24/11/16

Dual Degree (B.Tech. & M.Tech) 5th Semester (CST) Final Examination, 2016

Indian Institute of Engineering Science and Technology, Shibpur,
Howrah-711103, India.
Operating System (CS-503)

Time: 3hrs. Full Marks: 70

Answer any five questions. Credit will be given to precise answer.

1.

(a) Briefly explain the structure of a typical `Unix` kernel and the services provided by each subsystem in the kernel.

(b) Differentiate between process and program.

(c) Explain how to reap `zombie` processes?

(d) How does `copy-on-write` improves over `fork()`?

[4+2+4+4]

2.

(a) What happens to the parent process ID of a child process when its parent process terminates before it (i.e. the child process)?

(b) How does the sharing of files occur across a `fork()` system call? Explain it with a diagram.

(c) What do you understand by `context` of a process? Explain what happens when a `context switch` occurs.

(d) What are the limitations of using a `pipe` for interprocess communications? How can it be removed?

[2+4+4+4]

3.

(a) How many different types of solution exist to solve critical section problem. Write down the assumptions taken by these solutions.

(b) Show that if *`wait`* and *`signal`* operations are not executed atomically then mutual exclusion may be violated.

(c) Explain how more than two processes can get the same level in `bakery` algorithm.

(d) Implement the following scenario using `semaphore`.
There are two processes, P_1 and P_2 and they share a critical section (say C). I want P_1 and P_2 to execute C alternatively.

[3+3+4+4]

4.

(a) Is it possible to have deadlock involving only one single process?

(b) A system involving n processes is said to be in `safe state` only if there exists a `safe sequence`. Explain why existence of only one `safe sequence` is enough for the system to be in `safe state` instead of all $n!$ sequences.

(c) Consider a system consisting of N resources of the same type that are shared by $N-1$ processes, each of which needs at most two resources, show that the system is deadlock-free. In the above scenario if the number process becomes N, will it still be deadlock-free?

(d) What is `starvation`? Can a system detect that some of its process are `starving`? Explain how the system deal with `starvation` problem.

[3+3+4+4]

5.

(a) Why are page sizes always powers of 2?

(b) 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 process's memory? Why should it or should it not?

(c) What is PTBR? What are the disadvantage of using PTBR and how is it solved?

(d) Consider a paging system with the page table stored in memory.

(i) If a memory reference takes 200 nanoseconds, how long does a paged memory take?

(ii) If we add associative registers, and 75 percent of all page-table reference are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.)

[2+4+4+4]

6.

(a) Assume you have a page reference string for a process with m frames (initially empty). The page reference string has length p with n distinct page page numbers occurring in it. For any page-replacement algorithms,

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

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

(b) Explain the 2nd chance algorithm (clock algorithm) for page replacement with an example

(c) How stack can be used to implement LRU page replacement algorithm?

(d) What is working set? How it helps to prevent thrashing?

[4+4+3+3]

7.

(a) How many different types of directory organizations are there? Explain their advantages and disadvantages.

(b) Explain disk space allocation of file using following strategies with their pros and cons.

(i) FAT (File Allocation Table)

(ii) Indexed Allocation

(c) What are `inode`? Briefly explain the structure of a typical `inode` in `UNIX`.

[4+6+4]