1a) False, Caching is used to improve I/O efficiency for files that are being written and re read rapidly. Buffering sotres data in memory while transferring between devices to cope with device speed and device transfer size mismatch.

SC2005 Operating Systems

1b) False. In a non-blocking I/O, the requesting process continues while the request is being processed by I/O device.

executing after the system call

TUTORIAL TWELVE

I/O Systems

1c) False. Device drivers are not part of the kernel I/O subsystem

1. Indicate whether the following statements are true or false. Justify your answer.

   a) Buffering can be used to improve I/O efficiency for files that are being written and re-read rapidly.

   b) Process will be in waiting state after performing an I/O system call if non-blocking I/O is used.

   c) Device drivers are part of the kernel I/O subsystem.

2a) An asynchronous I/O and double buffer can improve performance

2. Suppose that in a multiprogramming system, a process reads blocks of data from a file on disk for processing. As shown below, it reads one block of data at a time to a buffer using synchronous I/O and then processes the data.

```
while ( not end of file) {
        buffer <- read a block of data from disk using synchronous I/O;
        process data in buffer;
}
```

   a) Discuss how the performance of the above process can be improved.

   b) For a system running mainly with this type of processes, which file allocation scheme is best in terms of I/O performance?

Contiguous file Allocation scheme

Disk

3. a) During his presentation, a salesman emphasized on the substantial effort his company has made to improve the performance of their UNIX version - one example he quoted was that the disk driver used the SCAN algorithm and also queued multiple requests within a cylinder in sector order. You bought a copy and wrote a program to randomly read 10,000 blocks spread across the disk. The performance measured was the same as what would be expected from FCFS algorithm. Was the salesman lying?

   b) Under what circumstances could a disk scheduling discipline not improve the performance or even degrade performance of the system?

Not necessarily. If requests are issued one at a time, the disk driver has no opportunity for SCAN optimisation (SCAN =FCFS)

4. Assume that a disk drive has 200 cylinders, numbered 0 to 199. The disk head starts at cylinder 0. A seek takes (20 + 0.1×T) milliseconds, where T is the number of cylinders to move. Rotational latency is 2 milliseconds and data transfer per request takes 8 milliseconds, assuming each request accesses the same amount of data. The following table shows the arrival time and destination cylinder number of requests:

34.5 31 30.5

Arrive Time (ms)	0	15	20	23	30	35	50	65	70	88
Cylinder Number	45	132	35	4	23	50	70	40	10	35
	1	10	2	7	5	8	9	3	6	4

Compute the average time to service a request using the Shortest Seek Time First (SSTF) disk head scheduling algorithm.

3a) No, since the requests in the cylinder are queued in sector order, the disk arm can start from sector 0, and move from one end to the other end of the disk. This is how FIFO works. SCAN is FIFO but the disk arms moves to sector 0 first. So if the disk arm starts from sector 0, SCAN is the same as FIFO.

b) When there is heavy load, starvation might occur

Under light load conditions. If overhead for scheduling is significantly more than average seek time.

4)To read N sectors, positioning time + Data transfer time = seek time + rotational latency + transfer time = ((20+ 0.1*45) +2+8 + 31 +30.5 + 30.5 + 31.3 + 31.2 + 30.6 + 34.6 + 32 + 36.2)/10 = 32.24 s