# Lab08

## Lab 8

Deadline: EOD (End of Day) Friday, November 6th

## Objectives:

- The student would explore the workings of virtual memory, specifically the TLB and the Page Table.
- The student would be able to analyze TLB hit rate and Page Table hit rate and figure out what accesses optimize these values.

## Important:

Per the relief package, you just need to show up to "check-in" with your TA to receive credit for the lab. The AG and checkoff questions mentioned in the lab are optional but **highly recommended** given that they are still in scope for the final.

## Setup

Pull the lab 8 files from the lab starter repository with

Write your answers in the provided `answers.txt` file. For numerical answers, write the number instead of spelling it out (e.g. "7" instead of "seven"). The Checkoff Question lines are ignored by the autograder and there for your convenience. The autograder assumes that the original formatting will not be changed, so please don't add additional lines, switch existing lines, or otherwise modify the current formatting.

For this lab we will mostly be using the virtual memory simulation features of Camera, a cache and virtual memory simulator. You may also find the cache simulations interesting, however we won't be working with those here. Unfortunately, Camera is known to have issues when trying to run it on the Hive or Linux machines, so it's recommend to download Camera from here, and simply double click on the jar file to run it on your own (non-Linux) laptop. If you're on a Mac, you may need to go to "Security & Privacy" in your settings and click "Open Anyway" to allow Camera to run. Some displays don't seem to play nice with the standard Camera app, if the values in memory are all squished together, try running this version of Camera. If you are unable to find a way to get Camera working on a machine, please partner up with someone who does.

Once Camera opens up, select the virtual memory option to open a visualization of the virtual memory system. In the top left you can see the contents of physical memory. Just below that is a listing of all the pages of virtual memory for this process. To the right of these items are the contents of the TLB and the Page Table. At this point these should all be empty as we haven't done anything yet. Read about the statistics of your memory system in the "PROGRESS UPDATE" box at the bottom of the window. This area will keep you updated on your status through the simulation as it progresses. You can move the simulation forward, backward or start it over from the beginning using the buttons to the right of the "PROGRESS UPDATE" box.

## Exercise 0 - Sanity Check

Before you continue, **MAKE SURE THAT YOU OPENED THE VM SIMULATOR AND NOT THE CACHE SIMULATOR**.

## Exercise 1 - Working with CAMERA

Click the button labeled "Auto Generate Add. Ref. Str." at the right-hand side of the window. This will generate a set of ten address references. You can think of these as a series of RISC-V "load word" instructions reading from the memory address specified. Click the button labeled "Next" to begin the simulation.

For the rest of this exercise you are at the mercy of the "PROGRESS UPDATE" box. After each click of the "Next" button examine the contents of the box and the current state of the memory system. Try to really get an understanding of what is going on in the TLB, the Page Table, and Physical Memory at each step.

Once you have reached the end of the simulation note the number of TLB Hits and Misses and Page Hits and Faults. Write these numbers down, along with the sequence of memory accesses used to show to your TA during checkoff.

# Checkpoint

1. Given the way the address was broken down, how big (in words) are the pages in this model? Leave out the units in your answer (e.g. 4 instead of 4 words).
2. How many TLB Hits and Misses did we have for the randomly generated set of ten addresses? What about for Page Hits and Page Faults? Your answer should be a comma separated list (e.g. 1, 2, 3, 4).
3. Did you have any Page Hits? (Why?) Can you think of a set of ten addresses that would result in a Page Hit? Your answer should be two [yes/no]'s separated by a comma.

**Checkoff Questions**

- Explain the process by which we turn a virtual address into a physical address for the very first access (emphasizing on TLB Misses and Page Faults).
- Why does the physical address only have 2 bits for the PPN while the virtual address has 3 bits for the VPN?

# Exercise 2 - Misses

Now that you've seen what a random workload looks like in the VM system let's try creating a custom workload with a specific property. Your goal for this exercise is to create a workload of ten memory accesses that will cause ten TLB misses and ten Page Faults. You should be able to come up with such a workload on paper, but then you should run it in CAMERA to verify your work. You can specify a custom workload in CAMERA by clicking the button labeled "Self Generate Add. Ref. Str." and entering in the addresses you want to reference one at a time.

# Checkpoint

1. Write down your ten memory accesses. The answer should be formatted as a comma separated list of hex values (e.g. 00, 01, 02, 03, 04, 05, 06, 07, 08, 09).

# Exercise 3 - Fixing our Faults

Given your sequence of memory accesses from Exercise 2, can you find a change to a single parameter (e.g. TLB Size, Physical Memory Size, Virtual Memory Size, Page Size) that would result in the same number (ten) of TLB misses but result in fewer than ten page faults?

# Checkpoint

1. Write down a parameter which if changed by itself (while all other parameters stay the same) would result in ten TLB misses but **fewer** than ten page faults.

# Exercise 4 - Bringing it All Together

We used VMSIM, another Virtual Memory simulator, to create this question. You have two options for this exercise.

- Watch this webm of a VMSIM simulation.
- Use the `appletviewer` command in your Terminal like so (doesn't work on Hive):

```
$ appletviewer https://denninginstitute.com/workbenches/vmsim/vm.html
```

Observe what is happnening and answer the following questions:

What is different about the setup of the system in this question as compared to the setup in CAMERA? In particular, what are P1, P2, P3, and P4? If you watch closely you'll see that this simulation reports a much higher percentage of TLB misses than random runs on CAMERA did. Why might this be? (If you have trouble following the simulation, use the `appletviewer` and turn down the speed using the slider on the bottom right.)

**Checkoff Question**

- Explain why there is a much higher percentage of TLB misses in this simulation

---

# Checkoffs

Please submit to the **Lab Autograder** assignment.

Note that the autograder is whitespace and case insensitive, but otherwise very simple and thus incapable of recognizing typos or misformatted answers.

The autograder looks for the `answers.txt` file.

During checkoffs, be prepared to go over the following questions with your TA:

Exercise 1

- Explain the process by which we turn a virtual address into a physical address for the very first access (emphasizing on TLB Misses and Page Faults).
- Why does the physical address only have 2 bits for the PPN while the virtual address has 3 bits for the VPN?

Exercise 4

- Explain why there is a much higher percentage of TLB misses in this simulation