Fall 1998 Comprehensive Exam: Software Systems (30 points total)

SOLUTIONS

1. (10 points) In UNIX the file directory structure (including hard and soft links) may be an acyclic graph. In MS-DOS it may only be a tree structure.

   a. Compared to a tree structure, what are the advantages and disadvantages of the acyclic graph directory structure?

Advantages: Files may be shared by more than one name in the directory structure (using links, they may have more than one name). This has many uses. For instance, one can maintain compatibility between old and new software that expects certain files to be in different locations.

Disadvantages: Traversing up the directory structure is more of a problem, since a file may have more than one parent if you include links.

A file may have more than one pathname, so when traversing the directory structure, you cannot rely on a new pathname indicating that it points to a file you have not already traversed.

Hard links require reference counting, which adds complexity.

Software links may be broken (left pointing at nothing).

You must have some way to ensure that cycles are not created in the directory structure. (UNIX does not allow users to create hard links to directories.)

   b. Why is a general graph directory structure not usually used for a file system?

The problem is cycles in the graph structure. If you remove certain elements, you may leave some files and directories “stranded” since their reference count will be zero, but there is no pathname by which to refer to them. In this case you must traverse the whole structure and do garbage collection.

2. (10 points) This question concerns page replacement.

   a. When is a process considered to be thrashing?

A process is considered to be thrashing when it is spending more time paging than executing.

   b. What is the difference between global and local page replacement algorithms?

In global replacement, the replaced page may come from the address space of a process other than the one that took the page fault. In local replacement, the page fault may only grab a page frame belonging to another page in the same process's address space.

   c. What are the advantages and disadvantages of the two types of page replacement algorithms?

With global replacement, overall system performance may be better, since a process that needs more pages can take page frames from a dormant process that isn't really using them. Thus the physical memory is more efficiently utilized. The disadvantage is that the paging behavior of one process can affect the performance of another process.

With local replacement, a badly faulting process cannot kick out the pages of another process, so its paging behavior will not hurt other processes. The disadvantage is that physical memory may not be as efficiently utilized.

3. (10 points) Let's assume that a new type of main memory is invented that is cheap and usually very fast -- almost as fast as hardware registers. The only problem is that 0.1% of the time you access this memory, it is actually very slow (as slow as a disk access). The problem is that you have no way of predicting ahead of time when these slow accesses will occur.

   a. Would you incorporate this new type of memory into your system? Why or why not?

I'll accept both answers here, if they include a convincing explanation. This new type of memory is tricky, though, since it causes more variability in system performance and behavior.

   b. Assuming you must incorporate it, would you make any changes to the virtual memory system or the file system or the network software or the process scheduler or the structure of the kernel?

Assuming the hardware will allow it, you will want to be able to put processes asleep if they access memory and it turns out to be one of the very slow accesses. (You could have a timeout whose expiration indicates it's a slow access.) This would affect the VM system and process scheduler. The network software may also be affected, since touching network buffers may trigger a slow access in the middle of, say, sending an ack. This means that timeouts in protocols may need to be extended. The kernel may need to be multi-threaded to allow some threads to sleep when they have triggered a slow access. File system cache hits no longer necessarily indicate that the file block sought will be speedily accessible. Thus you may need to allow processes accessing the file cache to sleep even on a cache hit. You may even want to make block/page sizes smaller so that there are fewer word accesses that could trigger a slow access if only a portion of the block/page is actually needed. (Overall, though, for the same amount of data actually copied or accessed, you'll suffer the same number of slow references.)