COMP 252 Principles of Systems Software

FALL 2000, FINAL EXAM TRAINING PROBLEMS

1. Suppose we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds.

Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

Answer:
200ns = (1-P) x 100ns + 0.3 x P x 8ms + 0.7 x P x 20ms
P = 0.000006

2. What is the cause of thrashing? How does the system detect the thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

Answer:
Trashing is caused by underallocation of the minimum number of pages required by a process, forcing it to continuously page fault. Trashing can be detected by evaluating the level of CPU utilization compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

3. Consider a demand-paged computer where the degree of multiprogramming is currently fixed at four. The system was recently measured to determine utilization of CPU and the paging disk. The results are one of the following alternatives (i.e. a, b, c). For each case what is happening? Can the degree of multiprogramming be increased to increase the CPU utilization?

CPU utilization 10%, disk utilization 95%
CPU utilization 90%, disk utilization 3%,
CPU utilization 10%, disk utilization 3%

Answer:
a) The system is spending most of the time serving page faults, i.e. trashing. We should decrease degree of multiprogramming in this case.

b) CPU utilization is sufficiently high to leave the system in the current state. However, still some increase in the degree of multiprogramming is possible.

c) System is underutilized. We should increase the degree of multiprogramming.

4. Explain the difference between internal and external fragmentation. Which one occurs in paging systems? Which one occurs in systems using pure segmentation?

5. What are the *advantages* of using a **linked allocation** of disk blocks, versus a **contiguous allocation**, for a large file? What are the *disadvantages*?

6. Consider two disks, one with 500 cylinders and 10 surfaces and the other with 1000 cylinders and 5 surfaces, both having the same number of sectors per track and the same read/write heads seek characteristics (the same seek time for the same number of traversed tracks). Which one will give better performance to the file system with contiguous block allocation.

Answer: The first one, since the cylinder capacity is larger, and less seeks are needed in accessing the files.

7. Consider a file currently consisting of 100 blocks. Assume that the file control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked and indexed (single level) allocation strategies, if for one block, the following conditions hold. In the contiguous-allocation case, assume that there is no room to grow in the beginning, but there is room to grow in the end. Assume that the block information to be added is stored in memory.

a. The block is added at the beginning.
b. The block is added in the middle.
c. The block is added at the end.
d. The block is removed from the beginning.
e. The block is removed from the middle.
f. The block is removed from the end.

Table 1: Answer

	Contiguous	Linked	Indexed
a	201	1	1
b	101	52	1
c	1	3	1
d	198	1	0
e	98	52	0
f	0	100	0

8. An operating system only supports a single directory, but allows that directory to have arbitrarily many files with arbitrarily long file names. Can something approximating a hierarchical file system be simulated? How?

Answer: We may structure the file name in such way to resemble the file path in the hierarchical file system, for example file name could be in the format /user/application/filename. By parsing the filename with respect to the character “/” we may simulate the hierarchical file system