ECE 3055a Laboratory Assignment 5
Due Date: Thursday, April 13, 2000

In this lab, you will use Java or C/C++ to write a program that will evaluate and compare the FIFO and LRU page replacement policies for virtual memory. Given a particular memory organization (size of virtual address space, size of a page, and size of physical memory), the program will read a file containing a memory trace (a sequence of virtual memory addresses) and determine which of the memory references will cause page faults under the two different replacement policies. The program should keep track of the total number of page faults generated by each policy over the entire trace. The total numbers of page faults for the two policies should be plotted for different page sizes and physical memory sizes (as detailed below). Do not forget to count compulsory page faults, i.e. automatic page faults that occur the first time each virtual page is referenced in the trace.

Files on 3055b WebCT, http://classweb.gatech.edu:8080/SCRIPT/ECE3055DA/scripts/serve_home.

There are 1,000,000 memory references in the trace file stored one per line. The virtual address is the second field on the line. The first field gives the type of the memory reference (2 for instruction fetch, 1 for store, and 0 for load). The third field gives the instruction value for a fetch and is always 0 for loads and stores. Your program only requires the virtual address and can ignore the other two fields. To help you get started on the lab, a simple program that reads the trace file, extracts the virtual address, and performs some simple bit-wise operations similar to what is needed to extract a virtual page number from the address is available on the Lab Web page in both C and Java formats.

The virtual address space is fixed to 32 bits and all virtual addresses are byte addresses. Plot the total numbers of page faults versus physical memory size for the two policies. Data points should be generated for physical memory sizes equal to 128 KBytes, 256 KBytes, 512 KBytes, and 1 MByte. Have plots for page sizes of both 4 KBytes and 16 KBytes.

You do not have to write code to plot the data, only to calculate it. The plotting can be done by any method you wish, e.g. by hand or with an available plotting program. You may want to simply edit the Matlab code handed out previously for plotting.

Caution: You will need to develop a page table data structure that is used by each policy to track the virtual pages that it keeps in physical memory as memory locations from the trace are accessed. If you use a page table indexed by virtual page number, it is likely that your program will experience memory allocation problems and/or excessive running time due to large page table size. Consider instead using an inverted page table, which is indexed by physical page number (see Silberschatz and Galvin, pages 269-270, for a description of inverted page tables). You do not need to implement hashing in your inverted page table because the time to search this relatively small page table is not a significant concern in your program. You also need to carefully consider what information to store in a page table entry to enable the FIFO and LRU policies to be implemented.