

Contents

Lab 4: Cache Geometries	1
Overview	1
Instructions	1
Your Tasks	2
Submitting Your Work	2

Lab 4: Cache Geometries

Overview

Chip D. Signer, Ph.D, is trying to reverse engineer a competitor’s microprocessors to discover their cache geometries and has recruited you to help. Instead of running programs on these processors and inferring the cache layout from timing results, you will approximate his work by using a simulator.

This lab should be done on a 64-bit machine. Use the provided VM or your own personal 64-bit computer (but at least check your solutions on the VM).

Instructions

Specifically, each of these “processors” is provided as an object file (.o file) against which you will link your code. See the file `mystery-cache.h` for documentation of the function interface that these object files export. Your job is to fill in the function stubs in `cache-test-skel.c` which, when linked with one of these cache object files, will determine and then output the cache size, associativity, and block size. Some of the provided object files are named with this information (e.g. `cache_64c_2a_16b.o` is a 64 KB capacity, 2-way set-associative cache with 16B blocks) to help you check your work. There are also 4 mystery cache object files, whose parameters you must discover on your own.

You can assume that the mystery caches have sizes that are powers of 2 and use a least recently used replacement policy. You cannot assume anything else about the cache parameters except what you can infer from the cache size. Finally, the mystery caches are all pretty realistic in their geometries, so use this fact to sanity check your results.

Perform an update to your course-materials directory on the VM by running the `update-course` command from a terminal window. On the script’s success, you should find the provided code for lab4 in your course-materials directory. As a convenience, here is an archive of the course-materials directory as of this lab assignment: lab4.tar.gz.

The provided `Makefile` includes a target `cache-test`. To use it, set `TEST_CACHE` to the object file to link against on the command line - i.e. from within the lab4 directory run the command:

```
make cache-test TEST_CACHE=cache_64c_2a_16b.o
```

This will create an executable `cache-test` that will run your cache-inference code against the supplied cache object. Run this executable like so:

```
./cache-test
```

and it will print the results to the screen.

Your Tasks

Complete the 3 functions in `cache-test-skel.c` which have `/* YOUR CODE GOES HERE */` comments in them.

Your functions should accurately determine the geometry of each of the four mystery caches as well as the caches that give away their geometries in their filenames.

Submitting Your Work

You will be able to submit your assignment with the `submit-hw` script that is bundled with the course-materials. Using the script should be straight-forward, but it does expect you to not move/rename any files from the “course-materials” directory. Open a terminal and type the following command:

```
submit-hw lab4
```

After calling the `submit-hw` script, you will be prompted for your Coursera username and then your submission password. **Your submission password is NOT the same as your regular Coursera account password!!!!** You may locate your submission password at the top of [the assignments list page](#).

After providing your credentials, the `submit-hw` script will run some basic checks on your code. For lab4, this means that it will check to ensure that your functions correctly determine the geometries of the 3 given caches. **It will not check that you determine the correct geometries of the mystery caches.** The mystery checks are left up to the auto-grader which will only tell you which characteristics you get right or wrong (and not what the correct answers should be).

Once confirming that you wish to submit, with a working Internet connection, the script should submit your code properly. You can go to [the assignments list page](#) to double-check that it has been submitted and check your score as well as any feedback the auto-grader gives you. You may also download your submission code from this page, if you wish.