Jason Anderson, Colter Assman, John Mangold
Program 3
CSC 456
Dr. C Karlsson

Description

This program is a simulation for multiple operating system topics including: shared memory, process schedulers, memory management, and page replacement algorithms. The user can type mailbox to use the shared memory and have access to the functions related to shared memory. Likewise the user can type scheduler to access the different process schedulers. The user can type mmu to access the memory management unit or else type pra to access the page replacement algorithms.

Process Scheduler

There are 3 different algorithms for process schedulers are Priority, Round Robin and Shortest Job First. In the priority algorithm the simulator will run the highest priority active process until a new process enters. It then checks the priority of the new process with the active ones to see if it needs to switch over to a different process. Round robin runs a processes until the quantum time is reached. It then switches over to the next active process once the quantum time was reached. Shortest job first runs the shortest remaining time until a new process enters. Once a new process has entered it checks all of the active processes and runs the one with the shortest time remaining.

Page Replacement Algorithm

There are 6 different algorithms for page replacement: FIFO, optimal, least recently used, least frequently used, clock, and second chance. The FIFO replace the oldest page in the frame and replaces it with the new page. The LRU(Least Recently Used) replaces the page that has been least recently referenced. The Optimal algorithm looks at what is coming in the future for pages and decides what would be the best page to replace. For this, I decided to look 3 pages ahead and decide based off of that. If the page is referenced in the next 3 pages, it is not replaced. In the Clock algorithm, once a page gets referenced, I set a bit in the bits vector. This is used when a page is going to be replaced. If this bit is set, it skips that page and sets that bit to 0. If a page has 0 bit set, it is replaced. The least frequently used algorithm keeps a count of the number of times a page has been used. It replaced the page with the smallest count. The Second chance algorithm is just like the clock algorithm except there is another bit that is set when a page is replaced. When a page comes up for replacement and one of the bits are set, it is set to 0. If both are set to 1, only the respective one is set to 0. If both are 0, then the page is replaced.

Memory Management Unit

The Memory Management Unit portion of the program simulates how pages are swapped from virtual memory to physical memory using the Translation Look-aside Buffer and the Page Table. This portion assumes that virtual memory, TLB, page table, and physical memory are already full so the reported statistics are not skewed by the initial entries causing page faults and such. The user will enter the number of pages to swap and those pages will be swapped between virtual memory and physical memory. The number of TLB hits and misses and the number of page hits and misses are tracked and once all pages have been swapped the statistics of each are printed out. The various pieces of the problem have varying capacities. The virtual memory can have 256 pages. The TLB table can have a maximum of 32 entries. The page table can hold the same as virtual memory and can hold 256 pages. The physical memory holds less than virtual memory and can only hold 128 pages.

External Files

The program uses function found in the external files process.cpp, mmu.cpp, pra.cpp, mailbox.cpp,

mailbox_functions.cpp.

Compilation

This program is compiled by using the makefile. Or by typing :
g++ -o prog3 Prog3.cpp process.cpp mmu.cpp pra.cpp mailbox.cpp mailbox_functions.cpp -std=c++11 -Wall
This program is run by typing ./prog3.

Usage

The program prompts the user to chose a topic: shared memory, process schedulers, mmu, or page replacement algorithms. It does this by giving the user instructions on what to enter and then giving them a space to enter the desired topic after “a >”.

a> mailbox
This will launch the shared memory mailbox shell.

a> mmu
This will launch the Memory Management Unit simulation

a> scheduler
This will launch the Process Scheduler simulation

a> pra
This will launch the Page Replacement Algorithm simulation

Testing

The program was tested by entering a space and <enter>, entering a <tab> and <enter>, and entering a <enter> with nothing else on the line. In all these cases the program prints out an error statement and then sends the user back to the prompt. Anywhere in the program where you are prompted for numbers and you enter characters (this includes '.' from doubles or floats) then the program will crash and segmentation fault.

Teaching Tool

This program can be used as a teaching tool as well as a simulation. It can be used to display real world results for things that happen all the time in a computer. For example, with the process scheduler program, it can show exactly what process scheduler is the most efficient. The MMU simulation shows exactly how memory is transferred and which parts of memory are accessed the most. The page replacement algorithm, again, gives real numbers for the efficiency of each page replacement algorithm. And finally the shared memory portion can be used to show two or more processes accessing the same information via shared memory in real time. These all provide real tangible numbers for something that is normally very abstract.

Submission

This documentation file is submitted along with the program file, function files (listed above), and the makefile.