Name: _____

CS 33211 **Final Exam** OS

Thursday 13 December 2007

1. **Why is linking multiple object files together into a single executable object file difficult? (10 points)**

   Combining all the individual text, data, bss, heap, and stack segments together into larger combined text, data, etc. segments isn't difficult, but the difficulty is that any addresses for instructions or data need to be modified to reflect the new locations of those instructions / data. Further, external references from one object file to addresses in other object files must be resolved using the patch list so that those instructions have the correct address in the executable file.

2. **Distinguish between "logical memory" and "physical memory". (5 points)**

   Logical memory is the memory space seen by an individual process, which always sees its memory space as starting at location 0 and continuing to some maximum address, regardless of where it is placed in physical memory. Physical memory is the actual memory space in the computer's DRAM memory, which is shared between the OS and multiple processes.

3. **What is the function of the Memory Management Unit (MMU)? (5 points)**

   Programs are compiled and linked into executable object files as if the object file will be loaded into memory at address of 0, with all instruction and data addresses relative to 0. When the program is loaded into memory at some other location, any reference to memory has its address modified on the fly by the MMU, which adds the program's base location to the (logical) memory address to produce a (physical) memory address.

4. **Consider dynamic variable-sized partitioning versus segmentation.**

   a. **In what way(s) are these techniques similar? (10 points)**

      Both are assigned variable-sized sections of memory, and as such suffer from external fragmentation. Both require the entire process to be in memory to run.

   b. **In what way(s) are these techniques different? (10 points)**

In dynamic variable-sized partition the entire process must be placed in a single memory section, while in segmentation the process can be broken up into multiple memory sections.

5. An important component of demand paging is a good page replacement policy. How difficult is it to implement Least Recently Used (LRU) replacement? Explain. (10 points)

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.

6. How does an OS such as Windows know what is stored in a file? (10 points)

Windows typically determines the file type from the file's extension — ".doc" indicates a Word file, ".jpg" indicates a picture, ".mp3" indicates music, etc.

7. What is the relationship between a directory and the file description representing that directory? (10 points)

In general, a file descriptor contains pointers to the data blocks storing the file's contents and contains accounting information such as the file's owner and creation time. The directory contains a list of file names for the files in that directory along with pointers to each file's file descriptor, but is itself pointed to by a file descriptor that represents the directory.

8. Describe the C-SCAN disk head scheduling algorithm, using a diagram if necessary. Be certain to describe it clearly enough to distinguish it from other disk head scheduling algorithms. (10 points)

The disk head starts at the inner or outer track and goes all the way in the other direction, servicing requests as it goes. Once it reaches the end it goes all the way back to the beginning (without servicing request), and then repeats this process. The key is that it continually goes all the way from one side of the disk to the other, but only services requests while going in one of the directions.

9. There are many techniques for organizing the disk blocks that comprise a file.

a. What advantage(s) does linked allocation have over continuous allocation? (10 points)

Linked allocation allows disk blocks to be placed at arbitrary locations on the disk, rather than requiring them to be placed contiguously. Thus it is only necessary to find enough free blocks, rather than using a more complex algorithm such as first fit or best fit to efficiently use disk space and avoid external fragmentation between contiguously-allocated files. It is also good for random access, and for allowing the file to grow.

b. How does multilevel indexed allocation improve on indexed allocation? (10 points)

Instead of having a single index of a (probably large) fixed size, multilevel index has multiple smaller indexes. The set level of file pointers is simply stored in the file descriptor, while the second set of file pointers is stored in a separate disk block but only used if needed, and yet a third set is stored in a separate disk block which in turn points to more pointers which finally point to the file block and again is only used if needed. Thus multilevel indexed allocation is very efficient — allocating disk blocks to hold file pointers only if needed — yet expandable enough to support very large files.