Computer Hardware Review
(Memory Hierarchy)

Chapter 1.4
Learning Outcomes

• Understand the concepts of memory hierarchy and caching, and how they affect performance.
Operating Systems

- Exploit the hardware available
- Provide a set of high-level services that represent or are implemented by the hardware.
- Manages the hardware reliably and efficiently
- *Understanding operating systems requires a basic understanding of the underlying hardware*
Memory Hierarchy

• Going down the hierarchy
  • Decreasing cost per bit
  • Increasing capacity
  • Increasing access time

• Decreasing frequency of access to the memory by the processor
  • Hopefully
  • Principle of locality!!!!!
Caching as a general technique

• Given two-levels of data storage: small and fast, versus large and slow,
• Can speed access to slower storage by using intermediate-speed storage as a cache.
A hardware approach to improving system performance?

CPU Registers
Fast

Cache Memory (SRAM)
Fast

Main Memory (DRAM)
Slow
CPU Cache

- CPU cache is fast memory placed between the CPU and main memory
  - 1 to a few cycles access time compared to RAM access time of tens – hundreds of cycles
- Holds recently used data or instructions to save memory accesses.
- Matches slow RAM access time to CPU speed if high hit rate
- Is hardware maintained and (mostly) transparent to software
- Sizes range from few kB to tens of MB.
- Usually a hierarchy of caches (2–5 levels), on- and off-chip.
Performance

• What is the effective access time of memory subsystem?

• Answer: It depends on the hit rate in the first level.
Effective Access Time

\[ T_{\text{eff}} = H \times T_1 + (1 - H) \times T_2 \]

- \(T_1\) = access time of memory 1
- \(T_2\) = access time of memory 2
- \(H\) = hit rate in memory 1
- \(T_{\text{eff}}\) = effective access time of system
Example

- Cache memory access time 1ns
- Main memory access time 10ns
- Hit rate of 95%

\[ T_{eff} = 0.95 \times 10^{-9} + (1 - 0.95) \times (10^{-9} + 10 \times 10^{-9}) \]
\[ = 1.5 \times 10^{-9} \]
Moving-Head Disk Mechanism

- track $t$
- spindle
- sector $s$
- cylinder $c$
- platter
- read-write head
- arm
- arm assembly
- rotation
Example Disk Access Times

• Disk can read/write data relatively fast
  • 15,000 rpm drive - 80 MB/sec
  • 1 KB block is read in 12 microseconds

• Access time dominated by time to locate the head over data
  • Rotational latency
    • Half one rotation is 2 milliseconds
  • Seek time
    • Full inside to outside is 8 milliseconds
    • Track to track .5 milliseconds

• 2 milliseconds is 164KB in “lost bandwidth”
A OS approach to improving system performance?

- CPU Registers
  - Fast

- Main Memory (DRAM)
  - Fast

- Hard disk
  - Slow…
A Strategy: Avoid Waiting for Disk Access

• Keep a subset of the disk’s data in main memory
⇒ OS uses main memory as a cache of disk contents
Application approach to improving system performance

Web browser
Fast

Hard disk
Fast

Internet
Slow…
A Strategy: Avoid Waiting for Internet Access
• Keep a subset of the Internet’s data on disk
⇒ Application uses disk as a *cache* of the Internet