CS621 Parallel and Distributed Computing

MID TERM CURRENT PAPER 2023

Solved by Masters

MCQS:

1. In serial computing, tasks are executed by the processor.

a. Simultaneously

b. randomly

c. in a sequence

d. in parallel

2. What is parallel computing architecture?

a. computing architecture where large problems are broken into smaller, independent parts.

b. A computing architecture where large problems are solved sequentially.

c. A computing architecture where large problems are solved by a single processor.

d. A computing architecture where large problems are solved by a single processor in sequential manner.

3. Which of the following is true for distributed computing?

a. All computers in the network must have the same operating system.

b. All computers in the network must be physically located in the same building

c. All computers in the network must have the same processing power.

D. Computers in the network can have different operating systems and processing power.

4. What is the main objective of fault tolerance in distributed computing?

a. To prevent all types of failures from occurring.

b. To ensure that the system can continue to operate despite failures.

c. To increase the speed of the system.

d. To decrease the cost of the system.

5. Before a program is written or a piece of software is developed, it must first go through a design process. For parallel and distributed programs, the design process will include the following issues except:

a. Decomposition

b. Communication

c. aggregation

d. Synchronization

6. _____refers to a type of parallel computing architecture where each processor has its own private memory, and communication between processors happens through message passing.

a. Sequential memory

b. Random Access memory

c. Distributed memory

d. Read only memory

7. In the context of distributed and parallel computing, synchronization between different processes or threads. Can be used to facilitate communication

a. direct memory access

b. shared memory

c. random Access memory

d. Read only memory

8. SISD stands for:

a. Single Instruction, Sequential Data

b. Single Instruction, Single Data
c. Sequential Instruction
d. Single Data read only memory

9. A personal computer processing instructions and data on single processor is an example of:

1. Single Instruction Single Data (SISD)
2. Single Instruction Multiple Data (SIMD)
3. Multiple Instruction Multiple Data (MIMD)

10. Sharing of computing resource among many users by means of multiprogramming and multi-tasking is the specific characteristic of:

a. Batch Era
b. Time Sharing Era
c. Desktop Era
d. Network Era

11. Evenly distribute network traffic to prevent failure caused by overloading a particular resource.

a. Locality
b. Scalability
c. Fault tolerance
d. Load balancing

12.The designates who does what. When multiple components of software are working on the same problem, they must be coordinated.

a. MBS(Memory Breakdown Structure)
b. MMS(Memory Merge Structure)
c. WBS(Work Breakdown Structure)
d. WMS(Work Merge Structure)

13: appear(s) in the form of a primary-backup protocol.

a. Primary-based protocols

b. Transmission control protocol

c. Replicated-write protocols

d. Address resolution protocol

14. _____ is the ability of a parallel and distributed computing system to continue executing tasks and completing jobs despite failures or disruptions.

a. Load balancing

b. Process resilience

c. Multithreading

d. Synchronization

15. The most common approach to failure masking is redundancy which is categorized into three types. Which one of the following options is not the category of redundancy?

a. Information redundancy

b. Time redundancy

c. Logical redundancy

d. Physical redundancy

16. _____ involves adding extra equipment or processes so that the system can tolerate the loss or malfunctioning of some components

a. Information redundancy

b. Time redundancy

c. Logical redundancy

d. Physical redundancy

17. _____ can lead to both the coordinator and the participants blocking, which may lead to the dreaded deadlock.

a. Single-phase commit protocol

b. Two-phase commit protocol

c. Three-phase commit protocol

d. Fourth-phase commit protocol

18. In techniques processes are assigned to the processors at compile time.

a. Dynamic mapping

b. Direct mapping

c. Cache mapping

d. Static mapping

19. In technique the process allocation cannot be changed during execution.

A. dynamic mapping

b. static mapping

c. direct mapping

d. cache mapping

20. _____mapping introduces non-determinism into the system.

a. Dynamic

b. Static

c. Direct

d. Cache

21. In centralized dynamic mapping whenever a process has no work, it takes a portion of available work from the ____

a. running process

b. slave process

c. block process

d. master process

22. If tasks are generated dynamically, then they must be mapped too.

a. topologically

b. statically

c. dynamically

d. demographically

23. ____-refers to the amount of data that can be transferred between memory and the processing units per unit time.

a. Bandwidth

b. Latency

c. Concurrency

d. Synchronization

24. _____is a mechanism that is used to control access to shared resources by enforcing exclusive and ordered access.

a. Locality of references

b. Latency

c. Synchronization

25. In a program is executed on multiple machines that communicate with each other over a network .

a. functional programming

b. parallel programming

c. distributed programming

d. procedural programming

26. In a single processing environment, concurrent tasks exist at the same time and execute within the same time period by

a. locality of references

b. latency

c. context switching

d. temporal locality

27. The development of the Memory Hierarchy occurred on a behaviour of a program known as

a. cache locality

b. spatial locality

c. temporal locality

d. locality of references

Subjective

Q1: There are several types of failures that can occur in parallel and distributed computing systems. You are required to mention the most common classification scheme for the failures. Also give the description for each failure type.

Answer:

The most common classification scheme for failures in parallel and distributed computing systems includes the following types:

1. **Fail-Stop Failures:** These occur when a component halts but doesn't lose any information. It can no longer perform any actions but can still respond to inquiries about its state.
2. **Byzantine Failures:** In these failures, a component behaves arbitrarily, sending incorrect or misleading information, which could potentially disrupt the system's correct functioning.
3. **Crash Failures:** A component stops functioning and does not recover. This type of failure results in the loss of any information stored in the failed component's volatile memory.

Q2. : How did the invention of desktop computing change the landscape of computing, and what were the key innovations of this era?

The invention of desktop computing made computing more accessible to individuals, revolutionized personal productivity through user-friendly interfaces and software applications, facilitated networking and connectivity, and led to hardware and software advancements, shaping the way people work and interact with technology.

Q3. : Flynn's classification of computer architectures is based upon the number of concurrent instruction (or control) and data streams available in the architecture. List down the names of these architectures.

1. **SISD (Single Instruction, Single Data)**
2. **SIMD (Single Instruction, Multiple Data**
3. **MISD (Multiple Instruction, Single Data**
4. **MIMD (Multiple Instruction, Multiple Data**

Q4: Write down the names of approaches used for the programming of concurrency in parallel and distributed computing

In parallel and distributed computing, several approaches are utilized for programming concurrency. Some of these approaches include:

1. **Message Passing:** Programs communicate by explicitly sending and receiving messages between separate processes or threads.
2. **Shared Memory:** Multiple threads or processes access shared memory locations to communicate and synchronize their actions.
3. **Task Parallelism:** Decomposing a task into subtasks that can be executed concurrently. These subtasks may run independently or communicate via message passing or shared memory.
4. **Data Parallelism:** Processing multiple data elements concurrently by distributing the data across multiple processors or cores. Each processor performs the same operation on its data set.

5. **Actor-Based Models:** Utilizes actors as independent units of computation that encapsulate both data and behavior, communicating via message passing.
6. **High-Level Parallel Constructs:** Libraries or language extensions (like OpenMP, MPI, CUDA, or Java's parallel libraries) that provide high-level constructs or APIs to simplify parallel and distributed programming.

Q5: Differentiate between serial and parallel computing

Answer:

Serial Computing:

1. **Sequential Execution:** In serial computing, tasks are executed one after another in a linear sequence.
2. **Single Processor:** It typically involves a single processor or core handling all computations.
3. **Limited Performance:** Performance gains are limited by the capabilities of a single processor.
4. **Shared Resources:** All tasks access shared resources sequentially, which can lead to potential bottlenecks.
5. **Simple Synchronization:** Synchronization is straightforward as tasks are executed in a single sequence.

Parallel Computing:

1. **Concurrent Execution:** In parallel computing, tasks are executed simultaneously, either by multiple processors or by breaking tasks into smaller, independent parts.
2. **Multiple Processors or Cores:** It involves multiple processors or cores working on different tasks concurrently.
3. **Enhanced Performance:** Parallel computing offers improved performance by leveraging multiple processors to execute tasks concurrently.
4. **Resource Distribution:** Tasks can utilize separate resources, reducing contention and potential bottlenecks.

5. **Complex Synchronization:** Requires synchronization mechanisms to coordinate and manage communication and data access between concurrent tasks or processors.

Note: Do check the answers by yourself too. If found any mistake correct accordingly.