Operating System Project 6

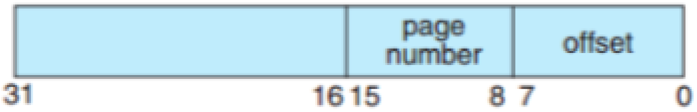
Project goal:

Designing a Virtual Memory Manager

- This project consists of writing a program that translates logical to 16 physical addresses for a virtual address space of size 2^{16} =65536 bytes.
- Your program will read from a file containing logical addresses and , using a TLB as well as a page table, will translate each logical address to its corresponding physical address and output the value of the byte stored at the translated physical address.
- The goal of this project is to simulate the steps involved in translating logical to physical addresses.

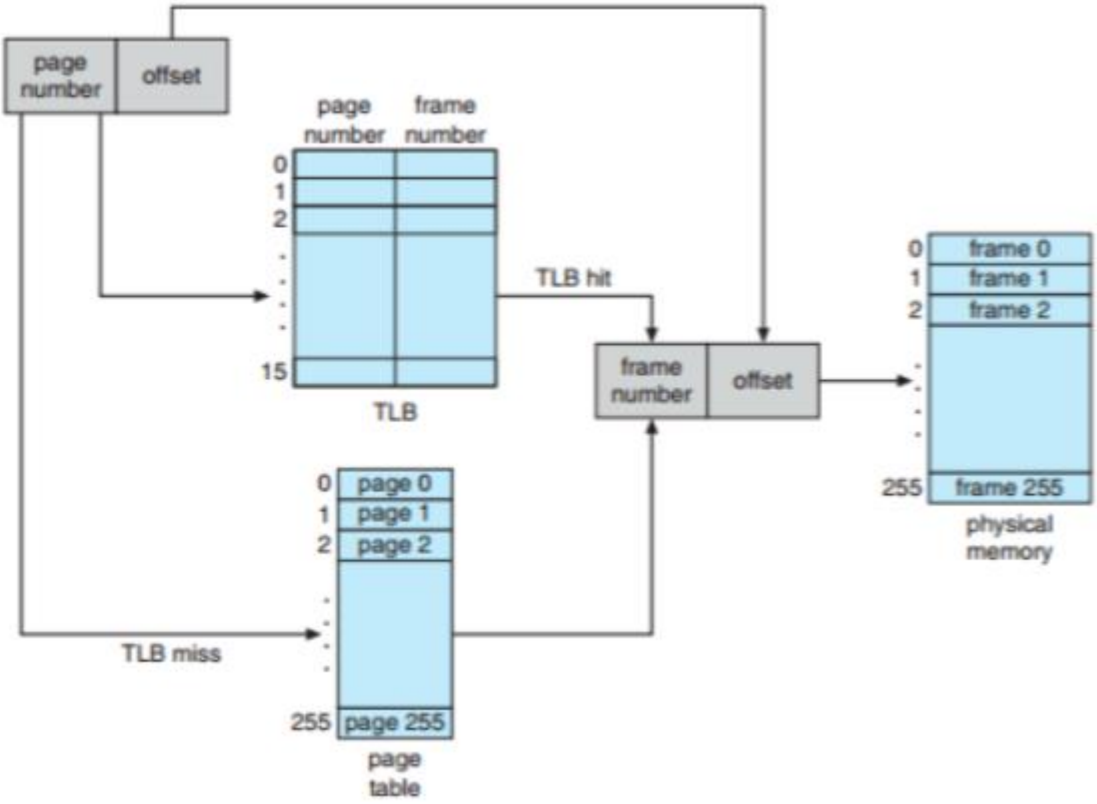
Specifics:

- Your program will read a file containing several 32-bit integer numbers that represent logical addresses. However, you need only be concerned with 16-bit addresses.
  - These 16 bits are divided into :
    - 8-bit page number
    - 8-bit page offset.



- Other specifics include the following :
  - 28 entries in the page table
  - Page size of 28 bytes
  - 16 entries in the TLB(Use FIFO replacement)
  - Frame size of 28 bytes
  - 256 frames
  - Physical memory of 65536 bytes(256 frames * 256-byte frame size)

- Address-translation process



Handling Page Faults:

- The backing store is represented by the file BACKING_STORE.bin, a binary file of size 65,536 bytes.
- When a page fault occurs, you will read in a 256-byte page from the file BACKING_STORE and store it in an available page frame in physical memory.
- For example :
  - If a logical address with page number 15 resulted in a page fault, your program would read in page 15 from BACKING_STORE (remember that pages begin at 0 and are 256 bytes in size) and store it in a page frame in physical memory.
  - Once this frame is stored (and the page table and TLB are updated), subsequent accesses to page 15 will be resolved by either the TLB or the page table.

- You will need to treat BACKING_STORE.bin as a random-access file so that you can randomly seek to certain positions of the file for reading. We suggest using the standard C library functions for performing I/O, including **fopen()**, **fread()**, **fseek()**, and **fclose()**.
- The size of physical memory is the same as the size of the virtual address space (65,536 bytes) so you do not need to be concerned about page replacements during a page fault.

Input:

- addresses.txt, which contains 500 logical addresses ranging from 0 to 65535.
- BACKING_STORE.bin

Output file:

1. The corresponding physical address : what your program translates the logical address to. (ex. 0xxxxxx_address.txt, 0xxxxxx is your student ID) (20%)
2. The signed byte value stored at the translated physical address. (ex. 0xxxxxx_value.txt, 0xxxxxx is your student ID) (20%)

Output on screen:

1. Page-faults (15%)
2. Page-fault rate : The percentage of address references that resulted in page faults. (15%)
3. TLB hit times (15%)
4. TLB hit rate : The percentage of address references that were resolved in the TLB. (15%)