SEM 1 - CAE 6123 - C N

Part - A (2M)

1. Computer architecture:

- The organization of the components which make up a computer system and the meaning of the operations which guide its function.
- The end-to-end structure of a computer system that determines how its components interact with each other in helping execute the machine's purpose (i.e., processing data)

2. Operations of Control Unit:

- fetches instructions from the CPU's memory,
- represented in bits,
- and translates those instructions into control signals in the form of pulses of electricity or light.
- OR
- Fetching instructions from memory.
- Decoding instructions to determine what operation to perform.
- Controlling and coordinating the execution of instructions.
- Managing data flow between various units of the computer.

3. Memory Register:

- In a computer, the memory address register (MAR) is the CPU register that either stores the memory address from which data will be fetched to the CPU registers, or the address to which data will be sent and stored via system bus.

4. Data Hazards:

- Data Hazards occur when an instruction depends on the result of previous instruction and that result of instruction has not yet been computed.
- whenever two different instructions use the same storage. the location must appear as if it is executed in sequential order.

5. Diff. HARD REAL TIME SYSTEM vs SOFT

HARD REAL TIME SYSTEM	SOFT REAL TIME SYSTEM
the size of data file is small or medium.	the size of data file is large.
response time is in millisecond.	response time are higher.
Peak load performance should be predictable.	peak load can be tolerated.
safety is critical.	safety is not critical.
A hard real time system is very restrictive.	A Soft real time system is less restrictive.

HARD REAL TIME SYSTEM	SOFT REAL TIME SYSTEM
In case of an error in a hard real time system, the computation is rolled back.	In case of an soft real time system, computation is rolled back to previously established a checkpoint.
Satellite launch, Railway signaling system etc.	DVD player, telephone switches, electronic games etc.
Guarantees response within a specific deadline.	Does not guarantee response within a specific deadline.
Catastrophic or severe consequences (e.g., loss of life or property damage).	Minor consequences (e.g., degraded performance or reduced quality).
Focused on processing critical tasks with high priority.	Focused on processing tasks with lower priority.
Highly predictable, with well-defined and deterministic behavior.	Less predictable, with behavior that may vary depending on system load or conditions.

6. Hit and Miss Ratios in Caches Memory:

- hit ratio:
  - It is a calculation of cache hits, and comparing them with how many total content requests were received.
- miss ratio:
  - It is the flip side of this where the cache misses are calculated and compared with the total number of content requests that were received.

7. Semaphore

- A semaphore is an integer variable, shared among multiple processes.
- The main aim of using a semaphore is process synchronization and access control for a common resource in a concurrent environment.
- The initial value of a semaphore depends on the problem at hand.
- Types:
  - Counting Semaphores
  - Binary Semaphores

Part - B

1.

```
a) Diagram the Functional Unit of the Digital Computer (16M)
```

https://www.geeksforgeeks.org/functional-components-of-a-computer/

```
b)
        1) Diff. Hardwired Control Unit vs Microprogrammed Control Unit
```

Hardwired Control Unit	**Microprogrammed Control Unit**
Generates the control signals needed for the processor using logic circuits	Generates the control signals with the help of micro instructions stored in control memory
faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares	slower than the other as micro instructions are used for generating signals here
Difficult to modify as the control signals that need to be generated are hard wired	Easy to modify as the modification need to be done only at the instruction level
More costlier as everything has to be realized in terms of logic gates	Less costlier than hardwired control as only micro instructions are used for generating control signals
It cannot handle complex instructions as the circuit design for it becomes complex	It can handle complex instructions
Only limited number of instructions are used due to the hardware implementation	Control signals for many instructions can be generated
Used in computer that makes use of Reduced Instruction Set Computers(RISC)	Used in computer that makes use of Complex Instruction Set Computers(CISC)

```
        2) Types of Control Hazards in computer networks:
```

In computer networks,

- control hazards refer to situations where,
  - the normal flow of control in a program is disrupted,
  - leading to potential issues or errors.
- Types of control hazards:
  1. Branch Prediction Errors:
     1. Branch prediction is a technique used by processors to guess the outcome of conditional branches in program execution.
     2. Control hazards occur when the prediction is incorrect, leading to wasted processing cycles.
  2. Control Flow Interruptions:
     1. These occur when the expected sequence of instructions is interrupted due to events such as interrupts, exceptions, or system calls.
     2. This can disrupt the normal flow of execution and lead to control hazards.
  3. Pipeline Stalls:
     1. In pipelined processors, control hazards can occur when instructions are fetched and executed out of order, leading to pipeline stalls or bubbles as the processor waits for the correct instruction to be fetched.
  4. Data Dependencies:
     1. Control hazards can also arise from data dependencies between instructions, such as when a conditional branch depends on the result of a preceding instruction that has not yet been executed.
     2. This can lead to delays in instruction execution and control hazards.

   5. Instruction Cache Misses:
      1. Control hazards can occur when instructions are not found in the instruction cache and must be fetched from main memory, leading to delays in instruction execution and potential control hazards.

Overall, control hazards in computer networks can lead to:

- inefficiencies,
- delays,
- errors in program execution,
- highlighting the importance of optimizing control flow
- minimizing disruptions in networked systems.

12.

```
b) Memory management requirements:
```

Memory management in computer networks refers to the processes and techniques used to efficiently allocate and manage memory resources across various network devices and systems. Here are some of the key memory management requirements in computer networks

1. **Resource Allocation**:
   1. Memory management involves allocating memory resources to different network functions, applications, and processes based on their requirements.
   2. This includes allocating memory for data buffers, packet queues, routing tables, and other network-related data structures.
2. **Dynamic Memory Allocation**:
   1. Network devices often need to dynamically allocate and deallocate memory resources to handle varying workloads and traffic patterns.
   2. Memory management systems must support dynamic memory allocation to efficiently utilize available memory and adapt to changing network conditions.
3. **Memory Protection**:
   1. In multi-user or multi-process environments, memory management systems must enforce memory protection mechanisms to prevent unauthorized access to memory regions.
   2. This includes enforcing access control policies and preventing memory corruption due to buffer overflows or other vulnerabilities.
4. **Memory Mapping**:
   1. Memory management systems may need to map physical memory addresses to logical addresses to facilitate communication between network devices and memory.
   2. Memory mapping techniques enable efficient data transfer between devices and memory without the need for costly memory copying operations.
5. **Buffer Management**:
   1. Network devices such as routers, switches, and network interface cards (NICs) often use buffers to temporarily store incoming and outgoing data packets.

    2. Memory management systems must efficiently manage these buffers to prevent buffer overflows, minimize packet loss, and ensure smooth data transmission.
6. **Memory Optimization**:
    1. Memory management systems should optimize memory usage to minimize memory fragmentation and maximize available memory capacity.
    2. Techniques such as memory pooling, garbage collection, and memory compaction can help optimize memory usage and improve overall system performance.
7. **Fault Tolerance**:
    1. Memory management systems should incorporate fault-tolerant mechanisms to ensure uninterrupted operation in the event of memory failures or errors.
    2. This may include redundancy, error correction codes (ECC), and error recovery mechanisms to detect and recover from memory faults.
8. **Scalability**:
    1. Memory management systems should be scalable to accommodate the growing memory requirements of modern network devices and applications.
    2. Scalable memory management solutions can efficiently handle increasing data volumes, traffic loads, and user demands without sacrificing performance or reliability.

Overall, effective memory management is essential for ensuring efficient operation, performance, and reliability of computer networks. By meeting these memory management requirements, network administrators can optimize memory usage, enhance system stability, and support the seamless operation of network infrastructure.

13.

```
b) Services of OS:
```

1. Program execution
2. Input Output Operations
3. Communication between Process
4. File Management
5. Memory Management
6. Process Management
7. Security and Privacy
8. Resource Management
9. User Interface
10. Networking
11. Error handling
12. Time Management

https://www.geeksforgeeks.org/operating-system-services/

Program Execution:

```
- It is the Operating System that manages how a program is going to be executed.
- It loads the program into the memory after which it is executed.
```

Input Output Operations:

```
- Manages the input-output operations and establishes communication between the user and device drivers.
- Device drivers are software that is associated with hardware that is being managed by the OS so that the sync between the devices works properly.
- It also provides access to input-output devices to a program when needed.
```

Communication between Processes:

```
- Communication between processes includes data transfer among them.
- If the processes are not on the same computer but connected through a computer network, then also their communication is managed by the Operating System itself.
```

File Management:

```
- helps in managing files also.
- If a program needs access to a file, it is the operating system that grants access.
- These permissions include read-only, read-write, etc.
- It also provides a platform for the user to create, and delete files.
```

Security and Privacy

```
- Security:
        - OS keep our computer safe from an unauthorized user by adding security layer to it.
        - Basically, Security is nothing but just a layer of protection which protect computer from bad guys like viruses and hackers.
        - OS provide us defenses like firewalls and anti-virus software and ensure good safety of computer and personal information.

- Privacy:
        - OS give us facility to keep our essential information hidden like having a lock on our door, where only you can enter and other are not allowed.
        - Basically , it respect our secrets and provide us facility to keep it safe.
```

User Interface:

```
- Users either interface with the operating system through the command-line interface or graphical user interface or GUI.
- The command interpreter executes the next user-specified command.
- A GUI offers the user a mouse-based window and menu system as an interface.
```

Networking

```
- This service enables communication between devices on a network, such as connecting to the internet, sending and receiving data packets, and managing network connections.
```

Error Handling

```
- The Operating System also handles the error occurring in the CPU, in Input-Output devices, etc.
- It also ensures that an error does not occur frequently and fixes the errors. It also prevents the process from coming to a deadlock.
```

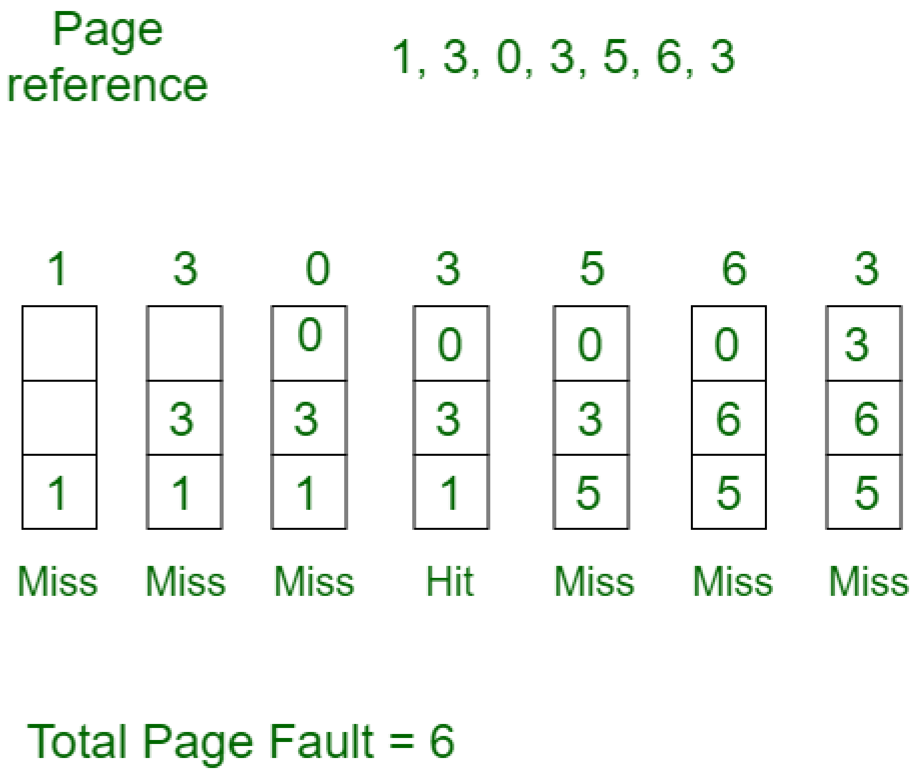
15.

```
b) Paper Reference String problem using 4 frames in computer networks
```

FIFO Stratergy

https://www.geeksforgeeks.org/reference-string-in-operating-system/

```
- Reference string is a concept that is used in memory management topics like page replacement algorithms.
- We can consider it as a sequence of numbers or sequence of strings .ie, a sequence of memory access requests while executing a program or any process.
- This request asks for a place in the program’s address space, which might have to access the data from the main memory.
```



A Scenario:

- Consider a process P, which has memory access requests in a sequence of 1,3,0,3,5,6,3. This memory request sequence is called a Reference String.

- 1 comes first, if 1 is in the frame, page hit occurs. Else request the page from main memory and add it into frames. Then 3 comes, 0 comes, 3 comes, 5 comes, 6 comes, 3 comes as in the sequence of Reference String.

Types:

- Sequential Reference String: Here strings follow a strict order like 1,2,3,4,5
- Random Reference String: Here the string will be in a non-predictable manner. That can be any random number like 1,6,3,9,5,1,…
- Looping Reference String: It follows a repetitive nature in the order of string elements like 1,2,3,1,2,3, …
- Localized Reference String: A localized reference string involves repeated requests for a specific memory location or a set of closely related memory locations like 5, 5, 5, 5, 5, 5.

Uses:

- It helps to evaluate an algorithm. In the case of the Page replacement algorithm, there is FIFO, Optimal, and LRU; we can compare the page hits, and page faults of each algorithm very easily.
- It could help to make a more optimized page table for different processes.
- We can predict the need for data requirements in the future for a particular process by analyzing the trend of the Reference String sequence.

Advantages:

- It helps to evaluate the efficiency of a page replacement algorithm.
- It helps to choose a better page replacement algorithm for a particular device or application.
- It helps to optimize memory and resource allocation.
- It could help the users to predict the future requirements of data.
- It could reveal the trend of memory requirement pattern.
- It helps to manage CPU cache memory.