

Computer Project04

Assignment Overview

This assignment focuses on memory management within an operating system. You will design and implement the C/C++ program which is described below.

It is worth 30 points.

Assignment Deliverables

The deliverables for this assignment are the following files:

Project04.makefile – the makefile which produces "proj04"

Project04.student.c – the source code file for your solution

Be sure to use the specified file names and to submit your files for grading via the CSE Handin system before the project deadline.

Assignment Specifications

The program will simulate the steps to manage primary storage using paging. The system to be simulated contains 1,048,576 bytes of RAM which is divided into 256, i.e. 2^8 , page frames. Logical addresses are 16 bits in length.

1. Your program will process a file which contains zero or more memory references. Each line of the file will contain the following information:

- a) logical address being referenced (4 hexadecimal digits, with leading zeroes)
- b) operation being performed (one character; R for read and W for write)

Items in the line will be separated by exactly one space, and the line will terminate with a newline.

2. For each memory reference in the file, your program will display one line with the following information:

- a) logical address being referenced (4 hexadecimal digits, with leading zeroes)
- b) page number (1 hexadecimal digit)
- c) page offset (3 hexadecimal digits, with leading zeroes)
- d) operation being performed (one character; R for read and W for write)

Items in the line will be separated by exactly one space, and the line will terminate with a newline.

3. After the simulation is completed, your program will display the following counts:

- a) total number of memory references
- b) total number of read operations
- c) total number of write operations

The summary information will be appropriately labeled and formatted.

4. Your program will display the contents of the page table at the start of the simulation, after every N memory references, and at the end of the simulation. The page table will not be displayed twice in succession (there must be some intervening output related to memory references).

The display will contain one line for each page table entry:

- a) index of the page table entry (one hexadecimal digit)
- b) V bit (one character; 0 for not valid, 1 for valid)
- c) R bit (one character; 0 for not referenced, 1 for referenced)
- d) M bit (one character; 0 for not modified, 1 for modified)
- e) frame number stored in that page table entry (2 hexadecimal digits, with leading zeroes)

Items within a line will be separated by exactly one space, and each line will terminate with a newline.

The page table display will begin and end with a blank line, and will include appropriate column headers.

The value of N will be a positive integer value; any other value (such as 0) will prevent the program from displaying the contents of the page table.

5. Your program will accept two command-line arguments: an integer representing the value of N and the name of the input file.
6. The program will include appropriate error-handling.

Assignment Notes

1. As stated above, your source code file will be named "proj06.student.c"; that source code file may contain C or C++ statements.
2. You must use "g++" to translate your source code file in the CSE Linux environment.
3. Your program must create a data structure representing the page table and set all the entries to zero at the start of the simulation.

For this assignment, your program will not update the page table. Therefore, all entries in the page table should be zero in every page table display. If time allowed I would assign subsequent assignments wherein your program would actively manage the page table (and thus the page table display would change over time).

4. The following are valid executions of the program (if data files are accessible):

- proj04 5 /user/cse410/Projects/project04.test1
- proj04 1 project04.refs
- proj04 0 mytestcases

In the first example, the program will process the specified file and produce the required output. It will display the contents of the page table at the start of the simulation, after every 5 memory references, and at the end of the simulation.

In the second example, the program will process the specified file and produce the required output. It will display the contents of the page table at the start of the simulation, after every memory reference, and at the end of the simulation.

In the third example, the program will process the specified file and produce the required output. It will not display the contents of the page table.