CS 367 Lab #3: Address Translation

Due: Sunday, May 3 at midnight

This is to be an individual effort. No partners.

For this assignment, you are going to use C to implement a virtual to physical memory mapping using bit-level operators for the given scheme:

- Virtual address – 15 bits with a 9-bit offset. This means we are using 6 bits to hold the VPN, so the page table will have up to 2^6 entries. The TLB is a 2-way cache where bits 14-12 will be the tag and bits 11-9 are the set index.

- Physical address – 28 bits – the PPN will be 19 bits.

Note that other than the number of bits used overall, this is similar to the setup in the memory example slides I went over in class.

Both the virtual address and physical address are fewer than 32 bits in length and will fit into a standard 32 bit integer. **You must use an integer (and bit-level operators) to do your work.** Use masks and shifting to get to the parts of the address you need to use at a given time.

Input:

- name of the files containing the page table and tlb. In the code below, the 'new' specifies that files 'new.pt' and new.tlb' contain the relevant page table and tlb for this run.
- Once you've read in these two files and stored the information in format, you will query for virtual addresses (which will be given in decimal). For each virtual address provided by the user, you must **output:**
  - Virtual address in hex – if this value is illegal (too large), output this info.
  - Physical address of legal virtual addresses in hex (if available from the tlb or page table) OR 'Page fault'. If the address was available, say where you would first find this information.

User enters -1 to end the input.

An example of what a run of your program might look like is given below. The underlined text is the user input. The text in red is the *required* part in terms of your output. Some additional info (like below) is fine as long as your answer is clearly marked in the output.

```
white@Zeus:$ translate white
Enter Virtual address in decimal (-1 to exit) : 3640
virtual: 0xe38
Physical Address = 0xa266038 (170287160) - from the TLB
Enter Virtual address (-1 to exit) : 11111
virtual: 0x2b67
Physical Address = 0xe250167 (237306215) - from the page table
Enter Virtual address (-1 to exit) : 22222
virtual: 0x56ce
```

```
Page fault
Enter Virtual address (-1 to exit) : 33333
virtual: 0x8235
Illegal virtual address
Enter Virtual address (-1 to exit) : -1
```

Implementation Notes

- **You must use an integer (and bit-level operators) to do your work.** Use masks and shifting to get to the parts of the address you need to use at a given time.
- Your program must read the page table and TLB from input files. The root of the name of the input files is given on the command line. For example,

  ```
  lab3 test
  ```

  will read in two files: `test.pl` and `test.tlb`. The file structures are pretty simple and described below.

- To make it easier on you (and to grade), I'm providing a program (**`/home/cs367/translate`**) that generates these text files and also then does addressing on these files so you can check your answers. The program expects a single (one word) parameter that will determine the name of the given files. Two files are generated based on the parameter – using different names will give you different starting files.
- Page table file format – 64 lines, each with a valid bit (0 or 1) and a physical page number (in decimal). This physical page number will be no more than 19 bits. An example of this is below.

  ```
  1 325498
  1 87806
  0 275771
  0 46991
  1 332592
  1 111639
  0 259531
  0 316494
  …
  ```

- TLB table file format -- There are 8 sets with two lines per set. In the file, each set is in a single input line with valid bit, tag and page number for each line. It is generated from the page table information and should contain both valid and invalid entries. An example of this is below.

  ```
  0 1 471091 0 1 141494
  0 2 518551 1 5 320251
  1 3 266557 0 2 287306
  0 2 252644 0 0 218534
  1 3 167077 0 2 216559
  …
  ```

Submitting & Grading

Submit this assignment on blackboard as a tar file. Be sure to include all needed files or your assignment will not be graded.

Your grade will be determined as follows:

- 75 points – Correctness.
- 25 points – use of C, comments, bit-level operators…