1. Which of the following is not one of the four cases when a preemptive CPU scheduler can take control?

   Upon making a function call

2. What is the conceptual difference between a dispatcher and a scheduler?

   The dispatcher switches to the process chosen by the scheduler

3. Which of the following correctly defines one of the scheduling criteria?

   Turnaround time is the amount of time from job submission to completion

4. Which of the following is true about shortest-job-first (SJF) scheduling?

   It selects the job with the shortest next CPU burst

5. Which of the following is true about first-come first-serve(FCFS) scheduling?

   Shorter jobs can suffer from the convoy effect

6. Which of the following is true about round-robin(RR) scheduling?

   For long time quantum, RR becomes similar to FCFS

7. What is one way of preventing starvation?

   Aging

8. Which of the following is true about multiprocessor scheduling?

   Asymmetric multiprocessing means multiple slave processors execute the schedule computed by one single master processor

9. which of the following is true about rate Monotonic scheduling, assuming no resource sharing?

   The priority depends on only the period but not the execution time of the task

10. Which of the following is true about earliest deadline first (EDF) vs. rate monotonic (RM) scheduling?

    RM tasks have static priority whereas EDF tasks have dynamic priority

11. Consider the following producer-consumer code, where the producer and consumer run concurrently and preemptively
    Which variable could have a race condition on most RISC-like processors?

```
// Producer
while (1) {
  nextItem = getItem();
  while(count==BUFSIZE) ;
  buffer[in] = nextItem;
  in = (in+1) % BUFSIZE;
  count++;
}
```

```
// Consumer
while (1) {
  while(count==0) ;
  item =buffer[out];
  out = (out+1) % BUFSIZE;
  count--;
}
```

    count

12. Continuing with the previous question, where can you put critical sections to eliminate the race condition?

    Around line 6 and another around line 12

13. which of the following is NOT a requirement of a critical section (CS)?

    After a process has requested to enter its CS, other processes are allowed to enter their CS for a finite number of times

14. Which of the following is true about Peterson's solution to the critical section problem?

    Mutual exclusion is achieved by the variable turn when flag[i] == flag[j] == TRUE

15. which of the following is true about atomic TestAndSet, which works on a Boolean variable named lock?

    To unlock, simply set lock = FALSE;

16. which of the following is true about wait(S) and signal(S), where s is a semaphore?

    Wait(S) may block, but signal(S) never blocks

17. Given the classical bounded n-buffer problem written using semaphores, what are the initial values for lines 1-3?

    1, 0, n

18. What values should be on lines 4, 5, 7, 8?

    empty, mutex, mutex, full

19. In the Readers-Writers classical synchronization problem, two semaphores named rw_mutex and mutex are declared. What are their purposes?

    rw_mutex allows at most one writer to block out all other processes or the first reader to block all writers

20. Which of the following is a correct usage of barrier synchronization implemented using a semaphore b, where S1 of process P1 should be executed before S2 of process P2?

    Process P2 should be written as wait(b); S2;

21. what is the difference between a deadlock and a livelock

    All processes in a deadlock blocked, whereas all processes in a livelock can all be Dispatched