Name: SWARNAVA CHAKRABORTY Date: 09-01-2021
Roll no.: 20
Dept.: CSE, Year: 4th
Subject Code: CS704A DISTRIBUTED OPERATING SYSTEM

1) i) ans:- (d)

1) ii) ans:- (b)

1) iii) ans:- (c)

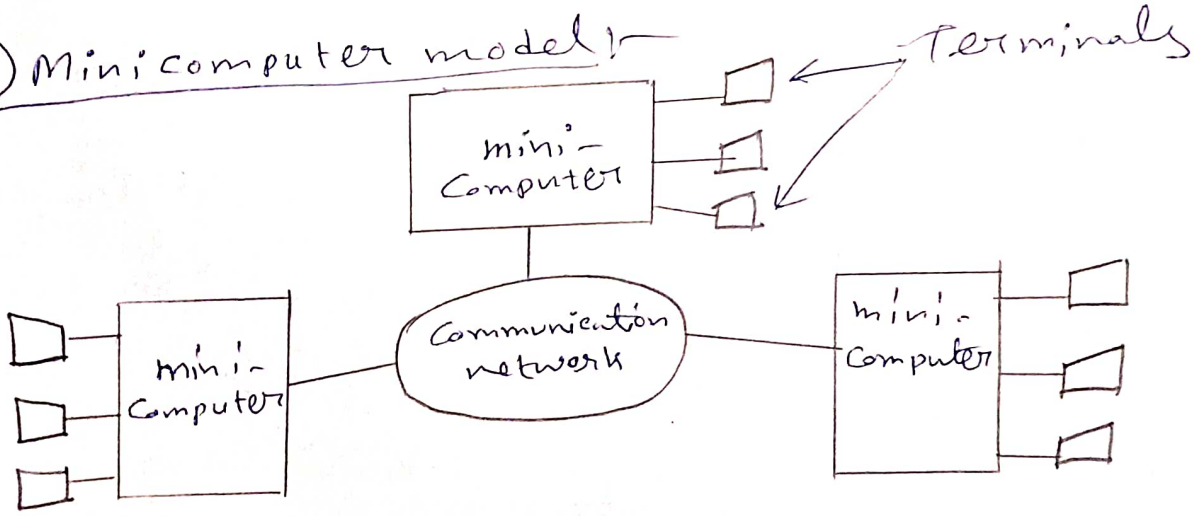
1) iv) ans:- (c)

1) v) ans:- (b)

(2) Tanenbaum and Renessa classified distributed systems into three broad categories :-

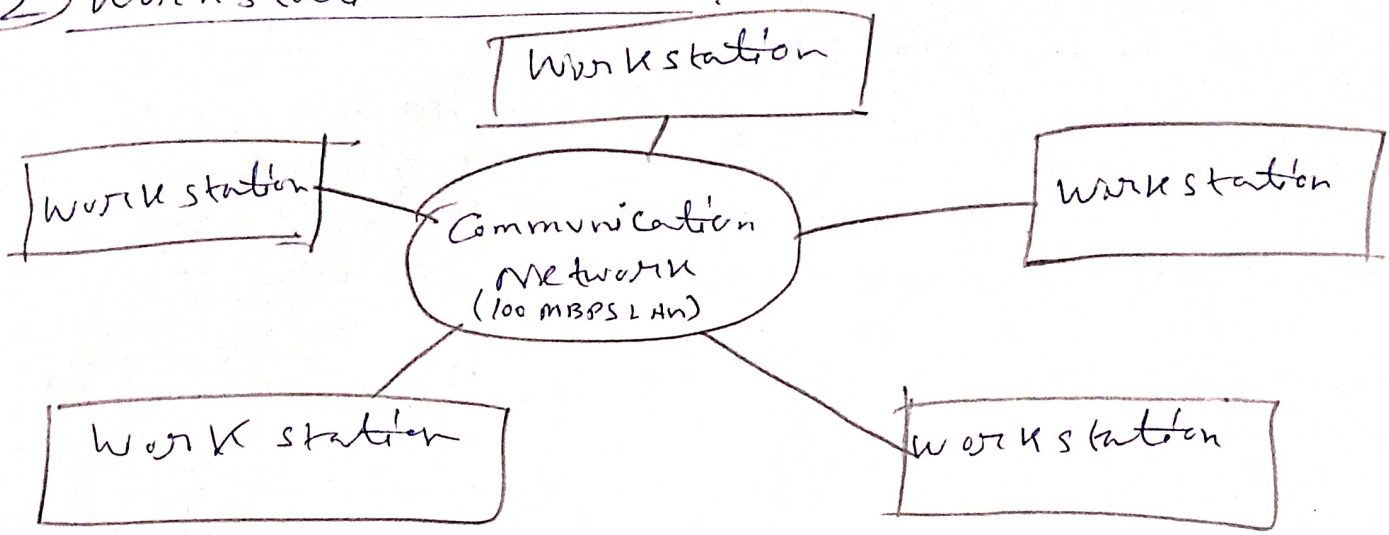
1) Minicomputer model
2) Workstation model
3) Processor Pool model

(1) Minicomputer model:-



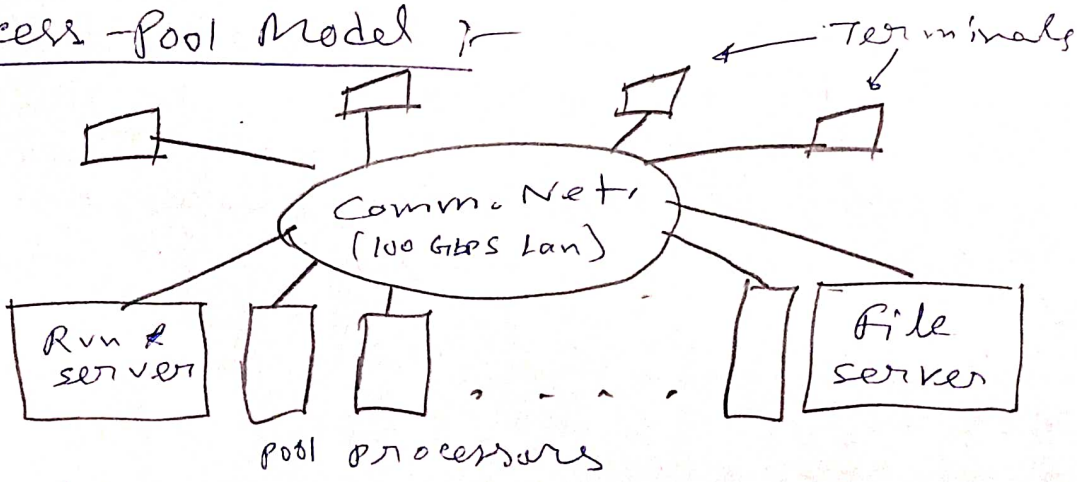
- This model is a simple extension of the centralized time-sharing system.
- Several interactive terminals are connected to each minicomputer. Each user logged on to one specific mini-computer has remote access to other mini-computer.
- The early ARPA net is an example of a distributed computing system based on the minicomputer model.

(2) Workstation Model :-



- ~~A distri~~
- This system based on the workstation model consists of several workstations interconnected by a communication network.
- In such environment, at any one time a significant proportion of the workstations are idle which results in the waste of large amounts of CPU time.
- Examples :- Sprite system & Xerox PARC.

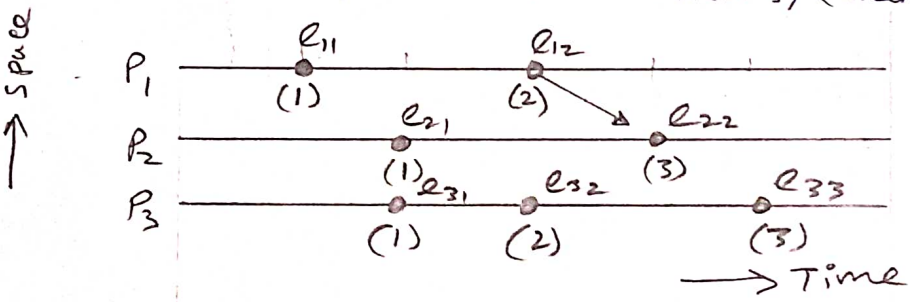
(3) Process-Pool Model :-



- This model is based on the observation that most of the time a user does not need any computing power.
- This model has better utilization of processing power & greater flexibility.
- Example :- Amoeba & the Cambridge Dist. Comp. Sys.

(3) Limitation of Lamport's clock :—

In lamport's system of logical clocks; if $a \rightarrow b$ then $C(a) < C(b)$. However, the reverse is not necessarily true if the events have occured in different processes. That is if a and b are events in different processes. That is if a and b are events in diff. process and $C(a) < C(b)$, then $a \rightarrow b$ is not necessarily true, events a & b may be causally related or may not be causally related. Thus Lamport's system of clocks is not powerful enough to capture such situation.



Space time diagram

From the above figure $C(e_{11}) < C(e_{22})$ and $C(e_{11}) < C(e_{32})$. Event e_{11} is causally related to event e_{22} but not to event e_{32}, since a path exists from e_{11} to e_{22} but not from e_{11} to e_{32}. Note that the initial clock values are assumed to be zero and d is assumed to equal 1. In other words in Lamport's system of clocks we can guarantee that if $C(a) < C(b)$ then $b \not\rightarrow a$, ~~however we can not say whether~~

So, The limitation is that each clock can independently advance due to the occurrence of local events in a process and the Lamport's clock system can not distinguish between the advancements of clocks due to

the ~~exch~~ local events from those due to the exchange of messages between processes.

Therefore using Lamport's clocks, we can not reason about the causal ~~rel~~ b/w 2 events occuring in different processes by just looking at the timestamps of the ~~events~~.

④ Issue in DOS :—

① Lack of Global knowledge :-

- communication delays are at the core of the problem.
- Information may become false before it can be acted upon..

② Naming :—

- named objects :- computers, users, files, printers
- namespace must be large
- logical to physical mapping needed.

③ Scalability :-

- broadcasting primitives, directories stored at every computer - these design will not work for large system

④ Process Synchronization :-

- test-&-set inst. won't work.
- Need all new sync. mech. for dist. sys.

⑤ Security :-

- Authentication : guaranteeing that an entity is what it claims to be

(6) Failure handling

- Computer systems sometimes fail
- fault occurs in h/w / s/w.
- Program may reduce.

(7) Structuring

(8) Res. Mgnt.

etc.