1. No, It enforces mutual exclusion but it violates the "no process not in its critical section can cause another process to block" constraint.
2. It certainly works with preemptive scheduling. However it could fail with non-preemptive scheduling. Consider the case in which *turn* is initially 0 but process 1   
   runs first. It will just loop forever and never release the CPU.
   1. If you run them through starting with the lowest time remaining, for each one running a minute, so at 8 minutes 2 will be done. Then going onto 4 so So then that finishes at 17. Then 23 28 and 31 for the remaining three. This is (8+17+23+23+31)/5 which equals 21.4 minutes.
   2. In Priority Scheduling the job with the highest priority is given the CPU. B has the largest Priority in this problem so it will execute first. Following this the rest of the order is E, A, C, and D. The time required for each process is: A: 24 min, B: 6 min, C: 26 min, D: 30 min, and E: 14 min. Therefore the total mean process turnaround time is (24+6+26+30+14)/5 = 20 min.
   3. First Come first serve is just each one runs after one another (10+16+18+22+30)/5 = 19 minutes for the mean process turnaround time.
   4. In shortest test-job first scheduling takes the jobs with the lowest CPU time first. The order for this will be C, D, B, E, A. the mean process turnaround time will then be (30+12+2+6+20)/5 = 14 min.
3. With this information we have 4 periodic events. P1 = 50msec, P2 = 100msec, P3 = 200msec, P4 = 250msec. And we have 4 CPU time events: C1 = 35 msec, C2 = 20msec, C3 = 10msec, C4 = x msec. The foruma follows: SUM(Pn/Cn) <= 1. Therefore: 35/50 + 20/100 + 10/200 + x/250 <= 1. From solving for x, we find that x <= 12.5 msec and we can infer that x = 12.5 msec is the largest possible value of x.
4. Lets say:

P1=False

P2=Fasle

P3 = False

P4= False

11012 <= 01021

No

01021 <= 01021

Yes, then

01010+01021=02031

P2 = true

11012 <= 02031

No

02031 <= 02031

Yes, then

02031 + 00001=02032

P3=true

11012 <= 02032

No

02110<=02032

No

P1 and P4 remain false.

There is a deadlock on P1 and P4