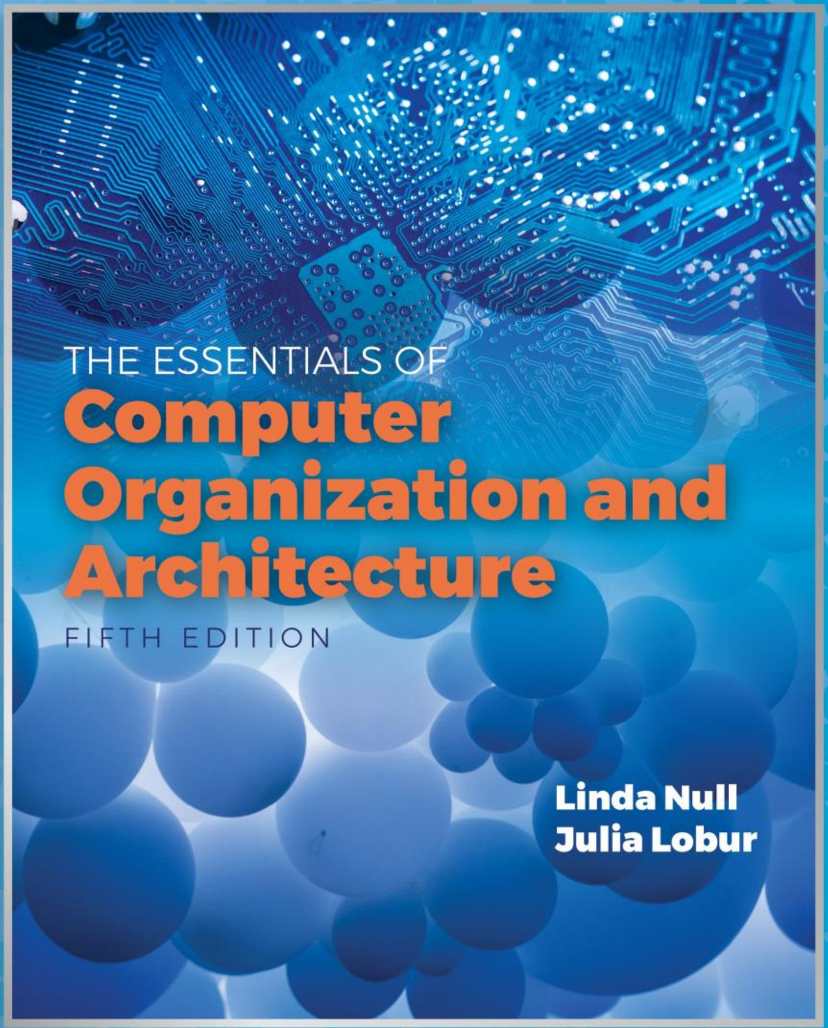
This is the sixth lecture of Chapter 6

Chapter 6

Memory (F)



Quick review of last lecture

- Two examples to compute hit rate and EAT for a given program

6.5 Virtual Memory (1 of 26)

Concept and Terminology (1)

- Cache memory enhances performance by providing faster memory access speed.
- Virtual memory enhances performance by providing greater memory capacity, without the expense of adding main memory.
- Instead, a portion of a disk drive serves as an extension of main memory.
- If a system uses paging, virtual memory partitions main memory into individually managed page frames, that are written (or *paged*) to disk when they are not immediately needed.

6.5 Virtual Memory (2 of 26)

Concept and Terminology (2)

- A *physical address* is the actual memory address of physical memory.
- Programs create *virtual addresses* that are *mapped* to physical addresses by the memory manager.
- *Page faults* occur when a logical address requires that a page be brought in from disk.
- *Memory fragmentation* occurs when the paging process results in the creation of small, unusable clusters of memory addresses.

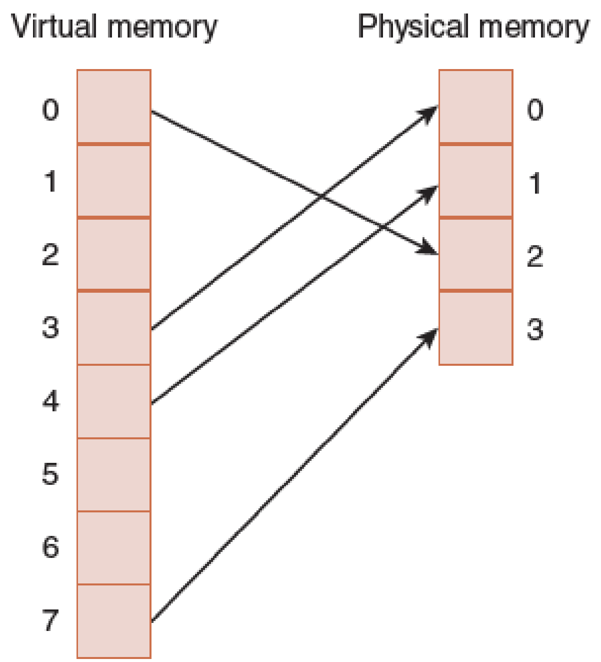
6.5 Virtual Memory (3 of 26)

Concept and Terminology (3)

- Main memory and virtual memory are divided into equal sized pages.
- The entire address space required by a process need not be in memory at once. Some parts can be on disk, while others are in main memory.
- Further, the pages allocated to a process do not need to be stored contiguously—either on disk or in memory.
- In this way, only the needed pages are in memory at any time, the unnecessary pages are in slower disk storage.

6.5 Virtual Memory (4 of 26)

- Information concerning the location of each page, whether on disk or in memory, is maintained in a data structure called a *page table* (shown below).
- There is one page table for each active process.



Page table

Page	Frame #	Valid Bit
0	2	1
1	-	0
2	-	0
3	0	1
4	1	1
5	-	0
6	-	0
7	3	1

6.5 Virtual Memory (5 of 26)

How the system access data (1)

- When a process generates a virtual address, the operating system translates it into a physical memory address.
- To accomplish this, the virtual address is divided into two fields: A *page field*, and an *offset field*.
- The page field determines the page location of the address, and the offset indicates the location of the address within the page.
- The logical page number is translated into a physical page frame through a lookup in the page table.

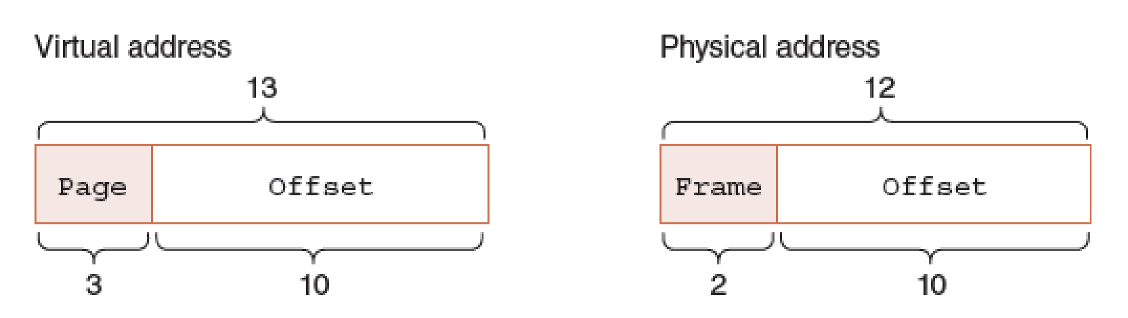
6.5 Virtual Memory (6 of 26)

How the system access data (2)

- If the valid bit is zero in the page table entry for the logical address, this means that the page is not in memory and must be fetched from disk.
  - This is a page fault.
  - If necessary, a page is evicted from memory and is replaced by the page retrieved from disk, and the valid bit is set to 1.
- If the valid bit is 1, the virtual page number is replaced by the physical frame number.
- The data is then accessed by adding the offset to the physical frame number.

6.5 Virtual Memory (7 of 26)

- As an example, suppose a system has a virtual address space of 8K and a physical address space of 4K, page size is 1k, and the system uses byte addressing.
  - We have $2^{13}/2^{10} = 2^3$ virtual pages.
- A virtual address has 13 bits ($8K = 2^{13}$) with 3 bits for the page field and 10 for the offset, because the page size is 1024.
- A physical memory address requires 12 bits, the first 2 bits for the page frame and the trailing 10 bits the offset.

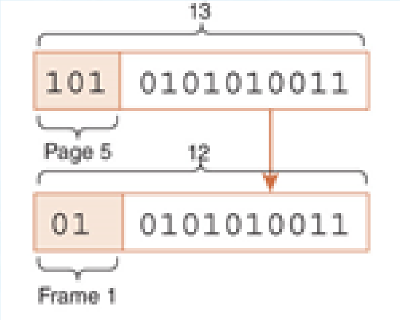


6.5 Virtual Memory (8 of 26)

- What happens when the CPU generates address $5459_{10} = 1010101010011_2 = 0x1553$?
- Page table is given below.

B Page table

Page	Frame	Valid bit
0	-	0
1	3	1
2	0	1
3	-	0
4	-	0
5	1	1
6	2	1
7	-	0



5459

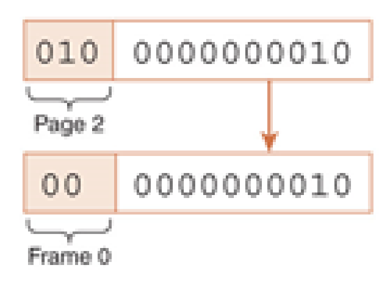
1363

6.5 Virtual Memory (9 of 26)

- What happens when the CPU generates address 2050_{10} = 010000000010_2 = 0x802?
- Page table is given below.

B Page table

Page	Frame	Valid bit
0	-	0
1	3	1
2	0	1
3	-	0
4	-	0
5	1	1
6	2	1
7	-	0



2050

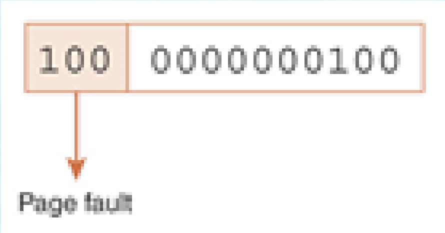
2

6.5 Virtual Memory (10 of 26)

- What happens when the CPU generates address $4100_{10} = 1000000000100_2 = \text{0x}1004$?
- Page table is given below.

B Page table

Page	Frame	Valid bit
0	-	0
1	3	1
2	0	1
3	-	0
4	-	0
5	1	1
6	2	1
7	-	0



4100

6.5 Virtual Memory (11 of 26)

Example 6.11

- Suppose a computer has 32-bit virtual address, pages of size 4k, 1G of byte addressable main memory. Given the partial page table below, convert virtual address 0x0011232A to physical address

Page table

Page	Frame	Valid
0000	00000	1
...	...	...
00111	0A121	1
00112	3F00F	1
00113	2AC11	1
...	...	...

Virtual address 0x0011232A