|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 🔑 **Essential** 🔑 | | | |  | 🏆 **Enhanced** 🏆 | | | |
|  |  |  |  |  |  |  |  |  |
| \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |  | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ | \_\_\_\_\_ |

**Score: \_\_\_\_\_**

**12 – Memory Hierarchy & Processor Cache**

**Activities**

COMP256 – Computing Abstractions

Dickinson College

Spring 2022

Prof. Grant Braught

**Name:**

Today’s class began talking about performance improvements to the stored program architecture. In particular, today focused on the different types of memory that are used in modern machines and how and why they are arranged into a hierarchy. In addition, the concept of caching and how it applies to the processor cache were introduced. In the following activities you will apply some of the ideas presented, fill in some additional details about the size, speed and cost of elements of the memory hierarchy. You will also gain some additional insight into why the processor cache works as well as how big of a performance improvement it actually provides.

They are not required viewing, but if you would like some additional perspective on today’s topic or some more information as you work through the exercises the following are good resources:

* Matthew Watkins’ video *Memory Hierarchy Introduction* gives an introduction to the memory hierarchy including discussions of cost/performance tradeoffs and locality of reference:
  + <https://www.youtube.com/watch?v=_kZY4orPQW0> (10:08)
* Carrie Ann discusses memory and processor cache in the first 5:35 seconds of the Advanced CPU Designs video from the Crash Course Computer Science series:
  + <https://www.youtube.com/watch?v=rtAlC5J1U40> (5:35)

**Memory Units:**

🔑 1. In class we saw that memory and storage sizes are sometimes specified using powers of 10 (e.g. Hard Disk Space) and sometimes using powers of 2 (e.g. Main memory and Cache). These two different values are approximately the same, but not exactly the same. Complete the table below with the values for these units using both powers of 10 and powers of 2. Use the first row as an example.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | **Memory Size** | **As a power of 10** | **As a whole number** | **As a power of 2** | **As a whole number** |  |
|  | **Kilobyte (KB)** | 103 | 1,000 | 210 | 1,024 |  |
|  | **Megabyte (MB)** |  |  |  |  |  |
|  | **Gigabyte (GB)** |  |  |  |  |  |
|  | **Terabyte (TB)** |  |  |  |  |  |
|  |  |  |  |  |  |  |

It is worth noting that because of the differences above, you will sometimes see abbreviations such as MiB (mebibyte) or GiB (gibibyte) to specifically refer to the value that is a power of 2.

🔑 2. It is helpful to have an idea of the relative size of the different measurements of memory size. Complete the table below to get a feel for these sizes. **Write your answers out both as a whole number and as a power of 2**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  | **As a power of 2** | **As a whole number** |  |
|  | How many KB in a MB? |  |  |  |
|  | How many MB in a GB? |  |  |  |
|  | How many GB in a TB? |  |  |  |
|  | How many KB in a TB? |  |  |  |
|  |  |  |  |  |

**The Memory Hierarchy:**

🔑 3. Tradeoffs between capacity, cost and speed affect the amount of each type of memory or storage that a computer system will have at each level of the memory hierarchy. Complete each of the sections and deal with these tradeoffs.

a. Fill in each of the blanks below with “are faster than” or “are slower than” to compare the typical speeds of the different types of memory/storage.

i. Registers Main Memories

ii. Caches Registers

iii. Hard Disk Drives Main Memories

iv. Solid State Drives Hard Disk Drives

b. Repeat part a, but this time fill in the blank with “costs more than” or “costs less than” to compare the typical cost per bit of the different types of memory/storage.

i. Registers Main Memories

ii. Caches Registers

iii. Hard Disk Drives Main Memories

iv. Solid State Drives Hard Disk Drives

c. Repeat part a, but this time fill in the blank with “are larger than” or “are smaller than” to compare the typical capacities of the different types of memory/storage.

i. Registers Main Memory

ii. Cache Registers

iii. Hard Disk Drive Main Memory

iv. Solid State Drive Hard Disk Drive

🔑 4. You may have noticed the terms SRAM and DRAM in the memory hierarchy diagrams in the class slides. SRAM and DRAM are two different ways to construct memory using electronic components. SRAM stands for Static RAM and DRAM stands for Dynamic RAM. While you will learn more about each of these and their differences in lab you can infer a few things from them based on what we know and where they are used in the memory hierarchy. Complete the table below by filling in the cells in the table below with SRAM or DRAM.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Is faster |  |  |
|  | Is less expensive |  |  |
|  | Is used for larger memories |  |  |
|  |  |  |  |

🔑 5. We have seen that different types of memory/storage in the hierarchy can be classified along different dimensions. For example, by persistence (volatile or non-volatile), or by access type (random, direct, or sequential).

a. Classify each of the following types of memory or storage as volatile or non-volatile, whichever fits best. **Give a brief justification for each of your answers.**

i. Main Memory (RAM)

ii. A Computer Backup Tape

iii. A classroom whiteboard

iv. The notes that you take during a class

b. Classify each of the following types of memory accesses as random access, direct access or sequential access, whichever fits best. **Give a brief justification for each of your answers.**

i. Retrieving the value from a CPU Register for a computation

ii. Reading a file from a Computer Backup Tape

iii. Getting the book “Frankenstein” from a shelf in the library

iv. Finding the 2 of diamonds in a shuffled deck of cards.

🏆 v. Thinking about a memory in your mind.

**Processor Cache:**

🔑 6. In class we used a plumber and his toolbox as a metaphor to explain the concept of processor cache. In about a paragraph of your own words, develop a metaphor of your own and use it to explain processor cache. You need to clearly identify the elements of your metaphor that play the roles of the main memory, the processor cache, the registers and the ALU.

🔑 7. Use the elements of your metaphor from question #6 to briefly explain each of the following terms:

a. Cache Hit

b. Cache Miss

🔑 8. Use the elements of your metaphor from question #6 to briefly explain each of the following cache principles. If you find it easier you may use the plumber metaphor from class to complete this question.

a. Spatial Locality

b. Temporal Locality

🏆 9. The idea of cache is used widely in computing, not just for processor caches. You have probably heard of *browser cache*. Your web browser uses its browser cache, basically a file on your computer, to store copies of web pages and other information that you have retrieved from servers around the web. Briefly explain how browser cache can use the principles of spatial and temporal locality to improve your web browsing experience.

**Seeing through the Abstraction:**

Like all of the other abstractions we have studied the idea is that we get to pay attention to the important information, while forgetting about (or never knowing) the unimportant details. Here the important details are the memory accesses. We can simply think of accessing the main memory as getting a value from an address without worrying about or even knowing what the cache is doing. The hardware provides an abstraction that hides from us all of the checking of the caches, dealing with cache misses and the copying of surrounding data and instructions to the cache. But as we’ve seen before that sometimes these “unimportant” details show through. The next question explores this with respect to processor cache.

🏆 10. Examine the program at:

* <https://repl.it/@braughtg/2dArrayCache#Main.java>

a. This program declares a 2d array. How many rows and columns are in the array?

b. This program has two functions downEachColumn and acrossEachRow. These functions both have the same effect. They just do it in two slightly different ways. What effect does calling either one of these methods have on the contents of the array?

c. Run the program a few times. Which function accomplishes this operation more quickly? Approximately how many times faster is it than the other?

Surprised? I know I was. Now… of course the question is why? To understand that, recall that Java stores 2d arrays as a 1d array of 1d arrays as shown here:

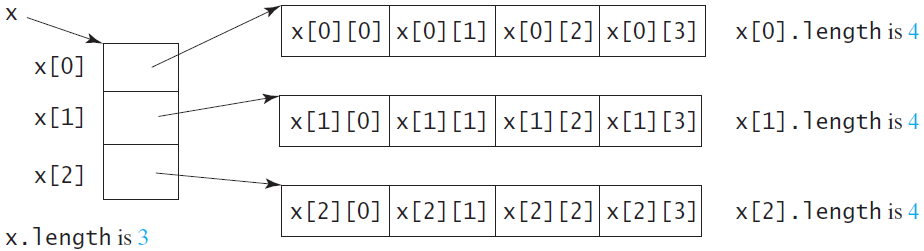


Image from <https://www.therevisionist.org/software-engineering/java/tutorials/2-dimensional-array/>

Note that arrays are stored in contiguous memory locations, that is if x[0] is stored at address 1000 then x[1] would be stored at address 1001. Similarly, if x[1][0] is stored at address 2000, then x[1][1] would be stored at address 2001.

🏆 11. Now, thinking about how arrays are stored and how cache works explain based on the principles of locality why the function you identified in #10c is faster than the other one even though they have the exact same effect.

Optional: To help me improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.