Homework 7 — due on Tuesday 5 December 2000

Total number of points available on this homework is 150. Full credit is equivalent to 100 points.

1. (15 pts) Suppose a physical page frame is shared by multiple processes. If the OS uses global page replacement with the clock algorithm, how does page sharing affect the page replacement algorithm? If the OS uses per-process page replacement with the clock algorithm, how does page sharing affect the page replacement algorithm?

2. (10 pts) Assume a page reference string for a process with m frames, initially empty. The page reference string has length p with n distinct page numbers occurring in it.

   (a) What is a lower bound on the number of page faults?

   (b) What is an upper bound on the number of page faults?

3. (15 pts) You are designing an operating system for a database server. The entire system is to be devoted to serving database queries on one large database. Users of the database may also add and delete entries in the database. How would you design your file index structure to support fast random access for this application? How expensive are additions and deletions in your scheme?

4. (15 pts) Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory. For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:

   (a) How is the logical-to-physical address mapping accomplished in this system? (For indexed allocation, assume that no file needs more than one index block.)

   (b) If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

5. (10 pts) When the average queue length is small, do all disk-scheduling algorithms reduce to FCFS? Why or Why not? Does locality of reference exaggerate differences in scheduling algorithms? Why or why not?

6. (20 pts) Disk requests are not usually uniformly distributed. For example, users access the file directory structures more frequently than other files. Assume that accesses to the file directory structures account for 50% of all accesses. (1) For each of the following algorithms, how would it perform, assuming that the file directory structures are co-located at a random spot on the disk and why? (2) Which algorithm takes best advantage of this locality? (3) For each algorithm, also specify where you would locate the file directory structure to improve the algorithm's performance, if some such placement exists.

   (a) FCFS

   (b) SSTF

   (c) SCAN

   (d) C-SCAN

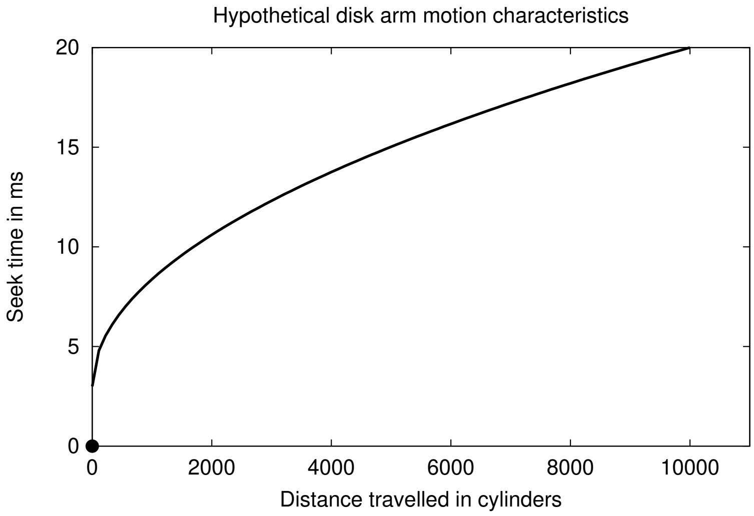
   Assume that SCAN and C-SCAN use their LOOK variations (i.e., they only continue to seek in a direction if there is a request in that direction).

7. (15 pts) Do problem 11.2 in Stallings. (Then look at the next question in this homework to see why 11.2 in Stallings is too simplistic.)

8. (Extra credit, 50 pts, due on Friday 8 December, should be handed in separately) Suppose that a disk is dedicated for use as backing store in a virtual memory system. Suppose further that the sequence of disk accesses is determined by the page faults experienced by your favorite program under your favorite page replacement algorithm of Project 2 (other than OPT).

   Let a be the lowest page number in the virtual address space of the program, and b the highest, such that $N = b - a + 1$ is the number of pages in the program's address space. Suppose that the entire disk is used to store pages of the program. The disk has $M = 10000$ cylinders, numbered 1 through M. Suppose that pages are laid out almost linearly on disk, such that page number n is on cylinder $\mathsf{round}(\frac{M}{N}(n - a + 1))$, where round rounds to the nearest integer in the range 1 through M. (This assumes that there are no big gaps in your program's address space. If you discover that there are gaps, consider not mapping them to disk.)

   Suppose that seek time, the time for disk arm movement, is independent of the absolute position of the disk arm, and only dependent on the relative distance travelled, given by the difference in cylinder coordinates d, as follows: $f(d) = C + L\sqrt{d/M}$, where $C = 3$ms and $L = 17$ms, except that $f(0) = 0$.

   Ignore rotational delay, transfer delay, and any other delays but seek delay.

   Assume that page faults occur exactly every 3ms. (Note: pretend that the sequence of page faults is actually an aggregate stream from a multitude of programs. Otherwise it would be difficult to explain how a program can generate new faults while it is waiting for a fault to be serviced.)

   Assume there is no difference between disk reads and writes.

   What is the total seek time for processing the entire stream of requests, if the set of pending disk requests is managed by the following algorithms:

   (a) FCFS

   (b) SSTF

   (c) SCAN

   (d) C-SCAN

   Plot the distributions of seek distances and seek times in each case. Calculate the average seek distance and average seek time.