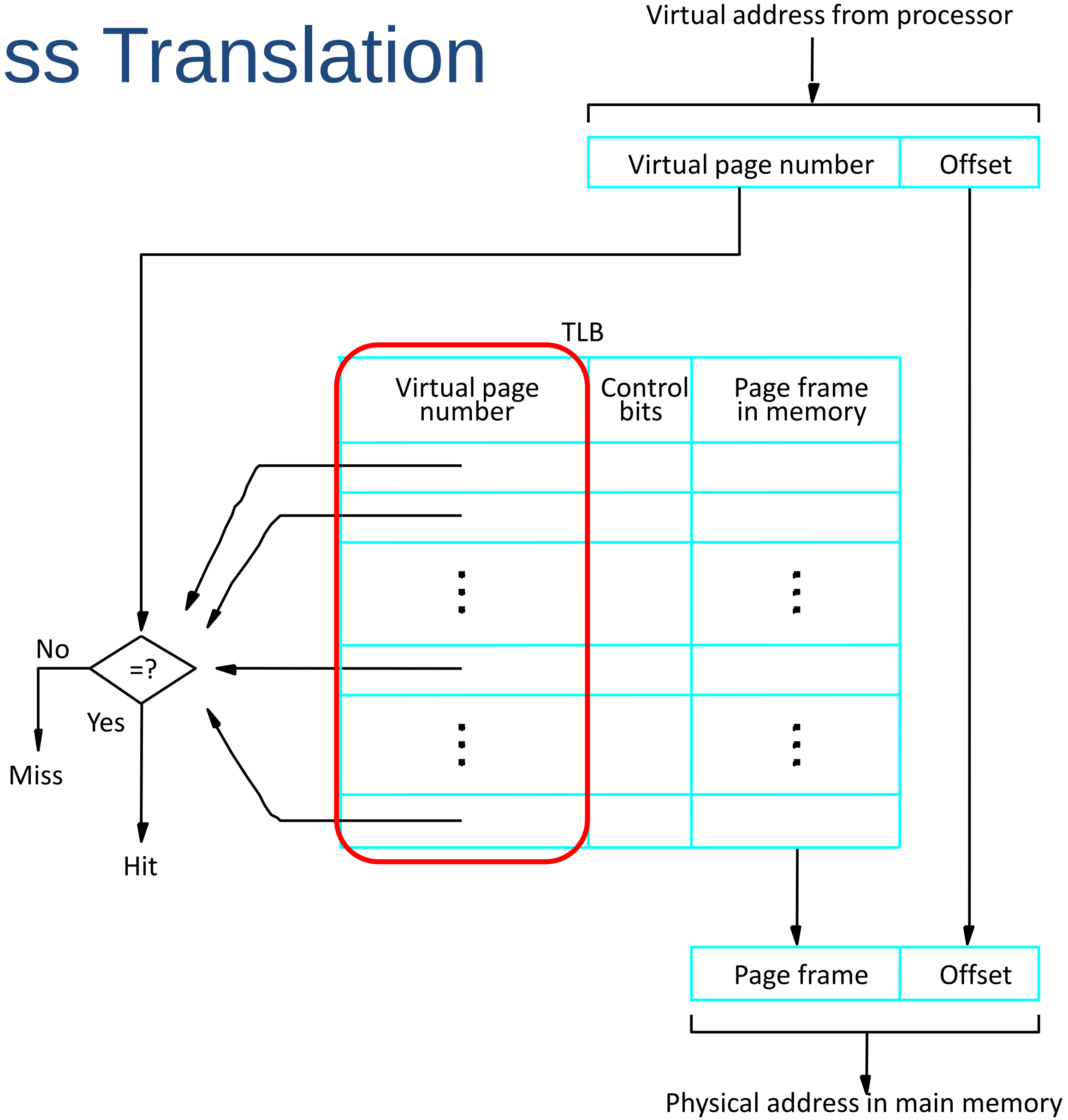
- Page Hit
  - A *page hit* is a reference to a VM word that is in physical main memory
- Page Fault (Similar to "Cache Miss")
  - A *page fault* is caused by a reference to a VM word that is not in physical main memory

Paging (8)

- Disadvantage of the page table in main memory
  - Every memory access by a program can take at least twice as long: one memory access to obtain the physical address and a second access to get the data.
- TLB (Translation Look-aside Buffer)
  - A small cache which is incorporated in MMU consists of a small portion of the page table.

Paging (9)

- TLB Address Translation



Paging (10)

- TLB Hit
  - A TLB hit eliminates a memory access
- TLB Miss
  - Because the TLB has many fewer entries than the number of pages in main memory, TLB misses will be much more frequent than true page faults.
  - A TLB miss incurs an additional memory access
- TLB Coherent with Page Table
  - When OS changes the contents of page tables, it must simultaneously invalidate the corresponding entries in the TLB.

Paging (11)

- Page Replacement
  - LRU (Needs *Use* bit)
  - FIFO
  - Software (OS) implementation
- Write Policy
  - Write back
  - Write through is not suitable for virtual memory

Paging (12)

- Disadvantage
  - Internal fragmentation problem
    - Last page is unlikely to be full
    - If the page size is *n* bytes, the average amount of space wasted in the last page of a program by internal fragmentation will be *n/2 bytes.*

Paging (13)

- Page Size
  - Choosing a page size is a question of balancing forces that favor a larger page size versus those favoring a smaller size.

- Advantages of choosing larger page size
  - The size of the page table is inversely proportional to the page size: Memory can therefore be saved by making the pages bigger.
  - Transferring larger pages to or from secondary storage, possibly over a network, is more efficient than transferring smaller pages.

Paging (14)

- Disadvantages of choosing larger page size
  - A large page size will result in more wasted storage (internal fragmentation) when a contiguous region of virtual memory is not equal in size to a multiple of the page size.

Quiz (1)

1. Which one of the following about benefits of virtual memory is *NOT* true?
    A. provide large address space
    B. relieve programmers from burden of overlays
    C. resolve internal fragmentation
    D. simplify relocation

Quiz (2)

2. In a paging system, which of the following is *NOT* true?

   A. When a TLB miss occurs, the operating system must copy the requested page from the disk into the main memory

   B. Transferring larger pages to or from secondary storage is more efficient than transferring smaller pages

   C. Paging has a problem called internal fragmentation

   D. “Write though” is not suitable for this system

Quiz (3)

3. A processor uses 46-bit virtual addresses with 2MB pages. Which bits in the virtual address correspond to the “offset” field?

   A. The most significant 34 bits

   B. The least significant 12 bits

   C. The most significant 25 bits

   D. The least significant 21 bits

4. A page fault is _____.

   A. an attempt by the computer to run instructions stored on the hard disk

   B. the process the computer uses to start itself

   C. an error the computer makes when a device driver is missing

   D. an out of memory error

Quiz (4)

5. About TLB, which of the following is true?

   A. It’s a small cache which consists of a small portion of the page table

   B. When a TLB miss occurs, the operating system must copy the requested page from the disk into the main memory

   C. TLB misses can only be handled in hardware

   D. Its content is accessed based on the address

6. Suppose the paging hardware with TLB has a 90 percent hit ratio. Page numbers found in the TLB have a total access time of 100 ns. Those which are not found there have a total access time of 200 ns. What is the average access time?

   A. 100ns B. 110ns C. 190ns D. 200ns

Quiz (5)

7. *True or False?* A Translation Lookaside Buffer (TLB) acts as a cache for the page table.
8. What are advantages and disadvantages of choosing smaller page size in a paging system?

    Advantages: A small page size will result in less wasted storage when a contiguous region of virtual memory is not equal in size to a multiple of the page size.

    Disadvantages:

    (1) The size of the page table is inversely proportional to the page size. Memory can therefore be wasted by making the pages smaller.

    (2) Transferring smaller pages to or from secondary storage, possibly over a network, is less efficient than transferring larger pages.

Quiz (6)

9. Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

   A page fault occurs when an access to a page that has not been brought into main memory takes place.

   The operating system verifies the memory access, aborting the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted.