## B.E. - COMPUTER SCIENCE AND ENGINEERING

### SIXTH SEMESTER
CIA-1
### SET B
### Answer key

1. 1.Nonblockingprimitive 2.Blockingprimitive 3. Asynchronousprimitive 4. Synchronousprimitive

2. An *asynchronous execution* is an execution in which
   There is no processor synchrony and there is no bound on the drift rate of processor clocks,
   Message delays (transmission + propagation times) are finite but unbounded, and
   There is no upper bound on the time taken by a process to execute a step.
   A *synchronous execution* is an execution in which
   (i)processors are synchronized and the clock drift rate between any two processors is bounded,
   (ii)message delivery (transmission + delivery) times are such that they occur in one logical step or round, and
   (iii)there is a known upper bound on the time taken by a process to execute a step.

3. a) Functional distribution  b) Load distribution/balancing  c) Replication of · processing power  d) Physical separation  e) Economics

4. The order of delivery of messages in a distributed system is an important aspect of system executions because it determines the messaging behaviour that can be expected by the distributed program.

5. (i)non-FIFO,(ii)FIFO,(iii)causalorder (iv) synchronousorder.

6. A.Evenifthestateofallthecomponentsisnotrecordedatthesameinstant,suchastatewillbemeaningfulprovidedeverymessagethatisrecorded asreceivedisalsorecordedassent.Basicideaisthatastateshouldnotviolatecausality—aneffectshouldnotbepresentwithoutitscause.Amessagecannotbereceivedifitwasnotsent.
Lamport invented a simple mechanism by which the happened before ordering can be captured numerically, called a *logical clock*.

A Lamport logical clock is a monotonically increasing software counter, whose value need bear no particular relationship to any physical clock. Each process $pi$ Keeps its own logical clock, $Li$ , which it uses to apply so- called *Lamport timestamps* to events. We denote the timestamp of event $e$ at $pi$ by $Li(e)$ , and by $L(e)$ we denote the timestamp of event $e$ at whatever process it occurred at.

7. A. When referring to a message without regard for the identity of the sender and receiver processes, we use $m^i$. For message $m^i$, its send and receive events are denoted as $s^i$ and $r^i$, respectively. More generally, send and receive events are denoted simply as s and r. When the relationship between the message and its send and receive events is to be stressed, we also use M, send(M) , and receive(M) respectively.
Severalorderingsonmessages havebeendefined
(i)non-FIFO,(ii)FIFO,(iii)causalorder (iv) synchronousorder.
b.

8. a.**Challenges**: 1.Heterogeneity  2.Openness  3.Security  4.Scalability 5.Failure handling 6.Concurrency 7.Transparency 8. Quality of service
**Design issues**: 1.Openness  2.Resource Sharing  3.Concurrency 4.Scalability5.Fault-Tolerance 6.Transparency 7.High-Performance

b.**Parallel system**: 1.process the data simultaneously, and increase the computational speed of a computer system. 2. Tasks are performed with a more speedy process.3. It is also known as a tightly coupled system 4. These systems share a memory, clock, and peripheral devices 5. In this, all processors share a single master clock for synchronization.
**Distributed system** :1.applications are running on multiple computers linked by communication lines. 2. consists of a number of computers that are connected and managed so that they share the job processing load among various computers distributed over the network 3.each processor has its own memory.  4. also known as loosely coupled systems. 4.These systems do not share memory or clock in contrast to parallel systems. 5. In this there is no global clock in distributed computing, it uses various synchronization algorithms.

**Dr.M.JEYAKUMAR**, M.E.,Ph.D.
**PRINCIPAL**
CHRIST THE KING ENGINEERING COLLEGE,
Chikkarampalayam Village,
Karamadai, Mettupalayam Taluk,
Coimbatore - 641 104.