Brett Shwom
Operating Systems
Homework 3

1) If one carefully examines the most simple example of the dining philosophers problem (2 philosophers and 2 sets of chopsticks) one will notice that the following scenarios are possible. [Let's call the two philosophers P0 and P1, and the chopstick to the right of P0—or to the left of P2—C0 and the chopstick to the left of P0, C1.

1. P0 picks up C0, P0 picks up C1, P0 puts down the chopsticks
2. P0 picks up C1, P0 picks up C0, P0 puts down the chopsticks
3. P1 picks up C0, P1 picks up C1, P0 puts down the chopsticks
4. P1 picks up C1, P1 picks up C0, P0 puts down the chopsticks
5. P0 picks up C0, P1 waits for C0, P0 picks up C1, P0 puts down the chopsticks, P1 picks up C0
6. P0 picks up C1, P1 waits for C1, P0 picks up C0, P0 puts down the chopsticks, P1 picks up C1
7. P1 picks up C0, P0 waits for C0, P1 picks up C1, P1 puts down the chopsticks, P0 picks up C0
8. P1 picks up C1, P0 waits for C1, P1 picks up C0, P1 puts down the chopsticks, P0 picks up C1
9. P0 picks up C0, P1 picks up C1, P0 waits for C1, P1 waits for C0 --> DEADLOCK
10. P0 picks up C1, P1 picks up C0, P0 waits for C0, P1 waits for C1 --> DEADLOCK

Notice that deadlock occurs when both philosophers attempt to pick up chopsticks in the same direction (i.e. Both to their right, or both to their left). So in order to avoid deadlock, one philosopher needs to be reaching in a direction different from the other. If we apply this principle to a table with a greater number of philosophers, we notice that as long as one of the philosophers is reaching in a different direction, a deadlock cannot occur. As long as one philosopher is reaching in direction X (where X is either left or right) and the rest are reaching in direction Y (where Y is either left or right), and X!=Y, progress will be able to be made. Having one philosopher reaching in the 'opposite' direction ensures that at least one philosopher is waiting for a chopstick from a philosopher who is currently using two chopsticks (or if the philosopher next to him is not eating, he can simply pick up the chopstick!). The fact that a philosopher has 2 chopsticks is very important, because it implies that the 2 chopsticks will become available at some point for someone else to use, whereas if a philosopher has only 1 chopstick, there is no guarantee that progress can be made (i.e. That the philosopher will successfully obtain the chopstick that he's missing). So the proposed solution will work because it ensures that at least 1 philosopher will be picking up his chopsticks in a direction that is contrary to the direction of another philosopher.

2a) It would lower the turn around time of that process because it would allow for the process to be executed more frequently.
2b) Advantages: Lower turn around time.
Disadvantages: If a process finishes its work while another pointer to that process is still in the ready queue, it could be difficult to properly remove this reference. It could also increase the wait time for certain processes.
2c) Multiply the time quantum by 2

3a) First come first serve. High priority is given to processes when they are run. So the first process to be run has the highest priority, so it continues to run until it finishes. And then the next thread in the queue is executed. And so on...
3b) Round robin. High priority is given to processes when they are waiting. So after a process is run, it has a lower priority than the other processes that have been waiting in the queue. This is repeated in a cycle.