Assignment 2

Student ID:2154312 Name:郑博远

1. If these two processes merely read the same file, no conflicts would occur; but if more than one process tries to alter a file at the same time, complications could arouse. So, the file manager would grant all of the requests of read access, but if one of them gain the write access, other requests will be denied.

2. (1) It takes $17 + 8 = 25$ ms to wait for the read/write head. So, $25 / 50 = 0.5$ time slices can be spent;

   (2) Given that 10 instructions can be executed each nanosecond($10 \times 10^6 = 10^7$ instructions can be executed each millisecond), $25 \times 10^7 = 2.5 \times 10^8$ instructions can be executed during the waiting period.

3. The I/O-bound process, which is the slower process. As a general rule of thumb, priority should be given to slower process. This is because after giving the priority to the I/O process, while waiting it to execute, time slices can be allocated to compute-bound process, which takes less time.

4. The throughput would be greater if one were I/O-bound and the other were compute-bound. This is because time slices can be given to the two processes alternately, so the I/O channel and the CPU can work simultaneously to save time. But if the two processes are both I/O-bound, it's hard for them to share time (unless they are going over different I/O channels).

5. a. The first requirement : there is a competition for nonshareable resources. The competition is removed because they are both allowed.

b. The first requirement : there is a competition for nonshareable resources. The competition is removed because they are both denied.

c. The third requirement : once a resource has been allocated, it cannot be forcibly retrieved. Their permissions have been forcibly retrieved.

d. The second requirement : the resources are requested on a partial basis; that is, having received some resources, a process will return later to request more. The student who has paid the fee don't have to request resource of permission.

6. a. Set a time limit. If the time a car has waited exceeds the limit, the green light will be given to the road which it's on. Another solution is that, the priority of the road goes up with the waiting time of the cars on it.

b. If a process of the highest priority has just completed a time slice, the next time slice will still be given to it because of its priority. So if this process takes lots of time to execute, other processes waiting may suffer from starvation. A solution is that the priority of processes which have been given time slices decreases, while the priority of processes waiting rises in time.