CST 334: Operating Systems

Dr. Glenn Bruns

# Caching

**Purpose**. Caching is one of the most important ideas to improve the performance of computing systems. The purpose of this assignment is to help you understand caching by working with a program that simulates operation of a cache.

**Instructions**. Copy the following tar file (on mlc104) to a directory of your own and untar it.

/home/CLASSES/brunsglenn/cst334/hw/hw7/cache.tar

You will see two C files: cache\_sim.c and random\_cache.c, and an include file random\_cache.h. Program cache\_sim.c models the use of a cached data store. Program random\_cache.c implements a cache with a random “cache replacement policy”. Read the program carefully and understand how it works. The Makefile shows how to compile it. Then answer the following questions by downloading and editing [cache.txt](https://drive.google.com/file/d/1--RzPnjk1NhokIpKJGkDUa-QbUdYpto5/view?usp=sharing).

1. What is the hit ratio of the cache? (The hit ratio is the fraction of accesses for which there is a cache hit. It can be computed as the number of cache hits divided by the number of accesses, which is the sum of cache hits and cache misses.) You will need to add a print statement to cache\_sim.c to get this.
2. The cache size in the program is 10, and the size of the data store is 1000 (look at the constants defined near the top of cache\_sim.c). Run the program with cache sizes of 3, 5, 10, 20, and 50. What is the hit ratio in each case? (Note: to run these tests you can either make cache size a command-line argument, or simply edit the CACHE\_SIZE value and recompile.) Reset the cache size to 10 after these tests.
3. Look at the main program. It makes a bunch of memory accesses, always accessing the value at address j. After every access, it modifies j. It does this by randomly selecting a value out of array move\_size1, and then adding that value to the current value of j. Within the loop over i in the main program of cache\_sim.c, modify move\_size1 to move\_size2, and rerun the program. What is the new hit ratio value?
4. Suppose that in 10,000 accesses, you get 6500 cache hits, and 3500 cache misses. Suppose also that time to perform an access is 0.1 ms if a cache hit, but 10 ms if a cache miss. What is the average time for an access? (Give your answer in ms.)
5. Repeat the same problem, but assume a poorer caching algorithm is used, and you have 4100 cache hits, and 5900 cache misses. What is the average time for an access in this case? (Give your answer in ms.)

**Submission**: Submit your edited cache.txt on iLearn.

**Grading**: Each problem is worth 10 points.