CS 33211

Homework #4

OS

Due in class at 12:30pm on Monday 3 December 2007

typed answers preferred

---

1. **In what way does demand paging resemble the instruction and data cache in the CPU?**

   Much like a CPU's instruction and data cache store frequently used code and data from the physical memory, demand paging treats the physical memory as a cache to store the frequently used code and data from the combined virtual memories of the running processes. Just as the CPU's cache is smaller but faster than physical memory, the physical memory ("cache") is smaller than the combined virtual memories but faster (since some of the data in virtual memory is stored on disk). Just as in instruction/data caching in the CPU, this process takes advantage of the Principle of Locality of Reference, requires a cache replacement policy, etc.

2. **Why is it so difficult to implement LRU page replacement, given all that's needed to do so is to record memory accesses times and maintain a sorted list?**

   Fully implementing LRU would require storing a time stamp for each memory access and likely maintaining some sort of sorted list. Just storing the time stamp would double the time required for every memory access, and then the sorting would take yet more time. Overall, this would be far too much overhead for a simple memory access!

3. **What is thrashing and why is it bad?**

   Thrashing is the wasted activity due to frequent paging that occurs when the combined working sets of the active processes are too big to fit into memory. In this situation, a particular process's active pages get paged into memory, paged out before the next time they are used, and then must be paged back in again — a huge amount of time that could be better used running processes instead of running OS code to handle paging.

4. **What is the advantage of multilevel naming over user-based naming?**

   With multilevel naming, file names only need be unique within a particular directory, whereas with user-based naming all files in that user's name space must be unique. So multilevel naming allows a student to have one directory for OS and one for database systems and have a file named "HW4.doc" in each, whereas user-based naming would force the student to differentiate between those file in a more cumbersome manner (e.g, "OS-HW4.doc" and "DS-HW4.doc").

5. **Why is it necessary to associate the current position in the file with a process rather than just with the file itself?**

Since multiple processes could be reading the file at the same time it is necessary to keep track of the location within the file on a per-process basis, which is easier to do if the information is stored in association with the process.