CS414 Fall 2004 Homework 8. Due in class on November 23, 2004

2. In class, we learned about the working set algorithm for managing paged memory. In the context of this algorithm:
   a) Many programs have "phases". For example, a compiler might have an input and parsing phase, a symbol-table analysis phase, a code generation phase, etc. Why would each phase be likely to have its own working set?
   b) Still in the context of a program that has phases, suppose that you discovered that all the pages for some segment of the program turn out to be in the working set from start to finish (e.g. despite the phase behavior, they stay in the working set). Give some possible explanations for what such pages might contain.
   c) Suppose that we were considering implementing a new system call to "pin" certain pages into memory. Once pinned, a page will no longer be a candidate for being paged out by the paging algorithm, and won't count against the working set of the program. Describe some situations where you might consider pinning a page using this new system call. (Note: no need to focus on programs that run in phases – part (c) is about programs in general).

3. We learned about dynamically linked libraries in class, too.
   a) List several advantages of using dynamic libraries
   b) We learned that each program must make its own copy of any data segments associated with a DLL, but that all programs using a DLL share the code. Explain the difference between a data segment and a code segment in just a few words (not an essay!) and then explain briefly why we need to create private copies for one, but not the other.
   c) While developing a new game program, suppose that you discover that a commonly used DLL has a bug. You track down the source file, fix the bug, and recompile the library. Now your game program begins to work perfectly. Yet other users are suddenly complaining that many of there programs have begun to crash! Why might fixing a bug in a DLL cause problems? Would the same issue have arisen if we were using traditional statically linked libraries? Explain.
   d) A motivation for the DLL concept was to reduce the total virtual memory size. Remind us of how DLL's can give this benefit. Now describe a situation in which moving to DLLs would actually cause the virtual memory in use to rise, presumably hurting performance relative to a solution that didn't have DLLs. You can assume that the library we are thinking of is the formatted I/O library for a programming language like C or C++ if it helps you "focus" on a specific scenario in answering this question.

4. Consider the disk buffer pool of a typical file system.
   a) Caching file system pages has pros and cons. List a few of each.
   b) Some studies show that as much as 50% of all files are deleted within a few seconds after being created! In fact one study of Microsoft Windows NT showed that a great many files are deleted within milliseconds after being created. How might you exploit this knowledge in designing a disk buffer pool (cache) management strategy?

5. Some designers of database systems complain that file systems and buffering only mess their performance up and that they would prefer to have complete control over (1) what gets prefetched, (2) what gets held in the kernel buffer pool, and (3) how files are mapped onto disk blocks.

   A typical scenario where this arises is when a database has some form of "index" mapping keys to blocks and records within those on the file system. For example, "Ken" might map to "14/7", meaning that the data associated with key "Ken" is in block 14. The "7" would mean that block 14 has multiple fixed-size records in it and Ken's is the 7^{th} record. For example, a 1024-byte block size could hold roughly 100 records each of size 100 bytes, with 24 bytes left over for other purposes.

   The index file itself would typically be a balanced tree, like a binary tree. Focusing on this case, look at each of the complaints mentioned above. Would control over the specified behavior have a potentially big performance impact, compared with the sorts of automatic policies we discussed in class?

6. Suppose that on a Linux file system, the GNU C compiler is in a file called "/usr/bin/gcc". Now Doug comes along and creates a (true) link to that file, calling it "/users/doug/bin/c".
   a) Doug's reason for creating the file system link is that he really hates 3-letter command names. Doug prefers to just type "c" instead of "gcc". But does he really get the same program that someone else would get when they run the program by the name "gcc"? Explain.
   b) Sally the system administrator needs to install a new version of GCC. So she compiles the new version, then deletes the old one, copies the new one into place, and gives it the identical name to the old one. When Doug runs "c" will he still get the old program, or will he get the new one now?
   c) Same as part (a), but now assume that Doug created a symbolic link, not a true link.
   d) Same as part (b), but under the same assumption – Doug's version is a symbolic link.