CMSI 587

OPERATING SYSTEMS (GRADUATE LEVEL)

Spring 2006

Assignment 0328

Not for Submission

Up next is file systems, so Chapters 10–11 will get you ahead for next week.

For Submission

Do the following exercises from SGG; submit your responses in hardcopy only. In case your edition is different from the 7th, I've added a note on the type of question involved (or even the entire question itself if it isn't too long); if you can't find the equivalent in your edition, then let me know and I can pass a copy to you.

1. SGG Exercise 9.5 — Given a demand-paged system that takes 8 milliseconds to service a page fault when an empty frame is available or a replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory access takes 100 nanoseconds, and 70% of pages to be replaced are modified. What should the page-fault rate be for an effective access time of ≤ 200 nanoseconds?
2. SGG Exercise 9.14 — Given a demand-paging system with disk access time of 20 milliseconds, main memory access time of 1 microsecond, page tables in main memory, and associative memory such as a TLB that reduces TLB hits on pages to a single memory reference: if the TLB hit ratio is 80%, and, of the TLB misses, 10% of those (or 2% of the total) cause page faults, then what is the effective memory access time of the system?
3. SGG Exercise 9.21 — Implement the FIFO and LRU page-replacement algorithms in software, where the virtual address space ranges from 0 to 9 and the available frames can vary from 1 to 7; demand paging is used.

   (additional instructions) Base your implementation on the following API, and use a series of *assert* statements to test your implementation. Place your functions in *pageReplacement.h* and *pageReplacement.c*, and place your test harness/main function in *pageReplacementTest.c*.

   > *void replacePagesFIFO(const char *refString, int frameCount, char *frameReport);*
   > *void replacePagesLRU(const char *refString, int frameCount, char *frameReport);*

   refString is a series of page requests (e.g. "043329129831"), *frameCount* is the number of available frames to simulate, and *frameReport* is the result of the reference string under the given algorithm and frame count, formatted as "xxxxxxx|xxxxxxx|xxxxxxx|…" such that the string between each pipe "|" represents the state of the *frameCount* frames before each page request. Use "-" to represent a free frame, and the corresponding page number when the frame is allocated.

   For example, *replacePagesFIFO("0867", 2, buffer)* would return "--|0-|08|68|67" in *buffer*. Note that you can predict the size of the output string based on the length of *refString* and *frameCount*, so memory allocation shouldn't be a huge issue.

   Commit your work to your Keck CVS repository under the path *homework/cmsi587/hw0328*. Place source code in a *src* subdirectory; if you have any other files, such as notes, documentation, configuration, or build files, commit those at the top-level directories. Make sure that your *.cvs* subdirectory (the physical location of your Keck CVS repository) is accessible to users other than yourself.