CS 4/53201 # Homework #3 **Operating Systems**

Due in class on Friday 1 December 2000

1. **(Exercise 8.9 from OSC) Why is it that, on a system with paging, a process cannot access memory it does not own? How could the operating system allow access to other memory? Why should it or should it not?**

   A process can only access memory using the page table that corresponds to that process, and since frames corresponding to memory that it does not own would not be in its page table, it can not access those frames.

   The OS could allow such access — allowing the process to share data with another process — but duplicating pages to be shared in both processes' page tables. However, this isn't very "clean" in that it allows sharing of a particular page rather than sharing of a particular data structure — which can be hard to determine and which also allows sharing of data structures for which sharing might not be desired. In general, this probably is not a good idea.

2. **(Exercise 9.20 from OSC) What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?**

   Thrashing is cased when the sum of all the active processes' working sets is greater than the physical memory, meaning that almost every memory access causes a page fault. It can be detected by watching for an abnormally high frequency of page faults or low CPU utilization, and eliminated by reducing the degree of multiprogramming (the number of processes that can run at one time), and perhaps by using local page replacement (only replacing pages in use by itself).

3. **(Exercise 10.8 from OSC) Some systems automatically open a file when it is reference for the first time, and close the file when the job terminates. Discuss the advantages and disadvantages of this scheme as compared to the more traditional one, where the user has to open and close the file explicitly.**

   Opening and closing automatically has the advantage of making the programmer's job slightly easier. However, two disadvantages are that it requires the OS to check at each file reference to see if the file has been opened yet or not, and requires a directory lookup rather than the use of a file identifier (which would be used in a system with explicit open / close operations). Also, this scheme leaves the file open longer than necessary, which could be a problem if there is a per-process limit on the number of open files.

4. **(Exercise 11.5 from OSC) Consider a system that supports the strategies of contiguous, linked, and indexed allocation. What criteria should be used in deciding which strategy is best utilized for a particular file?**

   Many possible answers here — the choice of strategy should depend on the type of access (sequential or random), fragmentation, size of the file, whether the file might grow or not, ease of free-space management, etc. The answer should expand on a few of these, and how the 3 strategies apply.

5. **(Exercise 13.1 from OSC) None of the disk-scheduling disciplines, except FCFS, are truly fair (starvation may occur).**

   a. **Explain why this assertion is true.**

      All of the algorithms except FCFS use shortest seek distance to determine which request to service next, and do not in any way consider how long a request has been waiting.

   b. **Describe a way to modify algorithms such as SCAN to ensure fairness.**

      Use length of waiting time in some way to increase the priority of older requests.

   c. **Explain why fairness is an important goal in time-sharing systems.**

      Fairness is important to give reasonable and predictable response time in honoring user's requests.

   d. **Give three or more examples of circumstances in which it is important that the operating system be unfair in serving I/O requests.**

      It may be desirable to give priority to demand paging over application I/O.

      Disk cache writebacks may need higher priority.

      When saving data in the event of a crash, writes may need to be given priority over reads.

      Other answers also possible.