

# Paper: Operating Systems Concepts

Code: INFO3102

## Chapter: Process & Thread

Full Marks: 100

1. Differentiate process and thread. "Thread is called light weight process" – Justify.	3+2
2. With proper code example explain the difference between Zombie and Orphan process.	5
3. What is the difference between fork and vfork? When should we use fork and wait system call combination instead of using vfork?	2+3
4. What is context? How context switching takes place? "Context Switching is overhead to the System" – Justify.	2+3+2
5. Which multithreading model is best according to you and why?	5
6. Differentiate between Kernel Level and User Level Thread.	5
7. Draw and explain process state diagram including both active and suspended states.	5
8. Draw the queuing diagram and state at which transition which scheduler works.	5
9. What is PCB? Why we need to have pointer in PCB? What is the purpose of storing CPU Register and Special Purpose Register values in PCB?	2+2+2
10. What is Mid-Term Scheduler? When it is required? Why the name is so?	5
11. Why CPU scheduler is called short term scheduler and Job scheduler is called long term scheduler? State this answer after clarifying the tasks of these two types of schedulers.	5
12. What is cooperating process and why we need them?	3
13. State the different types of message passing architectures and which one is required when. When should we use Shared Memory to communicate between processes?	8
14. Draw and explain the different steps that we need to follow to perform RPC.	5
15. What is socket? How it is used?	5
16. What is cascaded termination of processes? When it happens?	3
17. What does execlp system call do?	3
18. Consider the following code snippet. What will happen if it executes? Explain. [Consider the main process id is 3456 and all subsequent child processes are receiving their ids just by adding one with parent process; every process should have its own unique id] <pre>main(){     fork(); fork();     printf("%d\n",getpid());     fork();     printf("Hello\n");     return 0; }</pre>	5
19. What do you mean by address space of a process?	4
20. What will happen if we execute the following code snippet? Explain	6

<pre> main(){     int p,q=20;  p=fork();     if(p&lt;0){         printf("Fork failure\n");         exit(0);     }     else if(p==0){         q=100;         printf("q=%d\n",q);     }     else{         wait(NULL);         p=vfork(); </pre>	<pre>         if(p&lt;0){             printf("Fork Failed\n");exit(0);         }         else if(p==0){             q=200;             printf("q=%d\n",q);         }         else{             printf("q=%d\n",q);         }     }     return 0; } </pre>	
---	---	--