Queen's School of Computing
CISC324 – Fall 2024
Instructor: Dr. Anwar Hossain
Lab 4 - Due Date: Nov. 22, 2024

Two-Level Paging and Translation Lookaside Buffer (TLB) Simulation

This lab is designed to introduce students to memory management concepts such as paging and the use of the Translation Lookaside Buffer (TLB) to improve address translation efficiency. Students will implement a two-level paging system and simulate a TLB cache to reduce memory access latency for frequently accessed pages.

In this lab, you will demonstrate the following:

1. **Two-Level Paging**: Using a two-level page table to manage virtual addresses, breaking them down into Level 1 and Level 2-page indices and an offset.
2. **TLB Caching**: Implementing a TLB with an LRU eviction policy to speed up address translation for frequently accessed pages.
3. **Performance Measurement**: Evaluating TLB hit rates across different access patterns (sequential, random, and repeated small subset).

Answer Questions

Question 1: After completing the code, run the lab and report the outcome of all the three test cases (screenshot or copy-paste the outcome) in a document lab4<last 4-digit student ID>.pdf.

Question 2: Explain why certain test cases have higher TLB hit rates than others.
Question 3: Describe the impact of the LRU eviction policy on the TLB hit rate.
Question 4: If the TLB size were doubled, how would this impact TLB hit rates in each access pattern? Justify your answer with observations from your results.

Prepare your environment!

The boilerplate code, provided to you, can work on any environment (Linux, Windows, MacOS, etc.) so please feel free to search for how to install Python on your machine.

IDEs that you can use (feel free to use any other IDE)

1. Visual Studio Community Edition for Windows-based machines only (it's something different from VS Code).
2. Visual Studio Code (Can work on any operating system).

Queen's School of Computing
CISC324 – Fall 2024
Instructor: Dr. Anwar Hossain
Lab 4 - Due Date: Nov. 22, 2024

3. PyCharm (Can work on any operating system).

Install Python on Your Machine

There are too many methods to install Python (feel free to use any other method):

1. Download and install Anaconda (which works on any operating system).
2. Download and install Python from the official website (works on any operating system).

Download the Boilerplate Code

There are two methods to download the boilerplate code:

1. Download the whole repository of the course (Zip file) and open the folder named "Lab4".
2. If you are familiar with Git and Git commands, you can clone the repository or you can update your local repository if you already have it cloned previously.

What to Submit

1. **Source Code**
    - Place all your completed source code files in a folder named in the following format: 324-<last 4-digit student ID>-Lab4.
    - Example: For a student with ID 20196072, the folder name should be 324-6072-Lab4.
2. **Report**
    - Create a report named lab4<last 4-digit student ID>.pdf containing the answers to the lab questions.
3. **Submission**
    - Compress the folder into a .zip or .rar file.
    - Log into OnQ, navigate to the Lab 4 dropbox, and upload the compressed file.