**Trent University: Operating Systems (COIS3320)**

**Dr. Bin Guo**

**Lab 8: Main Memory Address Translation**

**Outline**

Assume that a system has a 32-bit virtual address with a 4-KB (=212) page size. The

physical memory address is also a 32-bit address. Consider a small program that needs

only 8 pages of memory. Below is the page table for this program.

**Page Table:**

A white rectangular paper with black text

Description automatically generated

In these labs you are to write a C (lab8.c) program that simulates an MMU’s (memory

management unit’s) address translation capability. To simulate a program’s memory

address requests, we use the text file named **labaddr.txt**. This file contains a sample of **20** logical addresses generated for this program. You are to read these addresses and output the

following for every address:

Logical/virtual address, its corresponding page number and offset, and its corresponding

physical address.

**There are three parts to this lab:**

1. Reading from a file.

2. Given a logical/virtual address output its page number and offset

3. Given the page table and the logical address output its corresponding physical address.

It is recommended that you approach these labs in the above order; that is, first read all the

logical addresses from the file and simply output it to the terminal. Then compute the page

number and offset and output these details for each logical address in the labaddr.txt file on

the terminal. Finally, compute the physical address and display it on the terminal.

**Part I – Reading from a file.**

1. Use the C library function openf()to open the ‘**labaddr.txt**’ file. Since you will be simply

reading from this file choose the ‘r’ (read) option.

Eg: FILE \*fptr = fopen("labaddr.txt", "r");

2. To read a logical address from the file use the fgets() function. Since the logical

addresses are no more than 10 characters long, you can read and store just 10 characters

at a time. Sample code is:

#define BUFFER\_SIZE 10;

char buff[BUFFER\_SIZE];

//Read from labaddr.txt till you read end of file.

while(fgets(buff, BUFFER\_SIZE, fptr) != NULL){...}

3. After reading a logical address print it to the terminal.

4. It is important that you close the file after you are done reading all the logical addresses

from the file.

Sample code: fclose(fptr);

**Part II – Given a logical address compute the page number and offset.**

1. Define PAGE\_NUMBER\_MASK, OFFSET\_MARK etc. as macro definitions. Sample code is

below, where you need to fill in the appropriate values in the blanks in your program.

#define OFFSET\_MASK \_\_\_\_

#define PAGES \_\_\_\_

#define OFFSET\_BITS \_\_\_\_

#define PAGE\_SIZE \_\_\_\_

2. For each logical address compute, the page number and offset using **operators in C** (See notes on it at the end of the document).

3. Print the logical address and its corresponding page number and offset to the terminal.

**Part III – Given the logical address and page table compute the corresponding**

**physical address.**

1. Define the page table as an integer array and store all the frame numbers as shown in

the page table under the outline section of this document.

Eg: int page\_table[PAGES] = {6,4,3,7,0,1,2,5};

2. After computing the page number (p) and offset (o), extract the frame number for the

page (p) from the page table.

3. Using the frame number compute the corresponding physical address **using bitwise**

**operators in C**.

4. Print the physical address along with the logical address and its corresponding page number and offset from PART II to the console.

5. Compile your program without errors and show the program’s output to your TA.

**Important:**

1. Note that for your program to run correctly (and to avoid segmentation faults) it is

important that you use correct data types for page number, frame\_number, virtual and

physical addresses and offset.

2. Refer to lecture notes on paging. This content is under the lecture notes slides on Main

memory (Chapter 9).

**Correct Program output:**

./lab8

Virtual addr is 19986: Page# = 4 & Offset = 3602. Physical addr = 3602.

Virtual addr is 16916: Page# = 4 & Offset = 532. Physical addr = 532.

Virtual addr is 24493: Page# = 5 & Offset = 4013. Physical addr = 8109.

Virtual addr is 8198: Page# = 2 & Offset = 6. Physical addr = 12294.

Virtual addr is 20683: Page# = 5 & Offset = 203. Physical addr = 4299.

Virtual addr is 18515: Page# = 4 & Offset = 2131. Physical addr = 2131.

Virtual addr is 28781: Page# = 7 & Offset = 109. Physical addr = 20589.