

Wednesday 6 November 2002

1. Define the network term “connection-oriented communication”. (10 points)

In connection-oriented communication, there are explicit connection, data exchange, and release phases, and in the data exchange phase the information is delivered as a stream of bytes in the correct order.

2. Shortest-Job-First (SJF) and Shortest-Remaining-Time (SRT) are two related CPU scheduling algorithms.

a. How do these two algorithms differ with respect to picking the next process to run on the CPU? (8 points)

Shortest-Job-First is non-preemptive, meaning it only picks a new process for the CPU when the current process terminates or blocks. In contrast, Shortest-Remaining-time is preemptive, and is also invoked when a new processes enters the ready list (a newly-created process, or one that was blocked and has now been awakened).

Once invoked, SJF picks the process to run that has the smallest next CPU burst (not counting the current process, which has just left the CPU), whereas SRT picks the process in the ready list that has the smallest next CPU burst, but only if that CPU burst is smaller than the remaining time left in the current process’s CPU burst.

b. How do these two algorithms differ with respect to their treatment of processes with a long CPU burst? (8 points)

In SJF, if a process with a long CPU burst gets on the CPU, it will be allowed to finish. In SRT it will likely be preempted, but at least having been on the CPU it will be shorter and more likely to be selected next time.

(Both algorithms penalize processes with a long CPU burst, in that those processes are unlikely to be selected for execution and could potentially starve, but this is a similarity between the algorithms, not a difference.)

3. It is theoretically possible to prevent deadlock by forcing a process to request all resources at the beginning of its execution and release them at the end of its execution.

a. Why would this prevent deadlock? (6 points)

It eliminates the hold and wait condition, which is necessary for deadlock.

b. Why is this solution impractical? (6 points)

Name: _____

It is difficult to know in advance what resources will be needed. This solution is also overly greedy and wasteful as it ties up resources for a prolonged period and thus reduces resource utilization. Finally, this solution can allow starvation.

4. Coffman's algorithm for deadlock detection can be summarized as:

Every process is initially unmarked.

- 1. Look for an unmarked process P_i for which the i -th row of the Request matrix is less than or equal to the Available vector**
- 2. If such a process is found, add the i -th row of the Current matrix to the Available vector, mark the process, and go back to step 1**
- 3. If no such process exists, the algorithm terminates**

When the algorithm terminates, any unmarked processes are deadlocked.

What does it mean to mark a process? Why is there no deadlock in the system if the algorithm results in every process being marked? Be explicit in your answer. (10 points)

The algorithm marks a process when it finds sufficient resources to meet that process's needs, meaning if it gives the process its requested resources, it will terminate, and release both those newly-requested resources as well as those that it held before (represented by its row in the Current matrix), back in to the Available pool.

If every process is marked in this manner, that means that resource allocator can grant resources to the processes in that order, and the system will not deadlock since there will always be sufficient resources available.

5. When a process is loaded into memory, it has four major components. List those four components and briefly describe what each is used for. (20 points)

Text segment (program code)

Data/bss segments (initialized and uninitialized static global data, and constants)

Heap (data allocated by malloc in C, objects allocated by new in C++)

Stack (parameters passed to functions, local variables in functions)

6. There are many techniques for allocating memory and providing transparency and safety / protection.

a. What advantages does segmentation provide over dynamic relocation? (10 points)

Instead of having to fit the entire process into one contiguous space in memory as in dynamic relocation, segmentation breaks the process into smaller segments, and only each segment has to fit into a contiguous space. That is to say, it is necessary to fit multiple small segments into memory, instead of one large process.

Individual segments can also be shared between processes, and specific forms of protection can be applied to certain segments (i.e., read-only access to code segments).

Name: _____

b. What disadvantages do the two techniques share)? (10 points)

With both techniques, it is necessary that the entire process reside in memory in order to run, external fragmentation is possible, memory allocation is complex, and virtual addresses must be dynamically converted to physical addresses as the program runs.

7. In paging, what is the function of the page table? (12 points)

The page table for a particular process contains the number of the frame in physical memory that corresponds to a particular page number in virtual memory. It is used by the memory management unit (MMU) to translate virtual addresses into physical addresses.