Cole Turner
Kenneth Choi
Logan Schwarz

Project 3 Modeling Report

In this simulator, the main components are the TLB, Page Table, Frames, Physical Memory, the Backing Store, the Reference Sequences, and the three PRA's (FIFO, LRU, OPT).

The TLB is modeled as a class with a fixed buffer of 16 indexes. This buffer stores the page number and the matching physical frame number as tuples. The TLB class has a function that adds to the buffer, it makes sure the page number is between 0-255 and if the buffer is full, it will pop off the top index and append a new tuple to the bottom to follow the TLB's FIFO replacement algorithm. The TLB class also has a function to return the frame matched to a specific page number if it is in the TLB.

The Page Table class is modeled as a class with a fixed table of 256 pages. This table stores a tuple with the physical frame number in its respective page, and a loaded bit which is set if the page is loaded in physical memory. This class has an add function that will first make sure the page is from 0-255 and will then add the frame number and loaded bit tuple to that passed in page number. The Page Table class has another function that will get the respective frame number from a passed in page number if it is there. This class also has a function to update the loaded bit associated with a page if it is in the Page Table.

The Frame class stores a boolean "is_occupied" that will default to false if there is nothing in that frame, and a "page_content" that holds the contents from backing store, or nothing if the frame is not occupied.

The Physical Memory class is modeled as a class with a fixed list of however many frames were specified on the command line, or 256 if none were specified. This class has a load function that will enumerate through the list of frames and find an unoccupied frame. If it can find an unoccupied frame it will occupy it and fill its contents, otherwise it returns none. Physical memory also has a function that checks if the physical memory is full and returns true if so and false if not.

The reference sequence is read in and parsed into a list of ints that describe the logical addresses to be translated. The backing store is read in as binary and parsed. Next we have the PRA's, where FIFO and LRU are modeled as queues and OPT is modeled as a function that checks the remaining addresses with the current addresses in memory, and replaces the furthest used address.

We modeled the actual execution of this simulator in the main function. It starts by parsing command-line arguments to determine the number of frames and the PRA to be used, with default values set to 256 frames and the FIFO algorithm if the frames and/or PRA are not specified. The function then reads a reference sequence from a file and a binary file representing the backing store. It initializes the TLB, page table, and physical memory, and then iterates through each address in the reference sequence, translating it to a physical address using the TLB and page table. The function handles page faults and TLB misses according to the specified PRA and updates the TLB and page table accordingly. Lastly, it prints out the specified statistics from the project description.