Revised: 8/15/2024

Memory Architecture III

Michael C. Hackett

Assistant Professor, Computer Science

COMMUNITY COLLEGE OF PHILADELPHIA

Lecture Topics

- Virtual Machines
- Virtual Memory
  - Page Faults
  - Exceptions
- Rotational Hard Disks

Virtual Machines

- A **virtual machine** (VM) is a software-based computer that emulates all major functions of a physical computer system.

- *Process virtual machines* are VMs that only run a single program
  - Like the Java Virtual Machine runs Java programs, but not an entire operating system with emulated hardware.

- *System virtual machines* emulate an entire computer system.
  - Software like VMware and VirtualBox are System VMs

Virtual Machines

- A *hypervisor* (or virtual machine manager) is software that manages virtual machines

- The underlying physical computer that the virtual machine runs on is called the *host*.

Virtual Machines

- Virtual machines have become significant in that they:
  - Allow multiple “machines” to run on (and share) one physical server
  - These machines might have entirely different software and operating system configurations
    - One physical server, many environments to test software on
  - Virtual machines can run on the same server but are isolated from each other.
    - If one is compromised, it doesn’t necessarily compromise the entire server.

Virtual Machines

- Many instruction sets were not created with virtualization in mind.
  - X86, MIPS, and ARM, to name a few

- The virtual machine cannot execute ISA-level instructions on the host system if the ISA does not support virtualization.
  - The virtualized system can only interact with virtualized resources

Virtual Machines

- Many instruction sets were not created with virtualization in mind.
  - X86, MIPS, and ARM, to name a few

- The virtual machine cannot execute ISA-level instructions on the host system if the ISA does not support virtualization.
  - The virtualized system can only interact with virtualized resources

Virtual Memory

- **Virtual memory** is when the system’s main memory gets used as cache for secondary storage devices (i.e., hard drives)

- Virtual memory allows for programs to only use a portion of main memory.
  - The program’s address space is a range of virtual memory addresses available to the program.

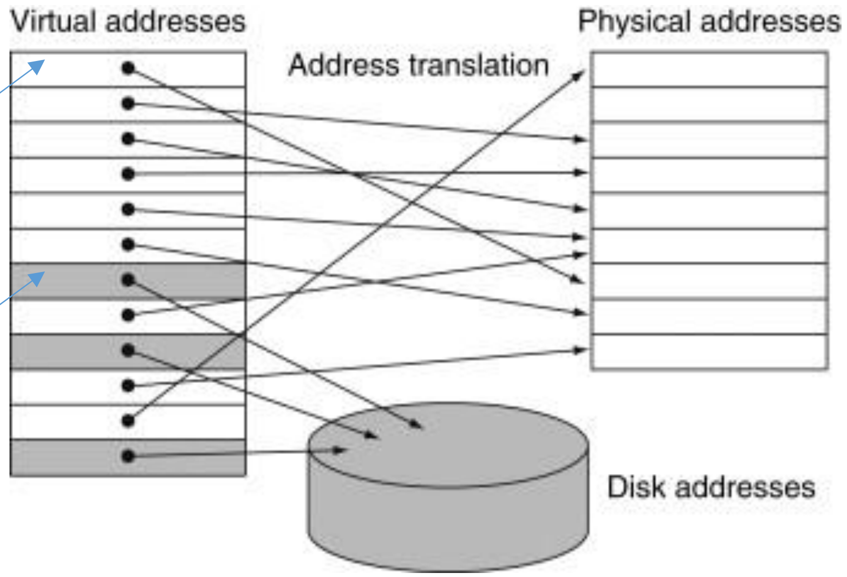
- Virtual memory translates the virtual addresses to real, physical addresses of main memory
  - Called *address mapping* or *address translation*

Virtual Memory

- Virtual memory also expands the capacity of main memory.
  - Data that can’t fit in main memory are stored in secondary storage devices.

Data stored in main memory

Data stored on the hard disk



Virtual Memory

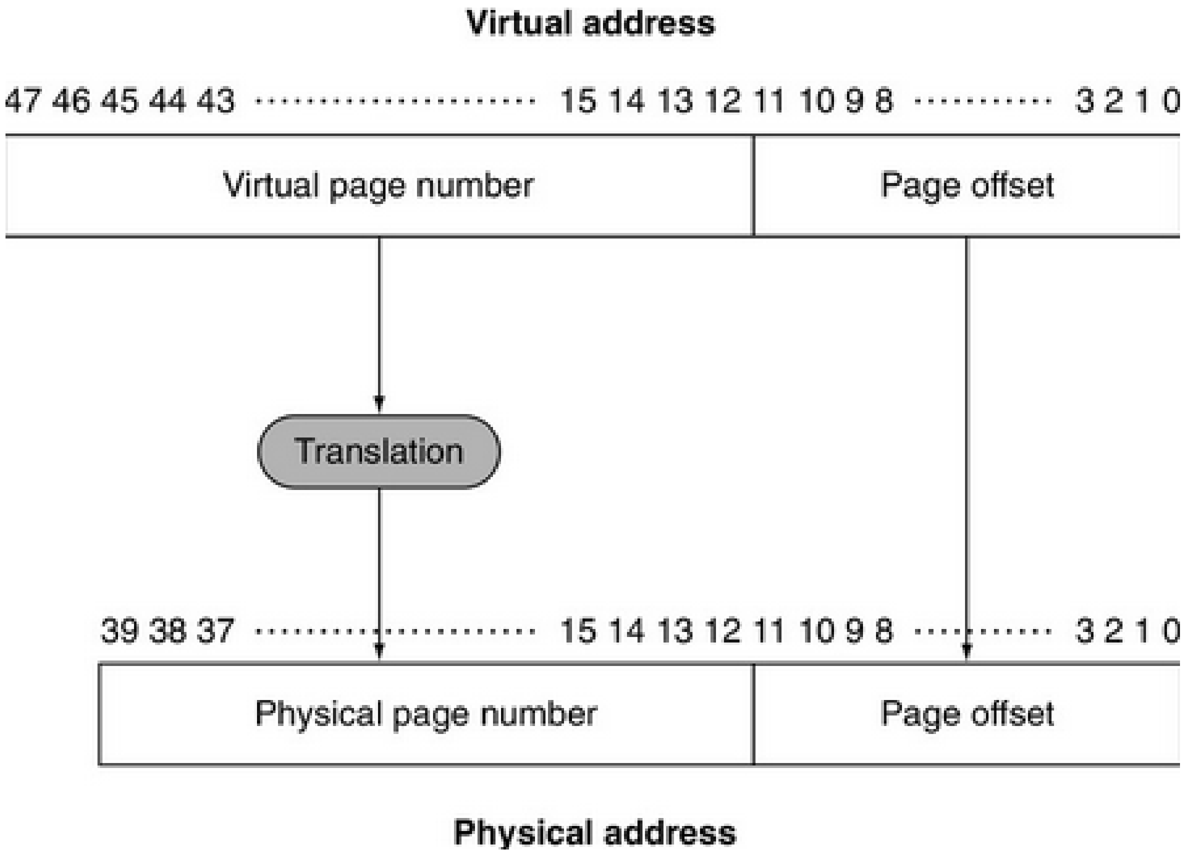
- Virtual memory and caches operate on similar principles.
- Blocks of virtual memory are referred to as *pages.*
- Like when cache memory experiences a cache miss, virtual memory experiences a *page fault* when the page sought is not present.

Virtual Memory

- Virtual memory addresses are split into two sections: the virtual page number and page offset

- The virtual page number is translated to a physical page number
  - The physical page number will be the upper portion of a physical address
  - The page offset remains unchanged and is the lower portion of a physical address.

Virtual Memory



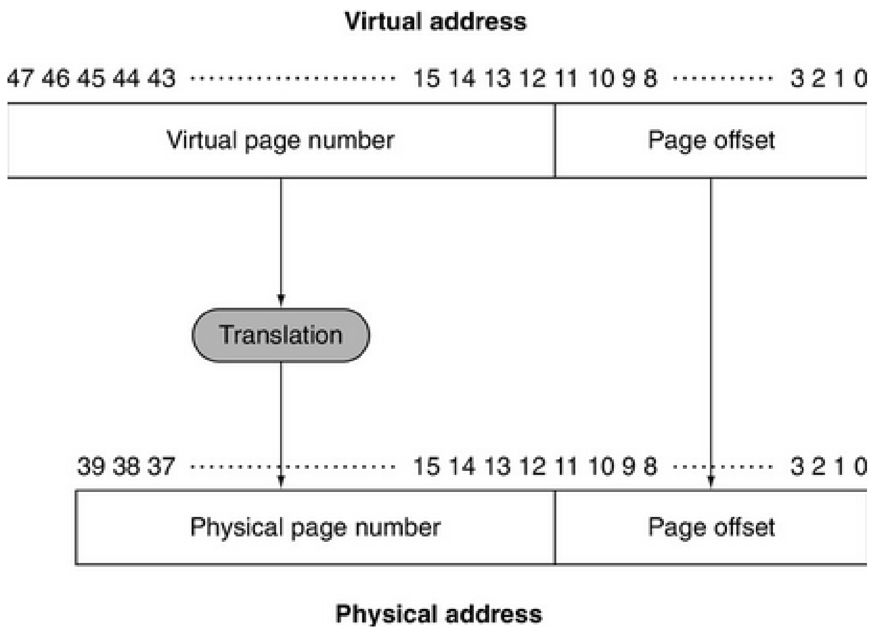
Virtual Memory

$$Page\ Size = 2^{Number\ of\ Offset\ Bits}$$

$$Physical\ Pages = 2^{(Physical\ address\ length\ -\ Number\ of\ Offset\ Bits)}$$

$$Address\ Space = 2^{(Address\ length)}$$

Virtual Memory



- $Page\ Size = 2^{12} = 4\ KiB$
- $Physical\ Pages = 2^{40-12} = 2^{28}$
- Virtual Memory
  - $Address\ Space = 2^{48} = 256\ TiB$
- Physical Memory
  - $Address\ Space = 2^{40} = 1\ TiB$

Virtual Memory

- Page faults to the hard disk are extremely time consuming.
  - By comparison, a page fault to main memory is about 100,000 times faster
- Pages should be large enough to compensate for high access times.
  - 4KiB to 16KiB are typical
- Optimizing page placement reduces the frequency of page faults.

Virtual Memory

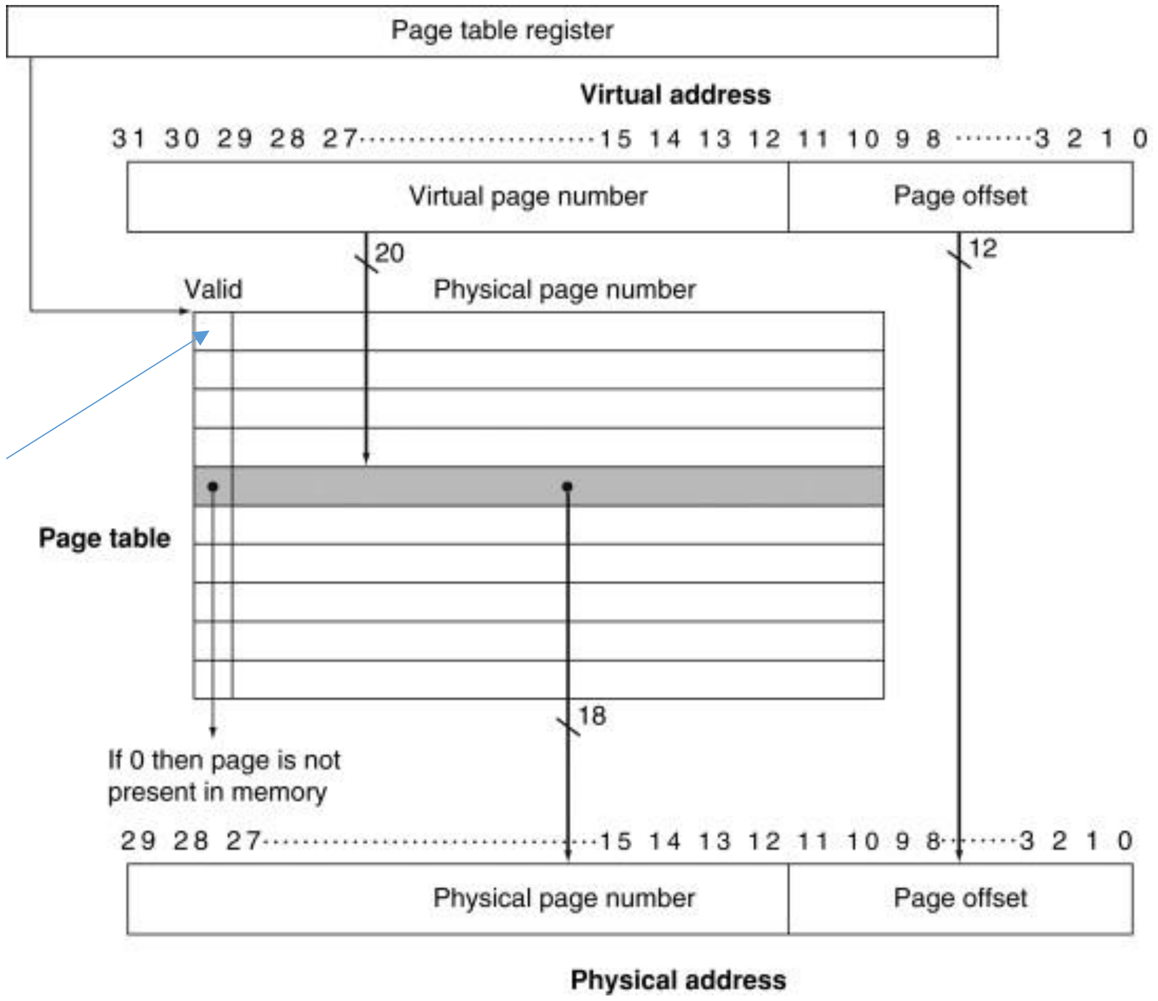
- Virtual memory systems use a *page table* to keep track of the virtual-to-physical memory translations.
  - Usually indexed by page number
  - Each table entry contains the physical page number for a virtual page number, provided the page is in memory

- Each program has its own page table that maps the virtual address space to physical memory addresses.

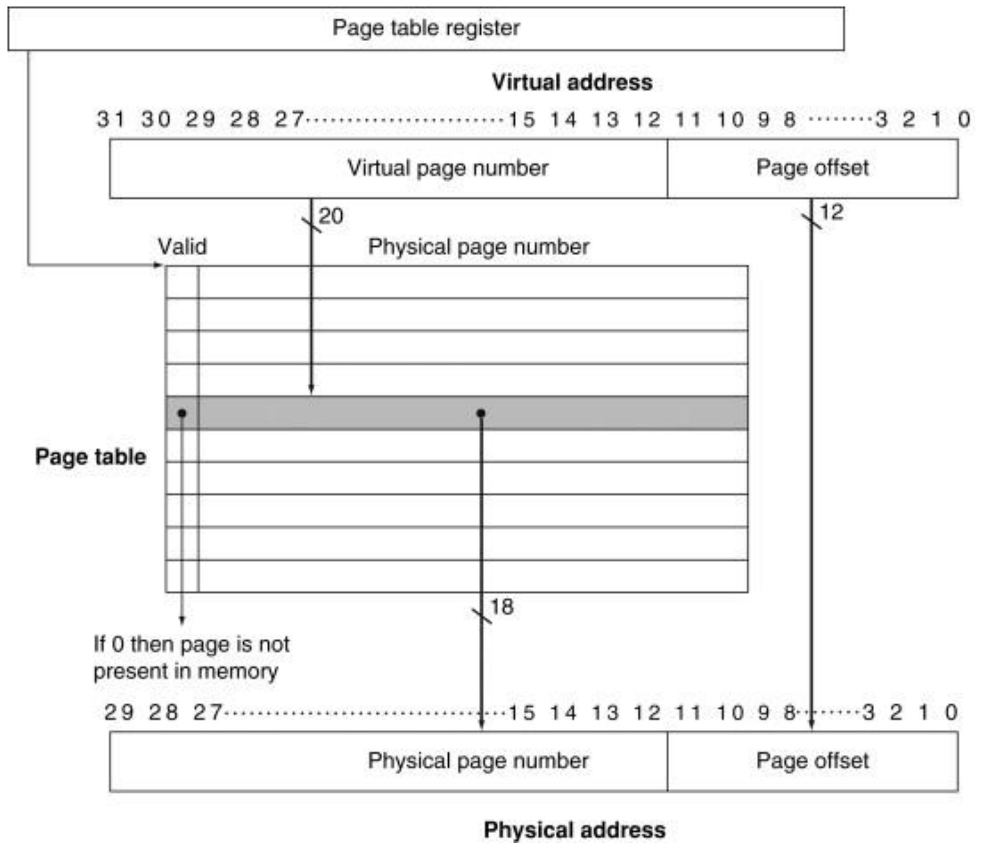
- The *page table register* points to the start of the page table

Virtual Memory

The valid bit is like valid/dirty bits in cache memory.
0 will indicate a page fault.



Virtual Memory



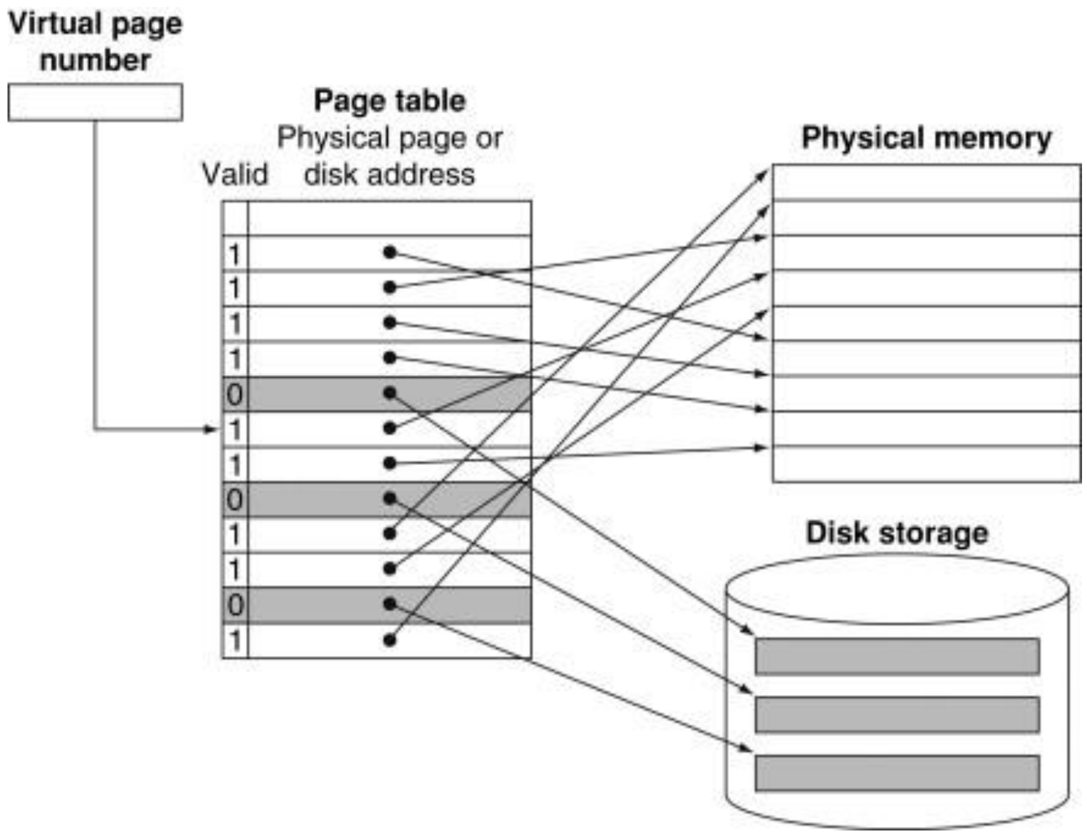
$$Page\ Table\ Entries$$
$$= 2^{Virtual\ page\ number\ length} = 2^{20}$$
$$= 1{,}048{,}576\ entries$$

Virtual Memory

- The virtual address alone does not indicate where the page is on a hard disk.
  - The location of each page on disk in the virtual address space must be kept track of.

- The operating system creates *swap space* on the disk for all the pages of a process when the process is created.
  - Also creates records of where each virtual page is stored on disk.

Virtual Memory



- If the valid bit is off, the page is on the hard disk
- The table of physical page addresses and disk page addresses, while logically one table, is stored in two separate data structures.
    - Dual tables are justified in part because we must keep the disk addresses of all the pages, even if they are currently in main memory

Rotational Hard Disks

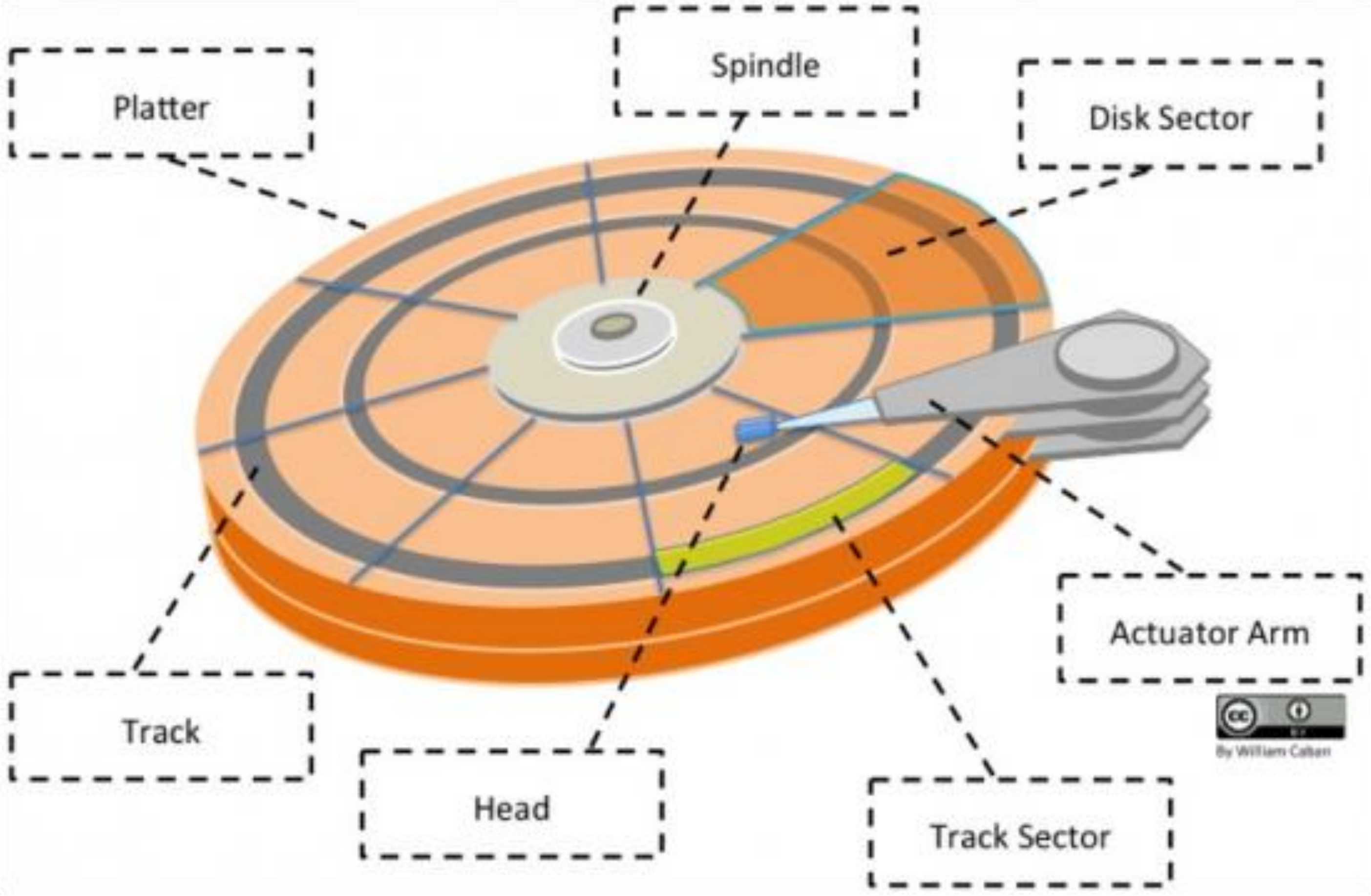
- Rotational hard disks consist of metal *platters* that rotate on a *spindle* at 5400 to 15,000 revolutions per minute.
  - The platters are covered with magnetic recording material on both sides.

- An *actuator arm* contains a small electromagnetic coil called a *head* is located just above the surface of each platter.
  - The actuator arm moves back and forth across the disk
  - The head reads and writes data to the disk.

Rotational Hard Disks

- The surface of a platter is divided into concentric circles called *tracks*.
- Platters are further divided into *sectors*, typically 4096 bytes in size.
  - A track sector is the smallest unit of storage that can be allocated by the disk.
  - For example, a file that is only 100 bytes will still take up one whole track sector; A 5000-byte file will need to use 2 track sectors.

Rotational Hard Disks



Rotational Hard Disks

- Hard disk performance is measured in a few different ways.

- Minimum seek time: The fastest the actuator can move a certain track
  - *Seeking* is the process of the actuator arm moving to a certain track
- Maximum seek time: The slowest the actuator can move a certain track
- Average seek time: Sum of all possible seek times divided by number of possible seeks

Rotational Hard Disks

- One the actuator arm's head reaches the right track, it must wait for the platter to spin to the correct sector
  - Call the *rotational latency*

- Average rotational latency is one half of a rotation divided by the rotations per minute.
- For a disk with 5400 RPMs:

$$Average\ rotational\ latency = \frac{0.5\ rotations}{5400\ RPMs}$$

$$= \frac{0.5\ rotations}{5400\ RPMs/60\ \frac{seconds}{minute}} = 0.0056\ seconds = 5.6\ ms$$