OS Process Scheduler Simulator

The simulation works as follows: Various processes will arrive / spawn during the simulation. Each process has the following 4 parameters: 1) Arrival Time (AT) - The time at which a process arrives / is spawned / created. 2) Total CPU Time (TC) - Total duration of CPU time this process requires 3) CPU Burst (CB) – A parameter to define the upper limit of compute demand (further described below) 4) IO Burst (IO) - A parameter to define the upper limit of I/O time.

Initially when a process arrives at the system it is put into CREATED state. The processes' CPU and the IO bursts are statistically defined. When a process is scheduled (becomes RUNNING (transition 2)) the cpu_burst is defined as a random number between [1 .. CB]. If the remaining execution time is smaller than the cpu_burst compute, reduce it to the remaining execution time. When a process finishes its current cpu_burst (assuming it has not yet reached its total CPU time TC), it enters into a period of IO (aka BLOCKED) (transition 3) at which point the io_burst is defined as a random number between [1 .. IO]. If the previous CPU burst was not yet exhausted due to preemption (transition 5), then no new cpu_burst shall be computed yet in transition 2 and you continue with the remaining cpu burst. The scheduling algorithms to be simulated are: FCFS, LCFS, SRTF, RR (RoundRobin), PRIO (PriorityScheduler) and PREemptive PRIO (PREPRIO). In RR, PRIO and PREPRIO your program should accept the time quantum and for PRIO/PREPRIO optionally the number of priority levels maxprio as an input. We will test with multiple time quantums and maxprios, so do not make any assumption that it is a fixed number. The context switching overhead is "0".

The processes during its lifecycle will follow the following state diagram :