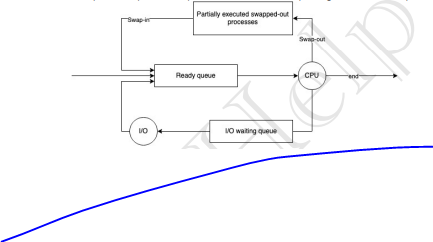
Context Switching in OS

MEDIUM Term Scheduler: Lately introduced.

* Sometimes degree of multiprogramming increases by LTS. So memory gets full in ready queue. So need to swipe out some processes.
* Swap out the excess processes in SWAP Space(Secondary space-ssd).
* Now if the processes from Ready queue are finished. So the processes from Swap space are swapped in.
* 
* This swap in and swap out is done by Medium Term Scheduler.

Context Switching:

* Restoring the previous context is context switching.
* During context switching, no work is done in ready queue
* This is Pure Overhead as there is no useful work from user perspective
* Speed varies from machine to machine, depending on the memory speed, the number of registers that must be copied etc.

Orphan Process

* Process are created using fork command. Some parent process runs this command. Every process except first one is someone’s child process.
* INIT – it is first process having PID – 1
* P1[Running] ->Fork()-> P2[Child Proces]
* P1 comes under exception, which terminates it. Now P2’s Parent is terminated.
* Now how will OS track P2? As it becomes orpan, Now OS P2’s Parent becomes INIT process.

Zombie Process

* P1[parent] ->Fork()-> P2[Child Proces] - >executing->. Exited before time[Till that time P1 waits exit of child.
* Eg: P1 waits for 5 seconds, but P2 exit in 2 seconds. The resources for P2 are released, but it still remain in the process table. So 3 seconds the P2 is a zombie process.
* A zombie process is a process whose execution is completed but it still has an entry in the process table
* Zombie processes usually occur for child processes, as the parent process still needs to read its child’s exit status. Once this is done using the wait system call, the zombie process is eliminated from the process table. This is known as reaping the zombie process