CS 162 Section 4: Deadlock and CPU Scheduling

True/False

1. Round robin scheduling provides a latency improvement over FCFS scheduling for interactive jobs.
   [True, since no process can hold the CPU longer than one quantum before a new job gets to run.]
2. The lottery scheduler prevents CPU starvation by assigning at least one ticket to each scheduled thread.
   [True.]
3. If the banker's algorithm finds that it's safe to allocate a resource to an existing thread, then all threads will eventually complete.
   [False, it only avoids resource-based deadlock. Threads can still infinite loop, etc.]

Short Answer

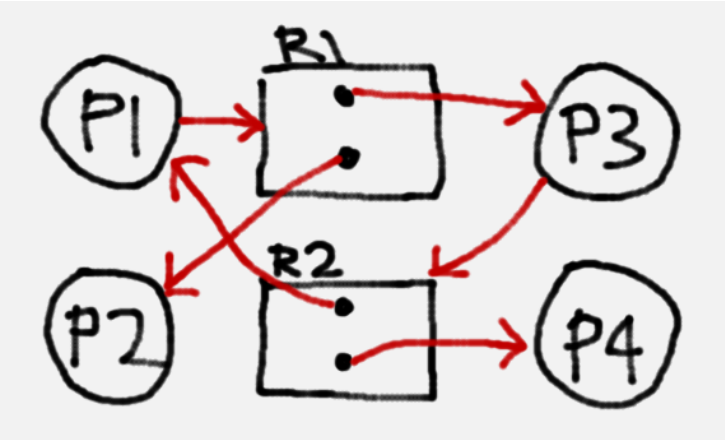
1. List the four requirements for deadlock.
   [mutual exclusion, hold and wait, no preemption, circular wait]

2. Give two ways to predict run-time for input to Shortest Remaining Time First (SRTF) scheduling
   1. Programmer provides best guess
   2. Use past behavior to model run time, e.g., keep an exponential average of each process's burst lengths
   3. Use multi-level feedback queues to automatically penalize longer running processes

Deadlock

Consider a system with four processes P1, P2, P3, and P4, and two resources, R1, and R2, respectively.
Each resource has two instances. Furthermore:
- P1 allocates an instance of R2, and requests an instance of R1;
- P2 allocates an instance of R1, and doesn't need any other resource;
- P3 allocates an instance of R1 and requires an instance of R2;
- P4 allocates an instance of R2, and doesn't need any other resource

1. Draw the resource allocation graph.



2. Is there a cycle in the graph? If yes name it.
   [Yes. P1 R1 P3 R2 P1 ...]
3. Is the system in deadlock? If yes, explain why. If not, give a possible sequence of executions after which every process completes.
   [No. There is a cycle, but no deadlock. P2 and P4 have all resources for completing. P2 P4, P1 P3.]

CPU scheduling exercise

Consider the following process arrival times, and run time requirements:

Process Name	Arrival Time	Running Time
A	0	3
B	1	5
C	3	2
D	9	2

For each scheduling algorithm, fill in the table with the process that is running on the CPU (for timeslice-based algorithms, assume a 1 unit timeslice). For RR and SRTF, assume that an arriving thread is run at the beginning of its arrival time, if the scheduling policy allows it. The turnaround time is defined as the time a process takes to complete after it arrives.

Time	FIFO	Round Robin	SRTF
0	A	A	A
1	A	B	A
2	A	A	A
3	B	C	C
4	B	B	C
5	B	A	B
6	B	C	B
7	B	B	B
8	C	B	B
9	C	D	B
10	D	B	D
11	D	D	D
avg. turnaround time	(3+7+7+3)/4 = 5	(6+10+4+3)/4 = 5.75	(3+9+2+3)/4 = 4.25