

1. <b>4 Primary Processing States</b>	Ready state, Running state, Waiting state and Finished state.	16. <b>Process Control Block (PCB)</b>	It is a data structure that keeps track of the key information about the process, such as its unique identifier, process state, allocated resources, priority, addresses, and so on.
2. <b>Cache</b>	It is based on the assumption that a significant amount of the data that was just removed from memory will be requested again in the very near future.	17. <b>Processing</b>	It is the work that the CPU performs.
3. <b>Cache memory</b>	Works as a temporary storage area on the CPU chip where recently used data is kept.	18. <b>Queue</b>	The order in which processes are put into. The processes themselves do not move, and neither do the PCBs. Instead, each PCB is linked from one to the next using a pointer, and the order is created through a series of links.
4. <b>COBEGIN and COEND</b>	These commands designate when several CPUs can execute several processes.	19. <b>Race</b>	A situation where the outcome of a computation can vary depending on some aspect of luck or chance.
5. <b>CPU</b>	It is the common name given to the Central Process Unit, also known as the processor.	20. <b>Ready State</b>	It is a state that a process is put in when it is first created. Also known as an incoming process. A process is also in the Ready State after it has used up its allotted time slice. It then waits for a new time slice before it can transition to a Running State again.
6. <b>CPU Responsibilities</b>	To interpret and execute program instructions or requests and perform every calculation for every user.	21. <b>Running state</b>	It is the state a process is in when it is being executed. Processors with multiple cores can have the same number of processes in the running state as cores available.
7. <b>Finished state</b>	It is the state a process enters in when it is complete with its processing and is ready to be terminated gracefully. This normally occurs immediately after the Running State is complete.	22. <b>Serial Processing System</b>	Describes a system that can execute only one process (or one thread) at once.
8. <b>Kernel</b>	It is responsible for managing communications that occur between the computer system's hardware and the CPU manager.	23. <b>System call</b>	Used to coordinate activities with the kernel.
9. <b>Multicore Systems</b>	They are systems with multiple CPUs and may have two or more CPUs built in a single chip.	24. <b>Thread Control Block</b>	Holds data such as its unique thread identifier, thread state, pointer to the process that created it, pointer to any other threads that this threads created, priority, and so on. By using pointers, they are linked in the exact same way as processes are linked.
10. <b>Multithreading Systems</b>	A single process can have multiple threads; the threads are scheduled for execution and the results are reported back to the process that spun them off.	25. <b>Threads</b>	They are smaller entities of a process in some operating systems (threading or multithreading systems)
11. <b>Order of operations</b>	It is a mathematical standard that deciphers the complexities of multipart equations so they will always be solved in the correct order and thus result in only one solution, no matter who or what is doing the solving.	26. <b>Waiting state</b>	It is the state a process is in when it must wait for another part of the system to provide the necessary action so it can move back to the Ready State. In other words, this process has been sidelined until it gets an answer.
12. <b>Parallel Processing</b>	Perform different processes at the same time.		
13. <b>Parallel Processing System</b>	It is a system that has several CPUs, and all of them can be working simultaneously.		
14. <b>Pointer</b>	It is the name given to a piece of data that refers directly to a location in memory where a value can be found.		
15. <b>Process</b>	It is the key piece of code that undergoes execution (processing) by the CPU.		