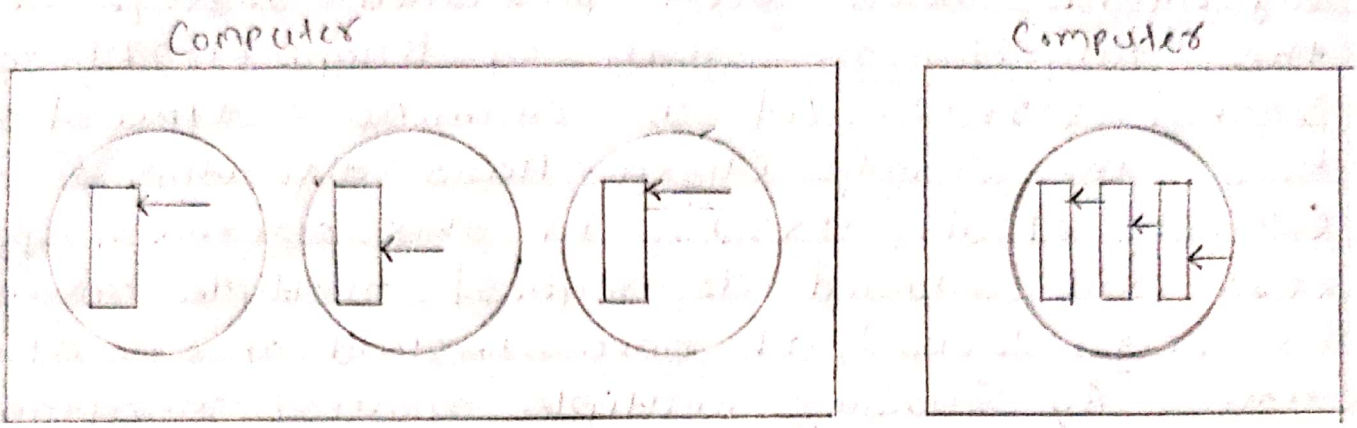
Threads :-



a) Three processes with one thread each
b) One process with three threads

CASE STUDY NO : 07

Date	/ /20
Page No.	35

Case Study On :- Study of processes and processors in distributed system. thread. system model

Processor allocation

Threads.....(cont) -

- All threads have exactly the same address space. They share code section, data section, and OS resources (open files & signals). They share the same global variables. One thread can read, write or even completely wipe out another thread's stack.

- Threads can be in any one of several states: running, blocked, ready or terminated.

- Team Model -

Is also a possibility here all threads are equal and each gets and processes its own requests. There is no dispatcher. Sometimes work comes in that a thread cand handle, especially if each thread is specialized to handle a particular kind of work. In this case a job queue is maintained with pending works kept in the job queue. With this organization a thread should check the job queue before looking in the system mailbox.

Advantages of using threads -

1. Useful for clients : If a client wants a file to be replicated on multiple servers, it can have one thread talk to each server.
2. Handle signals. Such as interrupts from the keyboard. Instead is de letting the signal interrupt the process, one thread is dedicated full time to waiting for signals.
3. Producer - consumer problems are easier to implemend using threads because threads can share a common buffer.
4. It is possible for threads in a single address space to run in parallel, on different CPUs.

* System Model -

- Processes runs on processor.
- In DOS with multiple processors, running a processor is a major design issue.
- The processors in DOS can be organized in different ways.
- Mainly they are work station model and processor pool model.

The workstation model :

The system consists of workstation scattered throughout a building or campus and connected by a high-speed LAN.

The systems in which workstation have

local disks are called diskful workstations. Otherwise, diskless workstations.

Hybrid Model -

- A possible compromise is to provide each user with a personal workstation and to have a processor pool in addition.
- For the hybrid model, even if you can not get any processor from the processor pool, at least you have the workstation to do the work.

Processor Allocation -

- Determine which process is assigned to which processor. Also called load distribution.
- Two categories :
- Static load distribution - nonmigratory, once allocated, can not move, no matter how overloaded the machine is
- Dynamic load distribution - migratory, can move even if the execution started. But algorithm is complex.

The goals of allocation

- Maximize CPU utilization.
- Minimize mean response time / Minimize response ration

Response ratio - the amount of time it takes to run a process on some machine, divided by how long it would take on some unloaded benchmark processor. Eg. a 1- sec. job that takes 5 sec. The ratio is 5/1.

Design issues for processor allocation algorithms-

- Determine versus heuristic algorithms.
- Centralized versus distributed algorithms.
- Optimal versus suboptimal algorithms.
- Local versus global algorithms.
- Sender-initiated versus receiver-initiated algorithms.

A large number of processor allocation have been proposed over the years. In this section we will look at some of the key choice involved in these algorithms and point out the various trade-offs. The major decisions the designers must make can be summed up in five issues:

Other decision also come up, but these are the main ones that have been studied extensively in the literature. Let us look at each of these in turn.