CSE 325 Spring 2020

Computer Project #9

Assignment Overview

This assignment focuses on memory management within an operating system, and is the first milestone in a larger simulation. You will design and implement the C/C++ program which is described below.

It is worth 20 points (2% of course grade) and must be completed no later than 11:59 PM on Thursday, 4/2.

Assignment Deliverables

The deliverables for this assignment are the following files:

`proj09.makefile` – the makefile which produces **`proj09`**
`proj09.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 page frames. Virtual addresses are 16 bits in length.

1. Your program will accept command-line arguments to specify the file which contains the memory references (the "-refs" option), and to control the display of debugging information (the "-debug" option).

2. The "-refs" option will be followed by the name of the file which contains zero or more memory references. Each line of the file will contain the following information:

a) virtual address being referenced (four 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. For example:

```
3608 R
0e90 W
10d4 R
```

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

a) virtual address being referenced (four hexadecimal digits, with leading zeroes)
b) operation being performed (one character; 'R' for read and 'W' for write)
c) page number (one hexadecimal digit)
d) page offset (three hexadecimal digits, with leading zeroes)

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

```
3608 R 3 608
0e90 W 0 e90
10d4 R 1 0d4
```

4. After the simulation is completed, your program will display the contents of the page table. 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) P bit (one character; '0' for not present, '1' for present)
d) R bit (one character; '0' for not referenced, '1' for referenced)
e) M bit (one character; '0' for not modified, '1' for modified)
f) frame number stored in that page table entry (two 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.

5. 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.

6. If the "-debug" option has been selected, your program will display:

a) the contents of the page table at the start of the simulation
b) the contents of the page table after each memory reference is processed

7. The program will include appropriate error-handling.

Assignment Notes

1. As stated above, your source code file will be named "proj09.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. Valid executions of the program might appear as follows:

```
proj09 -refs test1 -debug
proj09 –debug –refs test2
proj09 –refs test3
```

4. Your program must create a data structure representing the page table and set all of 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. In subsequent assignments, your program will actively manage the page table (and thus the page table display will change over time).