Problem Solving Session

Scheduling Algorithms

Prepared by Dhruv Singhal

Least Used First (LUF)

Least Used First (LUF): When given the choice to schedule two processes on the run queue, the scheduler will select the one that has used the *fewest* CPU cycles thus far. In case of a tie, the queue is otherwise FIFO. When a process is descheduled it goes to the end of the queue.

Duration: amount of CPU time it will need, in time units.

Turnaround time: time between arrival and the time the job finishes.

Response time: time between arrival and the time the job is first scheduled.

Waiting time: total amount of time the job spent on the run queue waiting to be scheduled (i.e., for CPU-bound jobs it is turnaround time – duration time).

Assume all jobs are purely CPU-bound.

Question 1: Fill out this table

Supposing a running job is only preempted when a new job arrives (no interrupts), fill the following table:

Job	Arrival Time	Duration	Turnaround Time	Response Time	Waiting Time
A	0	25			
B	15	25			
C	25	5			
D	40	5			

Execution Timeline

Time	Events
0	Job A starts.
14, 15	Job A preempted (after 15 units). Job B starts.
24, 25	Job B preempted (after 10 units). Job C starts.
29, 30	Job C complete (after 5 units). Job B resumes.
39, 40	Job B preempted (after 10 units). Job D starts.
44, 45	Job D complete (after 5 units). Job A resumes.
54, 55	Job A completes (after 10 units). Job B resumes.
59	Job B completes (after 5 units).

Answers

Supposing a running job is only preempted when a new job arrives (no interrupts), fill the following table:

Job	Arrival Time	Duration	Turnaround Time	Response Time	Waiting Time
A	0	25	55	0	30
B	15	25	45	0	20
C	25	5	5	0	0
D	40	5	5	0	0

Question 2: Fill out this table

Suppose the clock interrupts the CPU every 2 time units (i.e., at times 0, 2, 4, etc.) and jobs arrivals do not preempt the current job. If a job arrives at the same time as a clock interrupt, the job can run immediately. Fill in the following:

Job	**Arrival Time**	**Duration**	**Turnaround Time**	**Response Time**	**Waiting Time**
A	0	25			
B	15	25			
C	25	5			
D	40	5			

Execution Timeline

Time	Events
0…15	Job A start and runs (16 units).
16…25	Job B starts and runs (10 units).
26…30	Job C starts, runs (5 units) and finishes.
31…37	Job B resumes and runs (7 units).
38…39	Job A resumes and runs (2 units).
40…44	Job D starts, runs (5 units) and finishes.
45	Jobs B resumes and runs (1 unit).
46…57	Jobs A and B alternate every 2 units.
58	Job A runs (1 unit) and finishes.
59	Job B runs (1 unit) and finishes.

Answers

Suppose the clock interrupts the CPU every 2 time units (i.e., at times 0, 2, 4, etc.) and jobs arrivals do not preempt the current job. If a job arrives at the same time as a clock interrupt, the job can run immediately. Fill in the following:

Job	Arrival Time	Duration	Turnaround Time	Response Time	Waiting Time
A	0	25	59	0	34
B	15	25	45	1	20
C	25	5	6	1	1
D	40	5	5	0	0

Question 3: Can the LUF policy cause starvation?

Explain why.

Question 3: Can the LUF policy cause starvation?

Yes.

A process that has been running for a long time will be starved from CPU cycles if a series of new processes arrives that only run for a short amount of time.