CSCE 451/851 Homework 3

Solutions

Problem 1 (40 points)

Consider the following set of processes:

Process Name	Arrival Time	Processing Time
A	0	3
B	1	5
C	3	2
D	9	5
E	12	5

Perform FCFS, RR $(q = 4)$, SPN, SRT, HRRN, Feedback $\left(q = 2^i\right)$ on them and get the Finish Time and Turn-around Time for each process. For reference:

- FCFS: First Come First Served
- RR: Round Robin
- SPN: Shortest Process Next
- SRT: Shortest Remaining Time Next

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
FCFS	A	A	A	B	B	B	B	B	C	C	D	D	D	D	D	E	E	E	E	E
RR	A	A	A	B	B	B	B	C	C	B	D	D	D	D	E	E	E	E	D	E
SPN	A	A	A	C	C	B	B	B	B	B	D	D	D	D	D	E	E	E	E	E
SRT	A	A	A	C	C	B	B	B	B	B	D	D	D	D	D	E	E	E	E	E

Scheduler		**A**	**B**	**C**	**D**	**E**	**Average**
	T_a	0.00	1.00	3.00	9.00	12.00	
	T_s	3.00	5.00	2.00	5.00	5.00	
FCFS	T_f	3.00	8.00	10.00	15.00	20.00	N/A
	T_r	3.00	7.00	7.00	6.00	8.00	6.20
	T_r/T_s	1.00	1.40	3.50	1.20	1.60	1.74
RR	T_f	3.00	10.00	9.00	19.00	20.00	N/A
	T_r	3.00	9.00	6.00	10.00	8.00	7.20
	T_r/T_s	1.00	1.80	3.00	2.00	1.60	1.88
SPN	T_f	3.00	10.00	5.00	15.00	20.00	N/A
	T_r	3.00	9.00	2.00	6.00	8.00	5.60
	T_r/T_s	1.00	1.80	1.00	1.20	1.60	1.32
SRT	T_f	3.00	10.00	5.00	15.00	20.00	N/A
	T_r	3.00	9.00	2.00	6.00	8.00	5.60
	T_r/T_s	1.00	1.80	1.00	1.20	1.60	1.32

Problem 2 (20 points)

Consider a computer with two processes, `H`, with high priority, and `L`, with low priority. The scheduling rules are such that `H` is run whenever it is in ready state. At a certain moment, with `L` in its critical region, `H` becomes ready to run (e.g., an I/O operation completes). `H` now begins busy waiting, but since `L` is never scheduled with `H` is running, `L` never gets the chance to leave its critical region, so `H` loops forever. This situation is sometimes referred to as the priority inversion problem.

If instead of priority scheduling, we use round-robin scheduling, will we encounter the same problem? Please explain your answer in detail.

Answer: With round-robin scheduling it works. Sooner or later `L` will run, and eventually it will leave its critical region. The point is, with priority scheduling, `L` never gets to run at all; with round robin, it gets a normal time slice periodically, so it has the chance to leave its critical region.

Problem 3 (40 points)

Based on measurements, we know for a certain system the average process runs for a time X before blocking on I/O. It takes a time Y to do a process switch, which is effectively wasted (overhead). For round-robin scheduling with quantum Q, give a formula for the CPU efficiency for each of the following:

a) $Q = \infty$
b) $Q > X$
c) $Y < Q < X$
d) $Q = Y$
e) $Q \longrightarrow 0, Q \neq 0$

NOTE: The CPU efficiency is the useful CPU time divided by the total CPU time.

The CPU efficiency is the useful CPU time divided by the total CPU time. When $Q \geq X$, the basic cycle is for the process to run for X and undergo a process switch for Y. Thus, a) and b) have an efficiency of $X/(X+Y)$. When the quantum is shorter than X, each run of X will require X/Q process switches, wasting a time YX/Q. The efficiency here is then

$$\frac{X}{X + \frac{YX}{Q}}$$

which reduces to $Q/(Q+Y)$, which is the answer to c). For d), we just substitute Q for Y and find that the efficiency is 50%. Finally, for e), as $Q \longrightarrow 0$, the efficiency goes to 0.