Exercise

We have the following system state. 3 processes P1, P2, P3. Length of P1, P2 and P3 is 12, 20, and 24 instructions respectively. There are 5 types of I/O operations – IO1, IO2, IO3, IO4, IO5.

P1 – I/O request for IO1 at its 6th instruction with a duration equivalent to execution of 10 instructions.
P2 - I/O request for IO4 and IO5 respectively at its 4th, and 14th instruction with a duration equivalent to execution of 7 instructions.
P3 - I/O request for IO1 to IO5 respectively every third instruction with a duration equivalent to execution of 9 instructions.

For the following three scenarios, using a 5-state process model, show the behavior of every process clearly indicating the state of ready and blocked queues. Assume there is a single blocked queue for all I/O. Ignore Dispatcher instructions.

1. A process gets CPU time in one go for its complete execution.
2. A process gets CPU time till it encounters an I/O operation.
3. A process gets CPU time needed to execute 5 instructions.

Repeat for above three scenarios considering that the dispatcher executes 3 instructions.

Solution

Time instance	Process Id	Process state	Instruction (CPU use)	IO used	Ready Q (left to right)	Blocked Q (left to right)

Remember

CPU and IO can go concurrently as the IO is carried by IO units and not the CPU, as was the case many years ago.

The five and seven state models have the blocked state to indicate that a process is waiting on a IO/event to occur, and that once it has been assigned that IO, the process continues with IO operation, and although progressing, it is not using the CPU and neither ready for CPU assignment till it completes the IO. In that case, the process though no more Blocked is neither Ready as well, and so its state should neither be shown Ready or Blocked in the above table. In the context of these state models, you can indicate as “IO exection”.

Diagramming Exercise (10 marks)

Construct a queuing model for a system following a 7-state process life cycle model implementing shared process state queues in a 2-processor environment with separate run state for each of the processors represented by Run-1 and Run-2 respectively.

Diagramming Exercise (15 marks)

Construct a queuing model for a system following a 7-state process life cycle model implementing dedicated process state queues in a single processor quad-core environment with separate run state for each of the processor cores represented by its respective run state.

Assumptions:

- Assume that every process is of length ***t*** time units (i.e., it requires ***t*** units of processor time).
- Every process has an I/O operation half-way of its execution.
- OS allows every process to run for ***t/2*** time units at a time.

Exercise 1

Queuing model for MLFQ scheduling policy supporting 5 levels, for an OS using 7-state process state-transition model, at maximum servicing only 2 IO devices along with 2 other events on which a process waits and that each IO device as well as the event have separate queues for the requesting processes.

Exercise 2

Suppose a system has three threads (T1, T2, and T3) that are all available to run at time 0 and need one, two, and three time units of processing respectively. Suppose that each thread is run to its total processing time before starting another. Draw different Gantt charts, one for each possible order the threads can be run in. For each chart, compute the turnaround time (TAT) of each thread; that is, the time elapsed from when it was ready (time 0) until it is complete. Also, compute the average TAT for each order. Which order has the shortest average TAT? What would you name the scheduling policy that produces the order having shortest average TAT?