37. Hard Disk Drives

Operating System: Three Easy Pieces

Hard Disk Driver

- Hard disk driver have been **the main form of persistent data storage** in computer systems for decades (… and consequently file system technology is predicated on their characteristics).
  - The drive consists of a large number of **sectors** (512-byte blocks).
  - **Address Space :**
    - We can view the disk with `n` sectors as an array of sectors; `0` to `n-1`.

The Interface

- The only guarantee is that a single 512-byte write is **atomic**
  - Either complete all or nothing

- Multi-sector operations are possible.
  - Many file systems will read or write 4KB at a time.
  - **Torn write**:
    - If an untimely power loss occurs, only a portion of a larger write may complete.

- Accessing blocks in **a contiguous chunk** is the fastest access mode.
  - A sequential read or write
  - Much faster than any more random-access pattern.

Basic Geometry (Cont.)

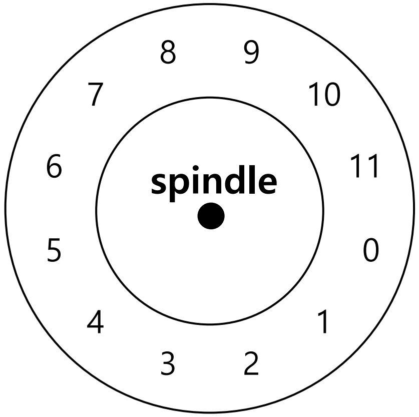
Spindle

- Spindle is connected to a motor that spins the platters around (constantly)
- The rate of rotations is measured in **RPM** (Rotations Per Minute).
  - Typical modern values : 7,200 RPM to 15,000 RPM.
  - E.g., 10000 RPM : A single rotation takes about 6 ms.

Track

- Concentric circles of sectors
- Data is encoded on each surface in a track.
- A single surface contains many thousands and thousands of tracks.

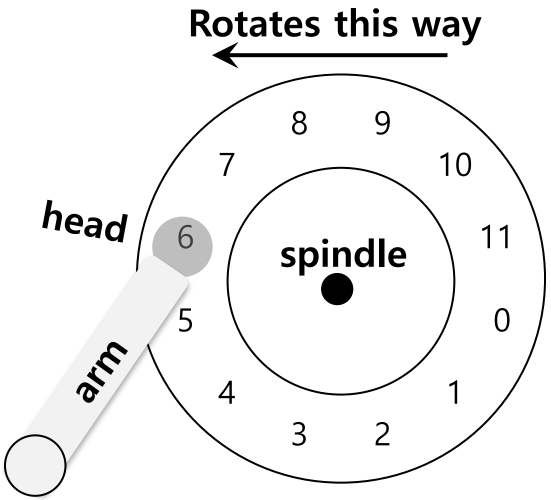
Basic Geometry



A Disk with Just A Single Track (12 sectors)

- **Platter** (Aluminum coated with a thin magnetic layer)
  - A circular hard surface
  - Data is stored persistently by inducing magnetic changes to it.
  - Each platter has 2 sides, each of which is called a **surface**.

A Simple Disk Drive



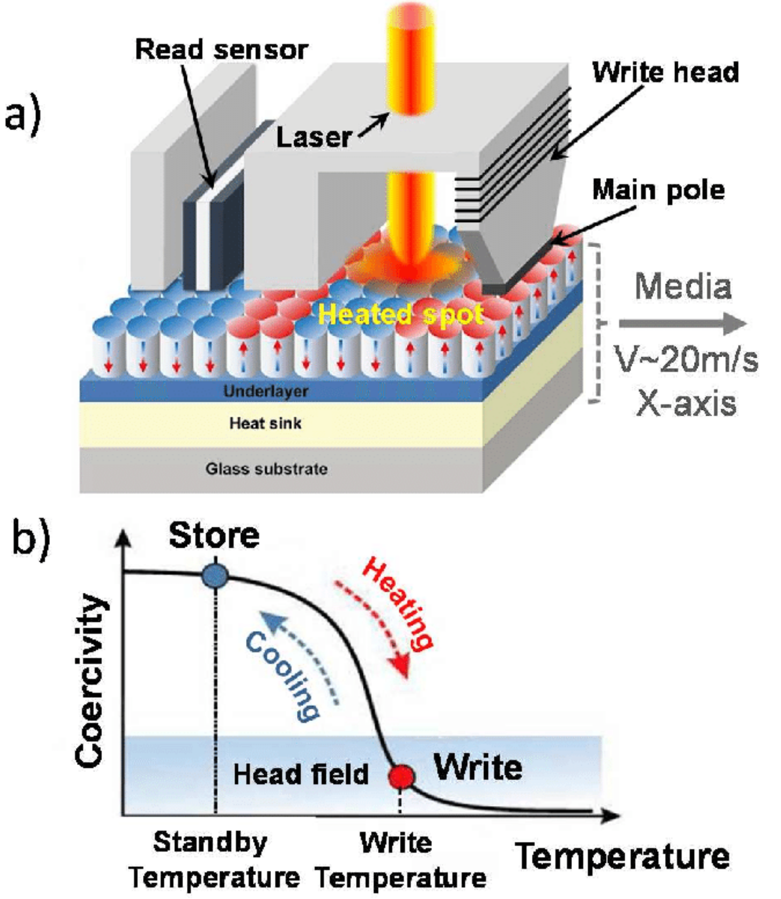
A Single Track Plus A Head

- **Disk head** (One head per surface of the drive)
  - The process of *reading* and *writing* is accomplished by the **disk head**.
  - Attached to a single disk arm, which moves across the surface.

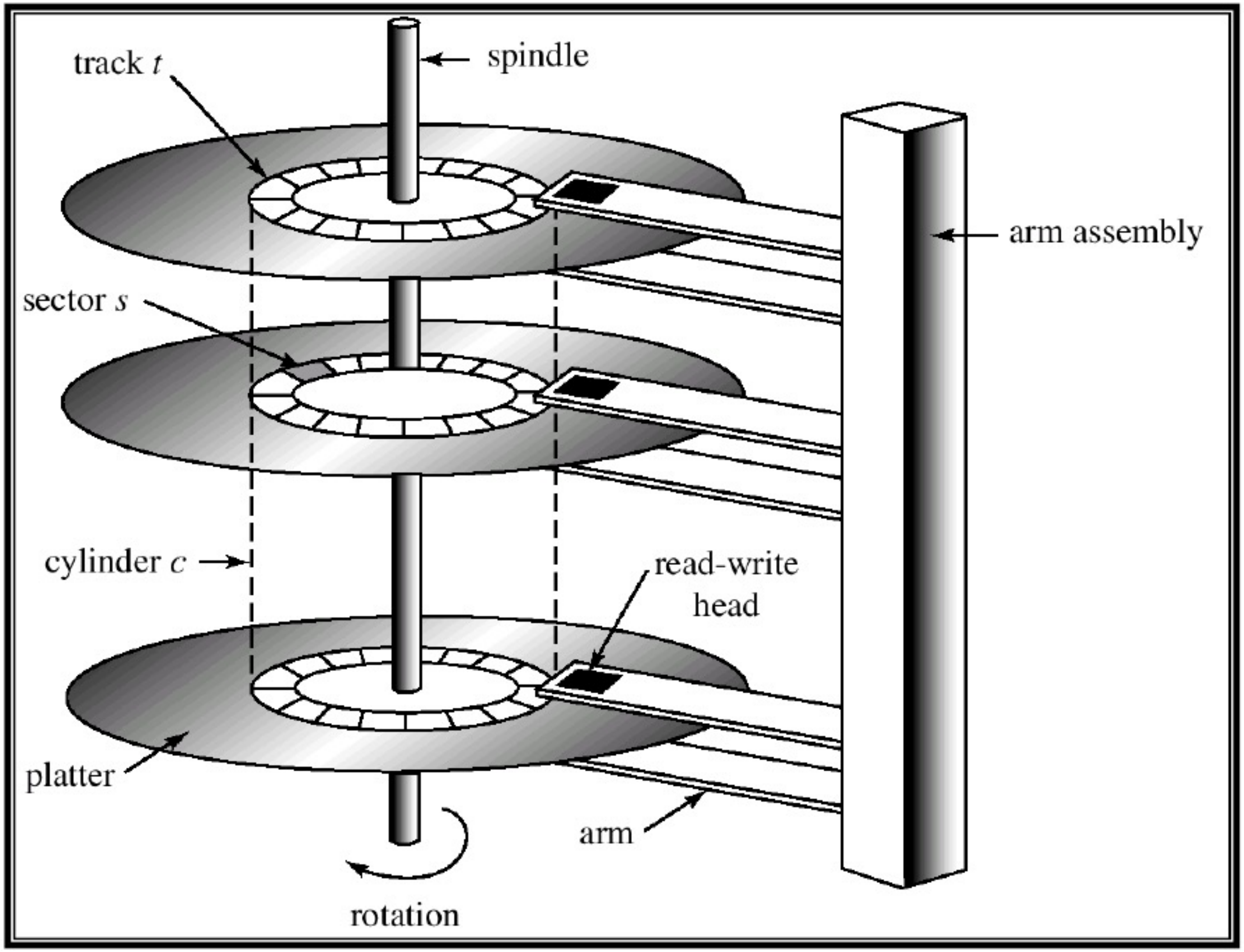
- Head is a “magnet”: uses Giant Magneto Resistance (GMR) and Tunneling magnetoresistive (TMR) [quantum effects] to read and write

Aside: Heat Assisted Magnetic Recording (HAMR)

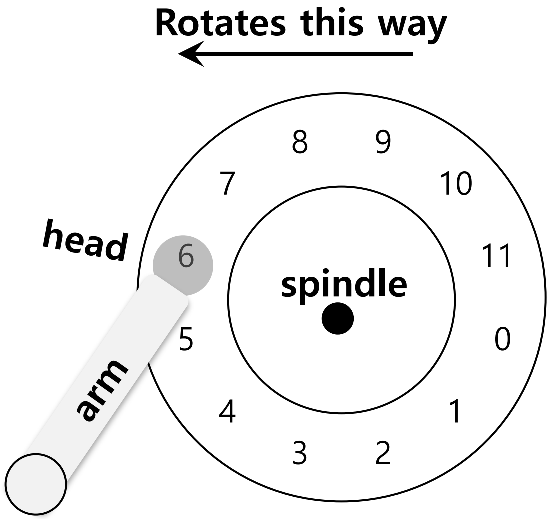
- Now disks 2.6 Tb/in^2 (+8Tb/in^2 in 2025), now 20TB (+100TB disks in 2025)



Example of a Disk



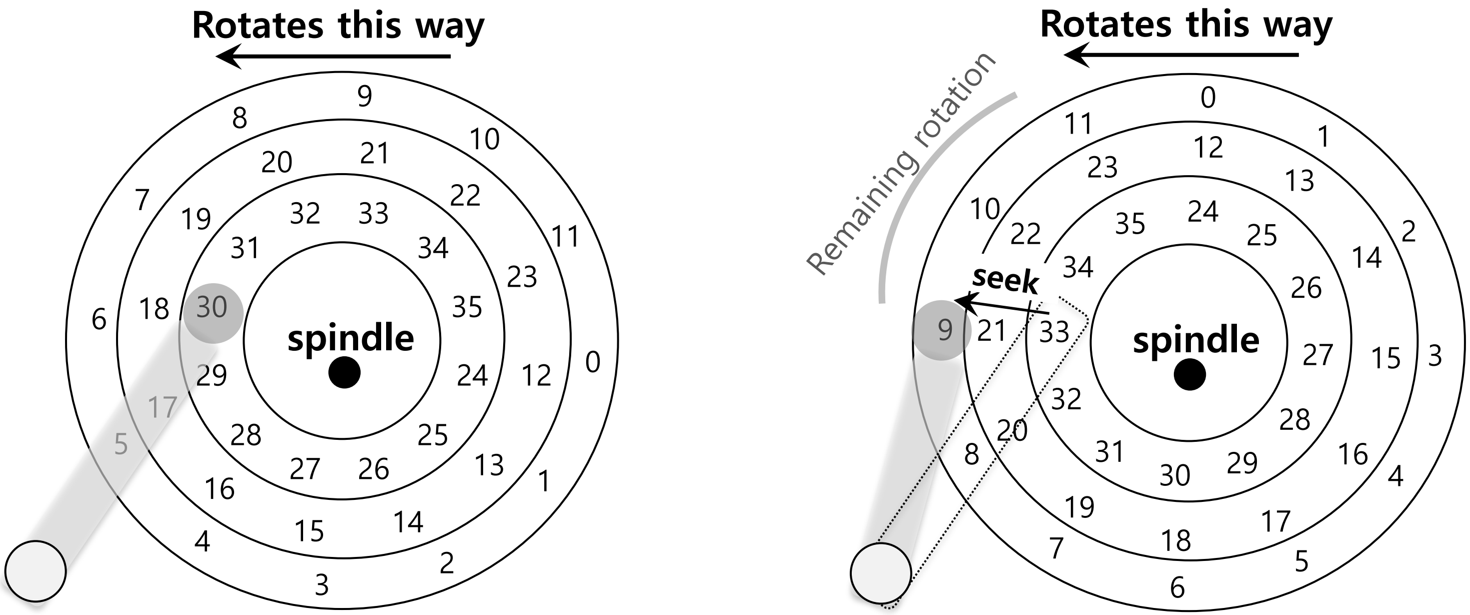
Single-track Latency: The Rotational Delay



A Single Track Plus A Head

- **Rotational delay:** Time for the desired sector to rotate
  - Ex) Full rotational delay is `R` and we start at `sector 6`
    - Read sector 0: Rotational delay = $\frac{R}{2}$
    - Read sector 5: Rotational delay = `R-1` (worst case.)

Multiple Tracks: Seek Time



Three Tracks Plus A Head (Right: With Seek)
(e.g., read to sector 11)

- **Seek**: Move the disk arm to the correct track
  - Seek time: Time to move head to the track contain the desired sector.
  - One of the **costliest** disk operations.

Phases of Seek

- Acceleration → Coasting → Deceleration → Settling
  - **Acceleration**: The disk arm gets moving.
  - **Coasting**: The arm is moving at full speed.
  - **Deceleration**: The arm slows down.
  - **Settling**: The head is *carefully positioned* over the correct track.
    - The settling time is often quite significant, e.g., 0.5 to 2ms.

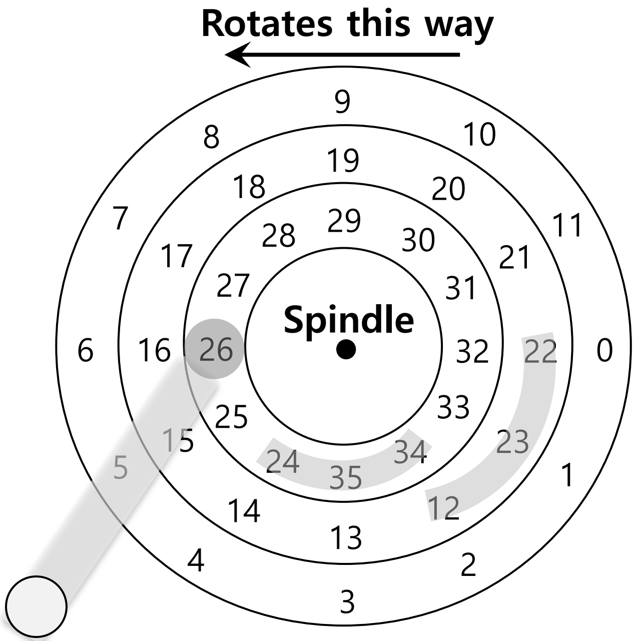
Transfer

- The final phase of I/O
  - Data is either *read from* or *written to* the surface.

- Complete I/O time:
  - **Seek**
  - Waiting for the **rotational delay**
  - **Transfer**

Track Skew

- Make sure that sequential reads can be properly serviced **even when crossing track boundaries**.



Three Tracks: Track Skew Of 2

- *Without track skew*, the head would be moved to the next track, but the desired next block would have already rotated under the head.

Cache (Track Buffer)

- **Hold data** read from or written to the disk
  - Allow the drive to quickly respond to requests (to recently seen sectors).
  - Small amount of memory (usually around 128(1TB)-512 MB (20TB))

Write on cache (a.k.a store buffer)

- **Writeback** (Immediate reporting)
  - Acknowledge a write has completed when it has **put the data in its memory**.
  - faster but dangerous

- **Write through**
  - Acknowledge a write has completed after the write has **have been written to disk**.

- Like in memory hierarchy (with store buffer), this can be source of **headaches**
  - **Consistency** issues (order is not guaranteed)

I/O Time: Doing The Math

- I/O time ($T_{I/O}$): $T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$

- The rate of I/O ($R_{I/O}$): $R_{I/O} = \frac{Size_{Transfer}}{T_{I/O}}$

	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects Via	SCSI	SATA

Disk Drive Specs: SCSI Versus SATA

ASIDE: The "Average" disk-seek

- Usually referred as a third of full seek time

- Average of all seek distances (for a disk with N tracks is), between two tracks x,y

$$\frac{1}{N^2}\sum_{x=0}^{N}\sum_{y=0}^{N}|x-y| \cong$$

$$\frac{1}{N^2}\int_{x=0}^{N}\int_{y=0}^{N}|x-y|dxdy = \frac{1}{N^2}\left(\frac{1}{3}x^3 - \frac{N}{2}x^2 + \frac{N^2}{2}x\right)|_0^N = \frac{N}{3}$$

Performannce

- $T_{rotation}$
  - 15000RPM -> T_{1rot}= 1/250RPS = 4ms AverageT_{1rot}= T_{1rot}/2 = 2ms
  - 75000RPM → AverageT_{1rot} = 4 ms

- Sequential access
  - 1 rotation and no extra time in rotation or seeking
    - 15K RMP → 2ms+**4ms**+100MB/125MB/s = 800 ms
    - $R_{I/O}$ = 100MB/800ms = 125 MB/s

- Random access
    - 15K RPM → 2ms+**4ms** + 4K/125 = 30 ms
    - $R_{I/O}$ = 4KB/30ms = 0.66 MB/s

I/O Time Example (doing the math)

- **Random workload**: Issue 4KB read to random locations on the disk
- **Sequential workload**: Read 100MB consecutively from the disk

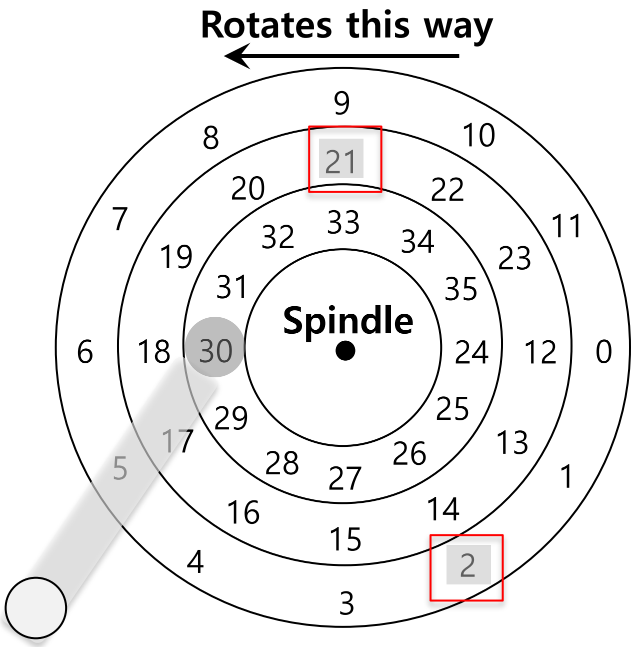
		Cheetah 15K.5	**Barracuda**
T_{seek}		4 ms	9 ms
$T_{rotation}$		2 ms	4.2 ms
Random	$T_{transfer}$	30 microsecs	38 microsecs
	$T_{I/O}$	6 ms	13.2 ms
	$R_{I/O}$	0.66 MB/s	0.31 MB/s
Sequential	$T_{transfer}$	800 ms	950 ms
	$T_{I/O}$	806 ms	963.2 ms
	$R_{I/O}$	125 MB/s	105 MB/s

Disk Drive Performance: SCSI Versus SATA

There is a huge gap in drive performance between random and sequential workloads

Disk Scheduling

- **Disk Scheduler** decides which I/O request to schedule next.
- **SSTF** (Shortest Seek Time First)
  - Order the queue of I/O request by track
  - Pick requests on the nearest track to complete first



SSTF: Scheduling Request 21 and 2

Issue the request to 21 → issue the request to 2

SSTF is not a panacea.

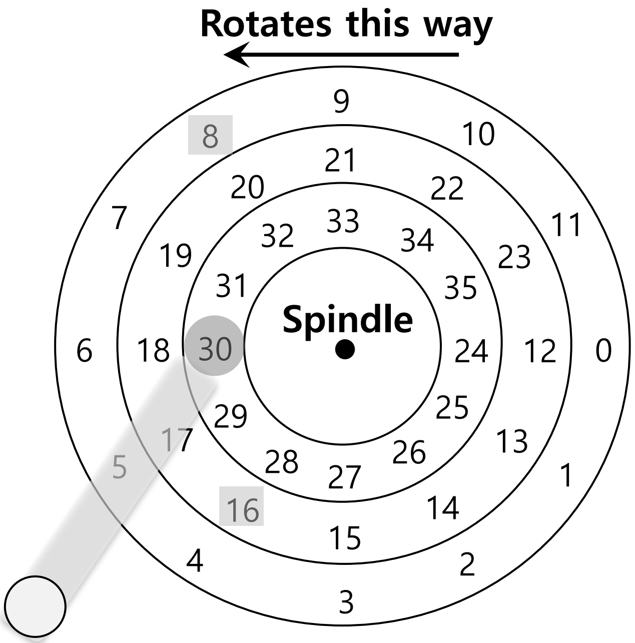
- **Problem 1**: The drive geometry is not available to the host OS
  - Solution: OS can simply implement Nearest-block-first (NBF). Assume that sector “address” means ”something”

- **Problem 2**: Starvation
  - If there were a steady stream of request to the inner track, request to other tracks would then be ignored completely.

Elevator (a.k.a. SCAN or C-SCAN)

- Move across the disk servicing requests in order across the tracks.
  - **Sweep**: A single pass across the disk
    - If a request comes for a block on a track that has already been services on this sweep of the disk, it is queued until the next sweep (in both sweeps)
  - **F-SCAN**
    - Freeze the queue to be serviced when it is doing a sweep
    - Avoid starvation of far-away requests
  - **C-SCAN** (Circular SCAN)
    - Sweep only from outer-to-inner, or inner-to-outer, etc. (better fairness since middle tracks are no longer favored)
- All of then **ignore rotation** (not real SJF)

How to account for Disk rotation costs?



SSTF: Sometimes Not Good Enough

- If rotation is faster than seek: request 16 → request 8
- If seek is faster than rotation : request 8 → request 16

On modern drives, both seek, and rotation are roughly equivalent:
Thus, SPTF (Shortest Positioning Time First) is useful.

Other issues

- Where is scheduling performed?
  - **Old systems**: OS
  - **Modern systems**: hardware can do a lot: accommodate multiple outstanding requests which allows accurate SPTF scheduling (the hardware knows itself),
    - Duplicate effort in OS and hardware. Hardware will do whatever is best (little effect of OS decision in order).
- **Reduce the number of request** sent to the disk and lowers overhead
  - E.g., read blocks 33, then 8, then 34:
    - The scheduler merge the request for blocks 33 and 34 *into a single two-block request*.
  - Performed at OS level
- How long wait before sent a request to the disk?
  - Wait for a bit can be better : **non-work-conserving** (anticipatory scheduling) see Linux

- This lecture slide set has been used in AOS course at University of Cantabria by V.Puente. Was initially developed for Operating System course in Computer Science Dept. at Hanyang University. This lecture slide set is for OSTEP book written by Remzi and Andrea Arpaci-Dusseau (at University of Wisconsin)