/\*test machine: csel-kh1250-04.cselabs.umn.edu

\* date: 11/07/21

\* name: Michael Allrich, Lee Knutson, Ke Wang

\* x500: allri004, knut0577, wan00802

\*/

-purpose of the program

Implement an algorithm that translates virtual addresses to physical addresses on a two-page table structure. Simulate a TLB (Translation Lookaside Buffer) that stores the 8 most recently used pages in a FIFO format.

-how to compile

Navigate to a working directory within a terminal, copy file contents into the directory. Then execute the makefile with `make`. Then execute the program with `./vmemory`. Then the output of the program should be displayed onto the terminal as well as in the tlb\_out.txt file in the same directory. If this file is deleted it will return after the next time the program is run.

-what the program does

Given an input file of virtual memory addresses, the program will translate them into indices and an offset to index into a page table to acquire the physical memory frame address, then concatenate the offset onto it. This physical address is then printed, or a -1 is printed given a page fault. Before an address is translated a TLB is checked to see if the address is already in it. If it is then the physical frame address is taken from the TLB instead of looking into the page table. The contents are printed into tlb\_out.txt before and after.

Initialize\_vmanager: Initializes the page table using functions in page\_table.o.

Translate\_virtual\_address: Taking in an unsigned int as a parameter to account for the entire range of 32 bits. And then printing the physical address in the hexadecimal format.

Print\_physical\_address: Prints a formatted version of the physical address

Get\_tlb\_entry: Searches the TLB for a virtual address matching a given value. If the given value matches any TLB entry then the corresponding physical frame address is returned, otherwise -1 is returned.

Populate\_tlb: Loops through the TLB to check if any entry is empty. If so, add in the new values at that entry. If there are no empty slots, shift all current entries forward one and add the new entry to the end of the array (FIFO).

Get\_hit\_ratio: Returns the total number of hits divided by the total calls to the TLB.

Print\_tlb: Prints the contents of the TLB into a tlb\_out.txt file. If the output file already has contents, the new additions are appended to the end.

-assumptions

1. 32-bit physical and virtual addresses
2. Using a 2-level page table
3. The 10 most significant bits (MSB) of a 32-bit virtual address are used to index into the first-level page table, which has exactly 1024 page table entries (PTEs). Each entry contains a pointer to the base of a second-level page table.
4. The next 10 bits of the virtual address are used to index into a second-level page table, which again has 1024 PTEs. Each PTE contains a physical frame number.
5. The 12 least significant bits (LSB) determine the offset within a given page/frame, with possible addresses explicitly ranging from 0x000 to 0xfff.

-contributions

Michael Allrich:

Ke Wang:

Lee Knutson:

All: Lots of debugging. Unlike previous projects, the work was not exactly split up cleanly in that we all collaborated on all parts of the program. All functions had significant input from all group members.