CSE 332 Bütünleme

Olcay Taner YILDIZ

I. QUESTION (18 POINTS)

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.

- The block is added at the beginning
- The block is added in the middle
- The block is added at the end

II. QUESTION (20 POINTS)

Consider the situation that there are $n = 99$ processes $P_1, P_2, P_3, \ldots, P_n$ arriving simultaneously in that same order $P_1, P_2, \ldots, P_n$ with CPU burst times as follows: Each of processes P_1 through P_{33} has a CPU burst time of 99 milliseconds. Processes P_{34} through P_{66} each has a CPU burst time of 66 milliseconds. And processes P_{65} through P_{99} each has a CPU burst time of 33 milliseconds. Consider running the following algorithms on this process sequence: FCFS, SJF, SJRF (pre-emptive), RR with quantum = 1 millisecond and RR with quantum = 50 milliseconds. Compare each of these five algorithms on this process sequence in terms of their average response times. In particular, order the five algorithms on this process sequence from best average response time to worst average response time.

III. QUESTION (22 POINTS)

We are given $n > 3$ processes, $P_0, P_1, P_2, \ldots, P_{n-1}$ with statements $S_1, S_2, \ldots, S_n$ respectively corresponding to each process (such that statement S_i is in the code of process P_{i-1}) with the following constraints:

- There are no constraints on S_1.
- If i is even, then S_i must execute after both statements $S_{\frac{i}{2}}$ and S_{i-1} has completed execution.
- If i is odd, then S_i must execute after statement S_1 has completed execution.

a. Using arrays of semaphores with appropriate initializations and separating into reasonable cases, write the code for the processes in such a way that enforces the above precedence constraints.

b. Consider adding the following additional constraint: If i is prime, then S_i must execute after $S_{(i+3) mod n}$ has completed execution. Using ideas from resource allocation graphs, where each statement is a resource, demonstrate that this additional constraint is guaranteed to create a deadlock situation. Similarly, explain why the original set of constraints does not create deadlock.

IV. QUESTION (20 POINTS)

The sleeping-Barber Problem: A barbershop consists of a waiting room with n chairs, and the barber room containing the barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy, but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers.

V. QUESTION (20 POINTS)

Let $n > 1$ be given, and say we have $n + 1$ memory partitions $2^n, 2^{n-1}, 2^{n-2}, \ldots, 2, 1$ kilobytes given in that order. Recall that $2^n = \Sigma_{i=0}^{n-1} 2^i + 1$. Consider that $2n$ processes arrive in the following size order: First n processes arrive with sizes in order $2^{n-1}, 2^{n-2}, 2^{n-3}, \ldots, 2, 1$ kilobytes followed by n more processes with sizes in the order $2^{n-1}, 2^{n-2}, 2^{n-3}, \ldots, 2, 2$ kilobytes (note that only the last process size is different: 2 instead of 1). Prove that amongst the algorithms First-Fit, Best-Fit, and Worst-Fit, only Best-Fit will succeed in placing all of the $2n$ processes into the available partitions.