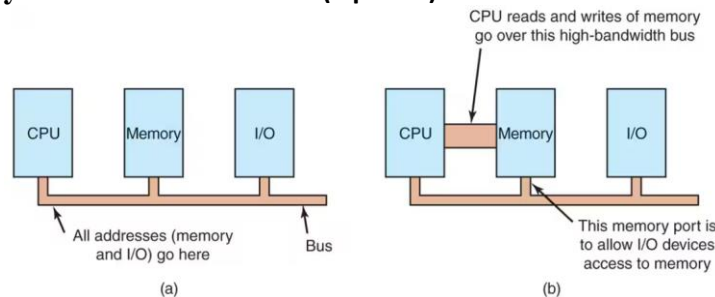


Assignment 4
CSC 4420 Computer Operating System
Chapter 5 and Chapter 6
Due: April 13, 2025
50 Points

1. Figure 5-3(b) shows one way of having memory-mapped I/O even in the presence of separate buses for memory and I/O devices, namely, to first try the memory bus and if that fails try the I/O bus. A clever computer science student has thought of an improvement on this idea: try both in parallel, to speed up the process of accessing I/O devices. What do you think of this idea? (5 points)



I do not think is a good idea because it adds additional complexity, power usage, and potential race conditions which outweigh the minor speed improvement. Instead sequential access (memory first, then I/O) remains more practical and reliable.

2. One mode that some DMA controllers use is to have the device controller send the word to the DMA controller, which then issues a second bus request to write to memory. How can this mode be used to perform memory to memory copy? Discuss any advantage or disadvantage of using this method instead of using the CPU to perform memory to memory copy. (5 Points)

To perform memory to memory copy:

- Controller has a source address where to read from -> Read a word from source memory into internal buffer
- Destination address where to write to -> Issue a second request to write the word to address
- Block size for bytes to transfer – then repeat process until the whole block is copied

Advantages using memory to memory copy: (instead of CPU)

- CPU will be free to execute other tasks (CPU offloading)
- Efficient for large transfers (reduces the number of CPU cycles)
- Less power consumption

Disadvantages using memory to memory copy: (instead of CPU)

- Slower because requires two bus transactions (one for read and one for write)
- Bus traffic (Bus overhead)
- Complex (handles sync, mem addr mngmt, buffer handling)

3. In which of the four I/O software layers is each of the following done? (5 points)

- a. Computing the track, sector, and head for a disk read.

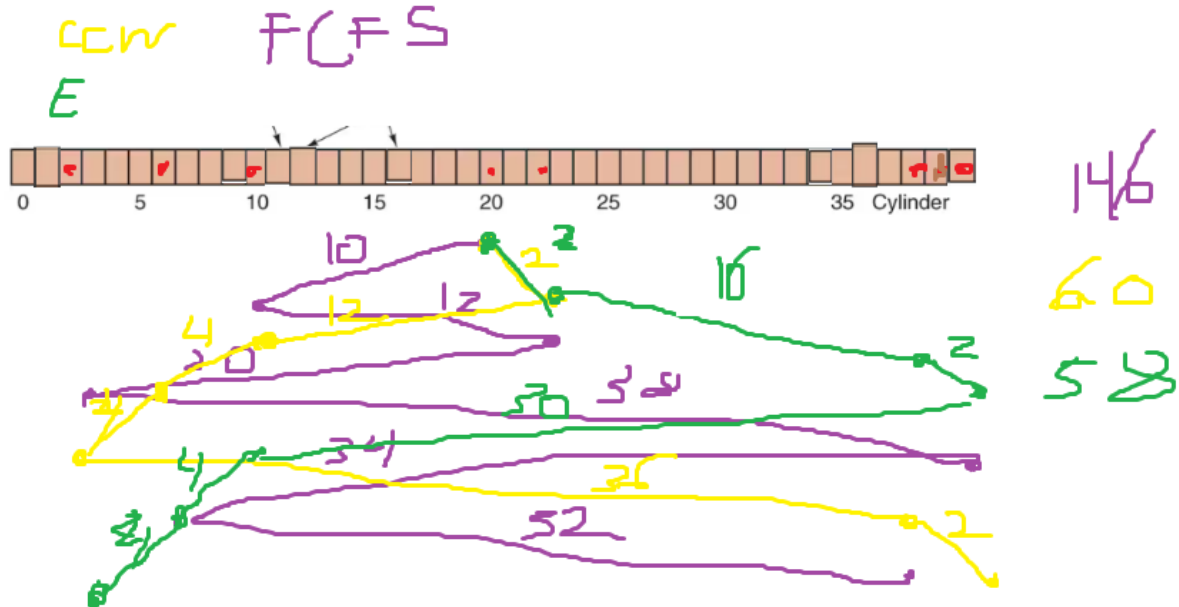
Device Driver

- b. Writing commands to the device registers.
Device Driver
- c. Checking to see if the user is permitted to use the device.
Device-Independent I/O Software
- d. Converting binary integers to ASCII for printing.
User-level I/O Software

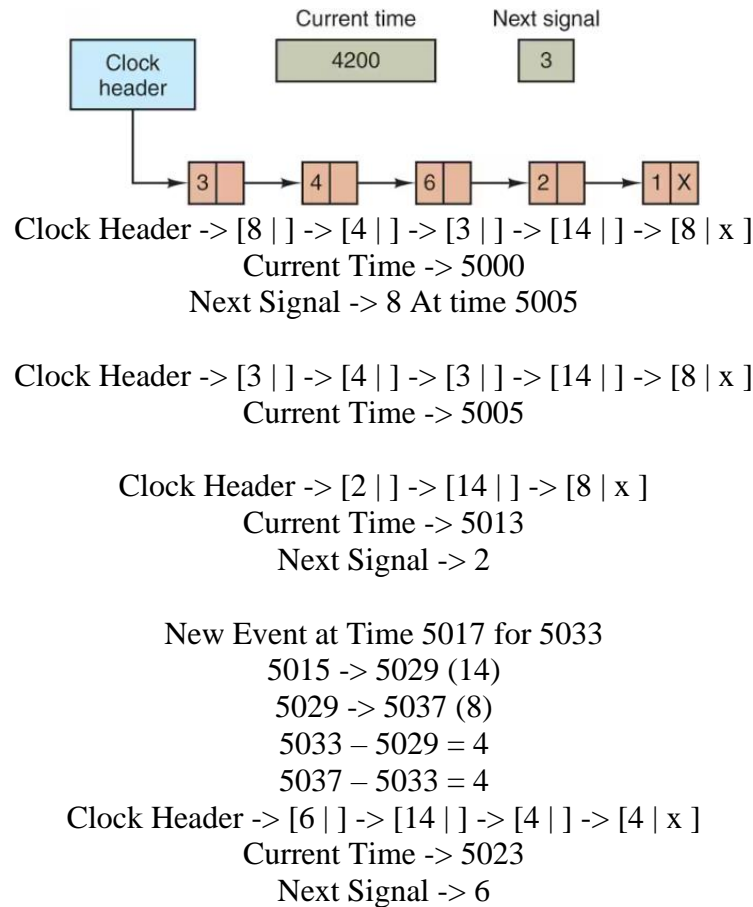
4. Disk requests come in to the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 6msec per cylinder. How much seek time is needed for

- a. First-come, first served.
146
- b. Closest cylinder next.
60
- c. Elevator algorithm (initially moving upward).
58

In all cases, the arm is initially at cylinder 20. (5 Points)



5. A system simulates multiple clocks by chaining all pending clock requests together as shown in Fig. 5-29. Suppose the current time is 5000 and there are pending clock requests for time 5008, 5012, 5015, 5029, and 5037. Show the values of Clock header, Current time, and Next signal at times 5000, 5005, and 5013. Suppose a new (pending) signal arrives at time 5017 for 5033. Show the values of Clock header, Current time and Next signal at time 5023. (5 points)



6. In the dining philosophers problem, let the following protocol be used: An even-numbered philosopher always picks up his left fork before picking up his right fork; an odd-numbered philosopher always picks up his right fork before picking up his left fork. Will this protocol guarantee deadlock-free operation? (5 points)

- The protocol guarantees deadlock-free operation because it removes circular waiting by enforcing an asymmetric fork acquisition strategy. But it does not prevent starvation. Some philosophers might eat significantly less frequently than others.

7. Students working at individual PCs in a computer laboratory send their files to be printed by a server that spools the files on its hard disk. Under what conditions may a deadlock occur if the disk space for the print spool is limited? How may the deadlock be avoided? which resources are preemptable and which are nonpreemptable? (5 points)

- A deadlock can occur in the print spooling system if the available disk space for spooling print jobs is limited and exhausted.

- To avoid deadlock we can limit job sizes, implement priority scheduling, cancel and remove the lowest priority job (preemption)

- Preemptable Resource -> Disk space (can delete partially stored job to free space)

- Nonpreemptable Resource -> Printer (once a job start printing, it cannot be stopped without corruption)

8. Consider the following state of a system with four processes, $P1$, $P2$, $P3$, and $P4$, and five types of resources, $RS1$, $RS2$, $RS3$, $RS4$, and $RS5$:

| | | | | | |
|-------|---|---|---|---|---|
| $C =$ | 0 | 1 | 1 | 1 | 2 |
| | 0 | 1 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 1 |
| | 2 | 1 | 0 | 0 | 0 |

| | | | | | |
|-------|---|---|---|---|---|
| $R =$ | 1 | 1 | 0 | 2 | 1 |
| | 0 | 1 | 0 | 2 | 1 |
| | 0 | 2 | 0 | 3 | 1 |
| | 0 | 2 | 1 | 1 | 0 |

$E = (24144)$
 $A = (01021)$

Using the deadlock detection algorithm described in Deadlock Detection with Multiple Resources of Each Type in the class, show that there is a deadlock in the system. Identify the processes that are deadlocked. (5 points)

Work = {0, 1, 0, 2, 1}

$P1$'s request = {1, 1, 0, 2, 1} -> Cannot be fulfilled because Work[0] = 0

$P2$'s request = {0, 1, 0, 2, 1} -> Can be fulfilled because Work = {0, 1, 0, 2, 1}

$P2$ Finishes request and releases allocated resources {0, 1, 0, 1, 0}

Work = Work + C($P2$) = {0, 2, 0, 3, 1}

$P3$ 'request = {0, 2, 0, 3, 1} -> Can be fulfilled because Work = {0, 2, 0, 3, 1}

$P3$ finishes and releases {0, 0, 0, 0, 1}

Work = Work + {0, 0, 0, 0, 1} = {0, 2, 0, 3, 2}

$P4$ request = {0, 2, 1, 1, 0} -> Cannot be fulfilled because Work[2] = 0

$P1$ request again = {1, 1, 0, 2, 1} -> Cannot be fulfilled because Work[0] = 0

$P1$ and $P4$ are still waiting and cannot finish therefore the deadlock processes are $P1$ and $P4$

9. A system has four processes and five allocatable resources. The current allocation and maximum needs are as follows:

| | Allocated | Maximum | Available |
|-----------|-----------|-----------|-----------|
| Process A | 1 0 2 1 1 | 1 1 2 1 3 | 0 0 x 1 1 |
| Process B | 2 0 1 1 0 | 2 2 2 1 0 | |
| Process C | 1 1 0 1 0 | 2 1 3 1 0 | |
| Process D | 1 1 1 1 0 | 1 1 2 2 1 | |

What is the smallest value of x for which this is a safe state? (5 points)

A M $Av = [0\ 0\ x\ 1\ 1]$ $N = M - A$
 [1 0 2 1 1] [1 1 2 1 3] [0 1 0 0 2]-> $P1$
 [2 0 1 1 0] [2 2 2 1 0] [0 2 1 0 0]-> $P2$
 [1 1 0 1 0] [2 1 3 1 0] [1 0 3 0 0]-> $P3$
 [1 1 1 1 0] [1 1 2 2 1] [0 0 1 1 1]-> $P4$

Applying Banker's Algorithm

P1 [0 1 0 0 2]

- Requires 1 unit of RS2 and 2 units of RS5
- Can be satisfied only if $x \geq 0$ and $Av[0] \geq 0$
- Not feasible yet because $Av[4] = 1 < 2$

P2[0 2 1 0 0]

- Requires 2 units of RS2 and 1 unit of RS3
- $Av[2] = x$ must be ≥ 1 and $Av[1] \geq 2$
- Not feasible because $Av[1] = 0 < 2$

P3 [1 0 3 0 0]

- Requires 1 unit of RS1 and 3 units of RS3
- $Av[0] = 0 < 1$ and $Av[2] = x < 3$
- Not feasible

P4 [0 0 1 1 1]

- Requires 1 unit of RS3, 1 unit of RS4, and 1 unit of RS5
- $Av[2] = x \geq 1$, $Av[3] = 1 \geq 1$, $Av[4] = 1 \geq 1$
- P4 can finish

Execute P4 and free resource -> P4 = [1 1 1 1 0]

Update $Av = [00\ x\ 11] + P4 = [11\ (x + 1)\ 2\ 1]$

P1 [0 1 0 0 2]

- $Av[1] = 1 \geq 1$ and $Av[4] = 1 < 2$
- Not feasible

P2 [0 2 1 0 0]

- $Av[1] = 1 < 2$
- Not feasible

P3 [1 0 3 0 0]

- $Av[0] = 1 \geq 1$ and $Av[2] = x+1 \geq 3$
- If $x \geq 2$, P3 can execute

The smallest value $x = 2$

- 10. Local Area Networks utilize a media access method called CSMA/CD, in which stations sharing a bus can sense the medium and detect transmissions as well as collisions. In the Ethernet protocol, stations requesting the shared channel do not transmit frames if they sense the medium is busy. When such transmission has terminated, waiting stations each transmit their frames. Two frames that are transmitted at the same time will collide. If stations immediately and repeatedly retransmit after collision detection, they will continue to collide indefinitely.**

- a. Is this a resource deadlock or a livelock?**

It is a livelock because occurs when processes are actively responding to a condition but never making progress (retransmitting frames)

- b. Can you suggest a solution to this anomaly?**

Exponential backoff breaks livelock by introducing randomness.

After collision stations wait for a random amount of time before attempting retransmission

c. **Can starvation occur with this scenario? (5 points)**

Yes because if a station keeps experiencing collisions while other stations successfully transmit, it might wait indefinitely to send its data.