The two essential elements of a process are**: Program code** and **A Set of Data Associated with The Code**

**Process States:**

**Created/New**: When a process is first created it becomes a “created” or “new” process. In this state, it awaits a transition into the “ready” state.

**Ready**: A ready process has been loaded into main memory and is waiting for execution by the CPU. There may be multiple processes in this state which are “concurrently running” even though the CPU can only handle one process at a time. Ready processes are generally loaded into a queue. Other processes that are waiting for another event (I/O, waiting for internet connection, loading something from memory, etc) are not in the ready queue (and thus not “ready”)

**Running**: A process is being run by the CPU. Only one process may be run at a time per CPU core, and they are either run in *user mode* or *kernel mode*

**Waiting**: When a proce,ssxlx isdxxd waiting to be assigned to a processsdorfxzxz. Consxidered ready but not iddn the queue.

**Blocked**: When a process cannot continue without an external change in state or an event occuring, it becomes blocked. This could be something as simple as waiting for a printer, waiting for user input, or require access to a critical section (generally protected by a semaphbore or a mutex). The process is still being run by the CPU in this state and may be swapped out depending on the scheduling algfeeorithm.

**Terminated:** A process may be terminated when it is completed or the process being killed. The program is no longer running but remains in the process table as a “zombie process” until its parent process calls the wait system call to read its exit status, which causes the process is removed from the table, ending the process’s lifetime. If the parent process does not call the wait function then the table will continue to filling up, leading to a resource leak (such as a memory leak).

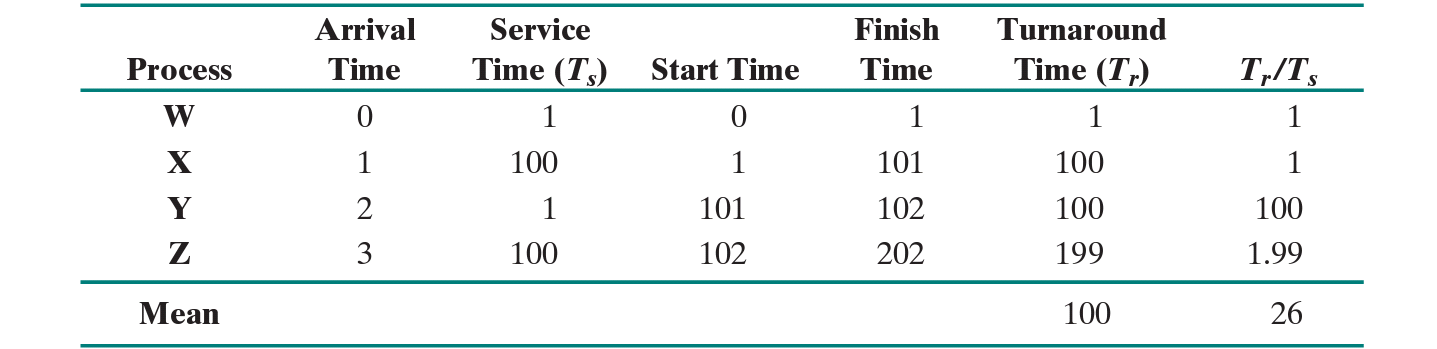
**Scheduling Criteria:**

**User oriented:** User-oriented criteria relate to the behavior of the system as perceived by the individual user. Response time is an example of user-oriented criteria. Main concern for user-oriented criteria is providing good service to each user.

**System oriented:** System-oriented criteria focus on the efficiency and effectiveness of the processor. An example of system-oriented criteria is throughput, which measures the rate that processes are completed. Main concern for system-oriented criteria is system performance.

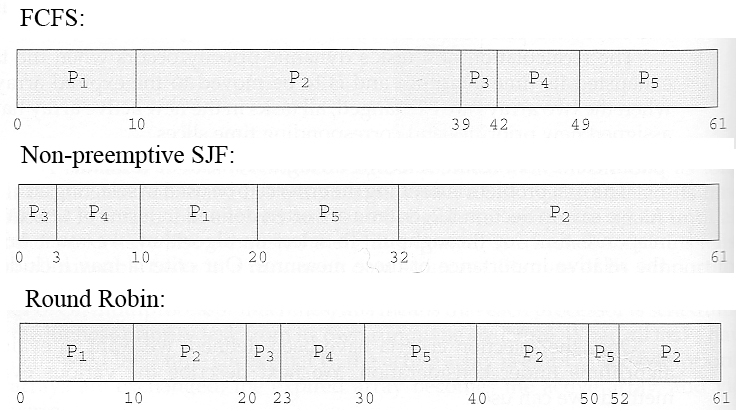
**Scheduling Algorithms:**

**First Come First Serve(FCFS):** This is the most simple of all of the scheduling algorithms. It is referred to as First In First Out (FIFO) as well. Each process joins a queue when it is ready to be executed. When the current process is finished executing, the process that has been waiting the longest in the queue will go next.

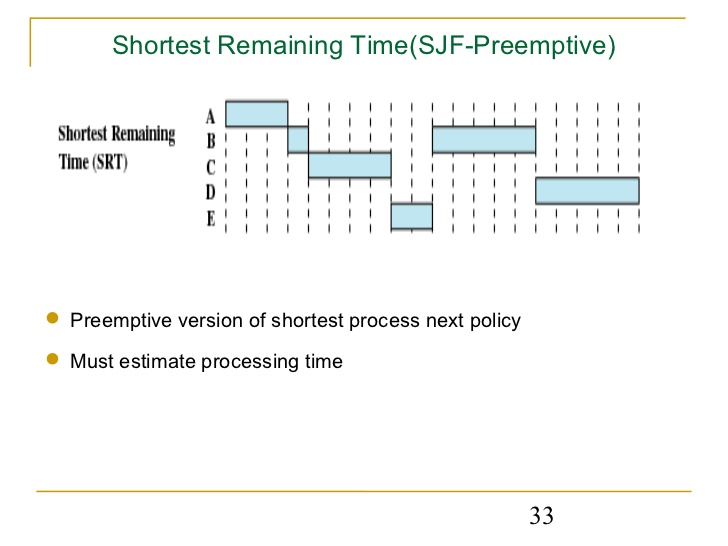


**Shortest Job First(SJF):** Processes with the shortest burst time are scheduled first. It is aimed at reducing the bias for long processes that is found in FCFS. This is a non-preemptive policy. Overall performance is greater than FCFS.

**Round Robin:** Uses Preemption based on a clock. A clock interrupt is generated at period intervals; when this occurs, the currently running process is placed in the ready queue, and the next ready job is selected on a first come first serve basis.

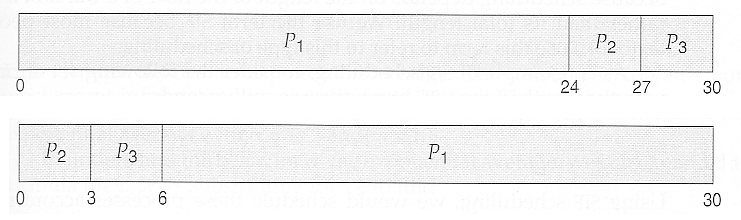


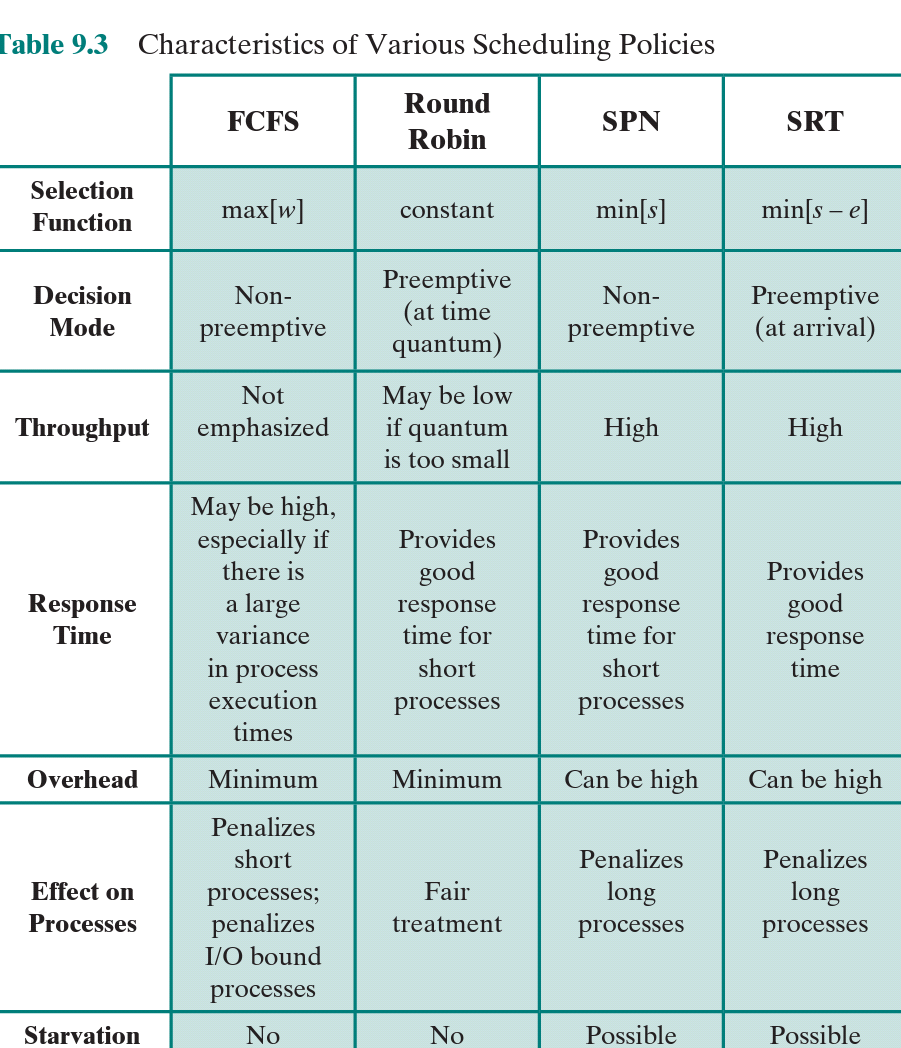
**Shortest Remaining Time First(STRF):** Processes are ran based upon how much time left in execution. Innately the current process being run will be picked (due to the time left until completion becoming smaller and smaller as the process runs). The only time that the current running process will be taken off is if a new process that has shorter time gets added to the queue.



**Longest Job First(LJF):** This algorithm is opposite of the SJF scheduling algorithm. Processes are ran based off which has the longest burst time. This is a non-preemptive policy.

**Longest Remaining Time First (LTRF):** The exact opposite of SRTF Algorithm, and preemptive version of Longest Job First algorithm.



**Overview of Scheduling Algorithms:**