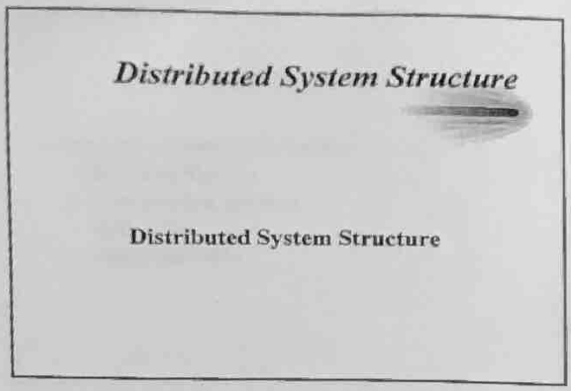
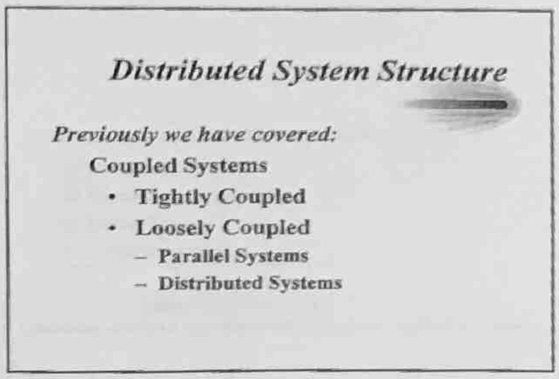


Distributed System Structure

Distributed System Structure

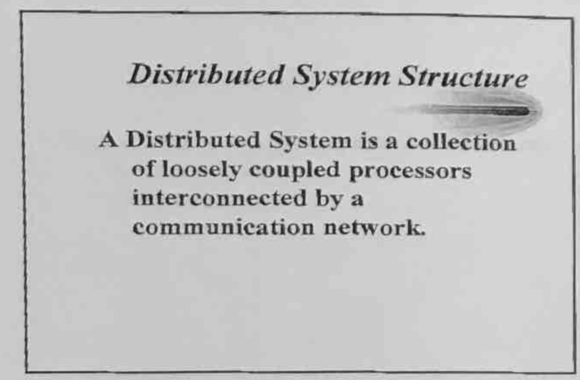


Distributed System Structure

Previously we have covered:

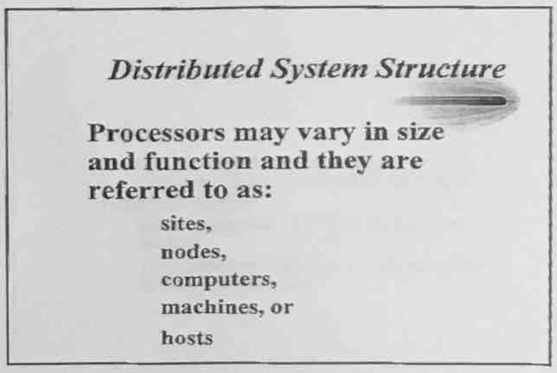
Coupled Systems

- Tightly Coupled
- Loosely Coupled
  - Parallel Systems
  - Distributed Systems



Distributed System Structure

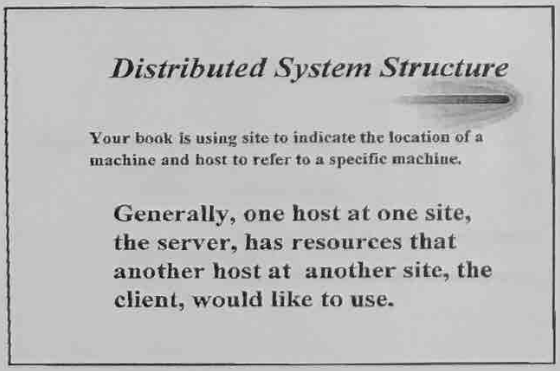
A Distributed System is a collection of loosely coupled processors interconnected by a communication network.



Distributed System Structure

Processors may vary in size and function and they are referred to as:

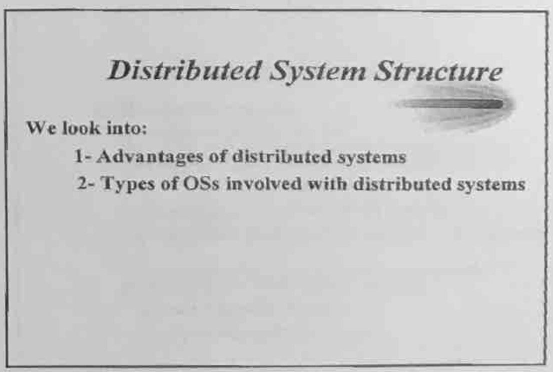
sites,
nodes,
computers,
machines, or
hosts



Distributed System Structure

Your book is using site to indicate the location of a machine and host to refer to a specific machine.

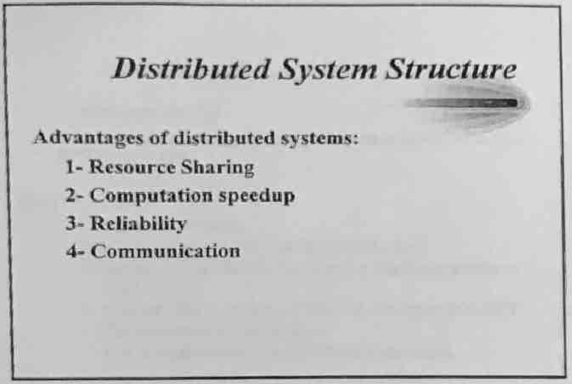
Generally, one host at one site, the server, has resources that another host at another site, the client, would like to use.



Distributed System Structure

We look into:

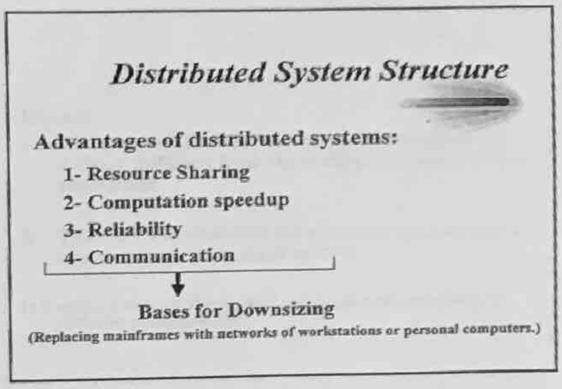
1- Advantages of distributed systems
2- Types of OSs involved with distributed systems



Distributed System Structure

Advantages of distributed systems:

1- Resource Sharing
2- Computation speedup
3- Reliability
4- Communication



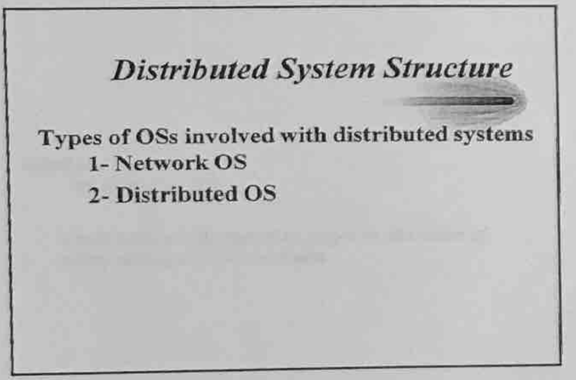
Distributed System Structure

Advantages of distributed systems:

1- Resource Sharing
2- Computation speedup
3- Reliability
4- Communication

Bases for Downsizing

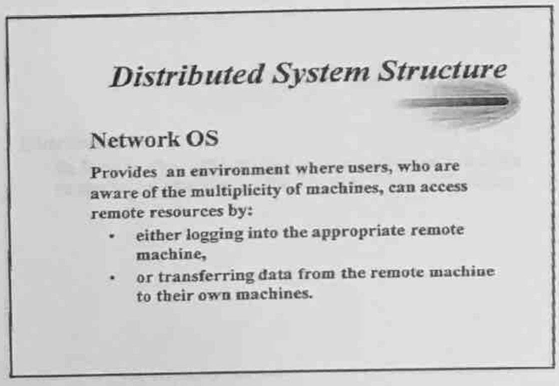
(Replacing mainframes with networks of workstations or personal computers.)



Distributed System Structure

Types of OSs involved with distributed systems

1- Network OS
2- Distributed OS

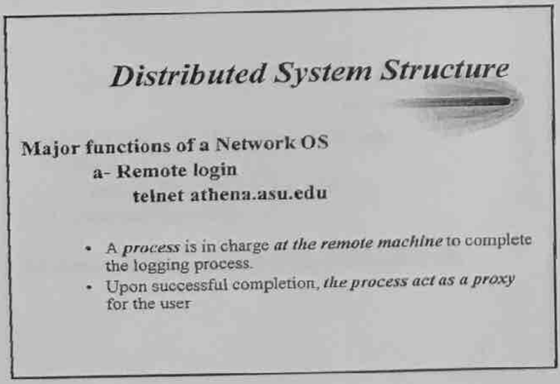


Distributed System Structure

Network OS

Provides an environment where users, who are aware of the multiplicity of machines, can access remote resources by:

- either logging into the appropriate remote machine,
- or transferring data from the remote machine to their own machines.



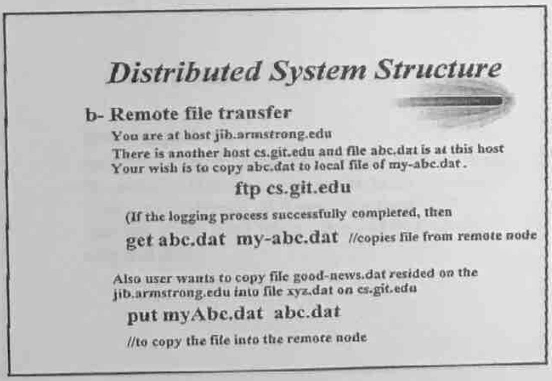
Distributed System Structure

Major functions of a Network OS

a- Remote login

telnet athena.asu.edu

- A *process* is in charge *at the remote machine* to complete the logging process.
- Upon successful completion, *the process act as a proxy* for the user



Distributed System Structure

b- Remote file transfer

You are at host jib.armstrong.edu
There is another host cs.git.edu and file abc.dat is at this host
Your wish is to copy abc.dat to local file of my-abc.dat .

ftp cs.git.edu

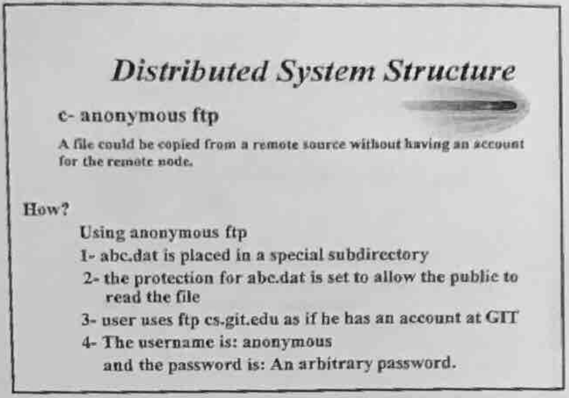
(If the logging process successfully completed, then

get abc.dat my-abc.dat //copies file from remote node

Also user wants to copy file good-news.dat resided on the jib.armstrong.edu into file xyz.dat on cs.git.edu

put myAbc.dat abc.dat

//to copy the file into the remote node



Distributed System Structure

c- anonymous ftp

A file could be copied from a remote source without having an account for the remote node.

How?

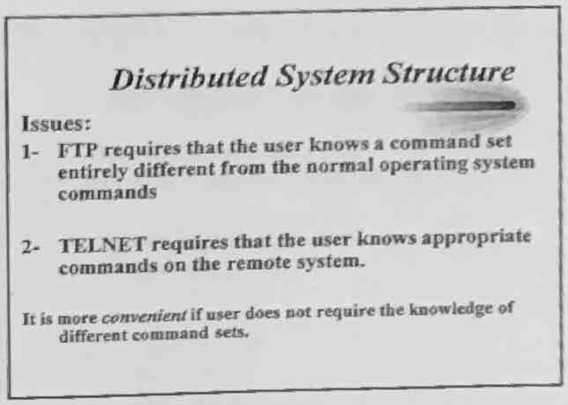
Using anonymous ftp

1- abc.dat is placed in a special subdirectory

2- the protection for abc.dat is set to allow the public to read the file

3- user uses ftp cs.git.edu as if he has an account at GIT

4- The username is: anonymous
and the password is: An arbitrary password.



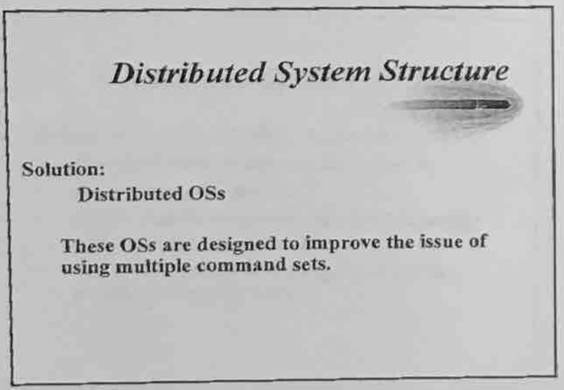
Distributed System Structure

Issues:

1- FTP requires that the user knows a command set entirely different from the normal operating system commands

2- TELNET requires that the user knows appropriate commands on the remote system.

It is more *convenient* if user does not require the knowledge of different command sets.

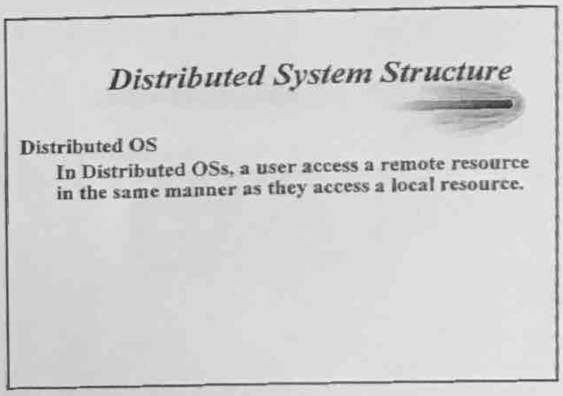


Distributed System Structure

Solution:

Distributed OSs

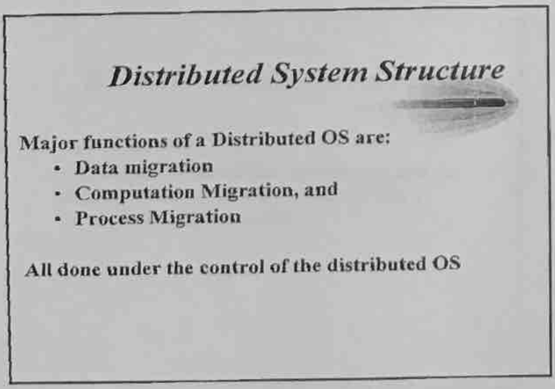
These OSs are designed to improve the issue of using multiple command sets.



Distributed System Structure

Distributed OS

In Distributed OSs, a user access a remote resource in the same manner as they access a local resource.

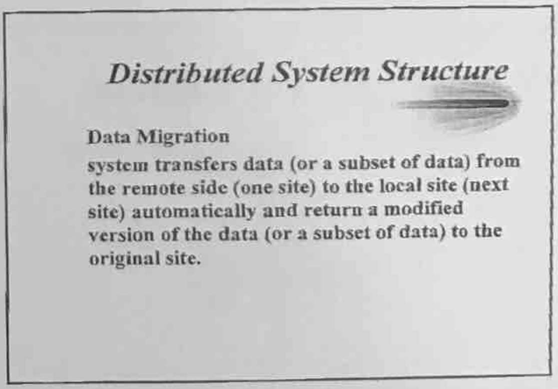


Distributed System Structure

Major functions of a Distributed OS are:

- Data migration
- Computation Migration, and
- Process Migration

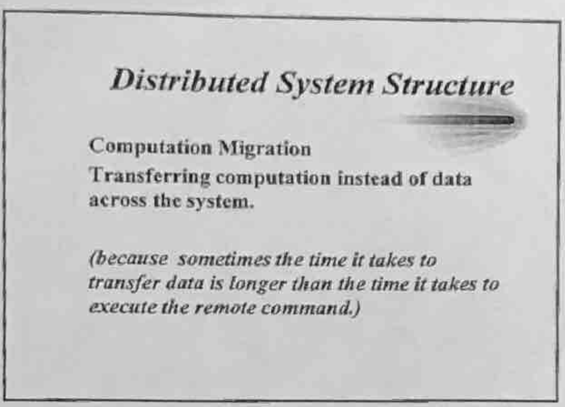
All done under the control of the distributed OS



Distributed System Structure

Data Migration

system transfers data (or a subset of data) from the remote side (one site) to the local site (next site) automatically and return a modified version of the data (or a subset of data) to the original site.

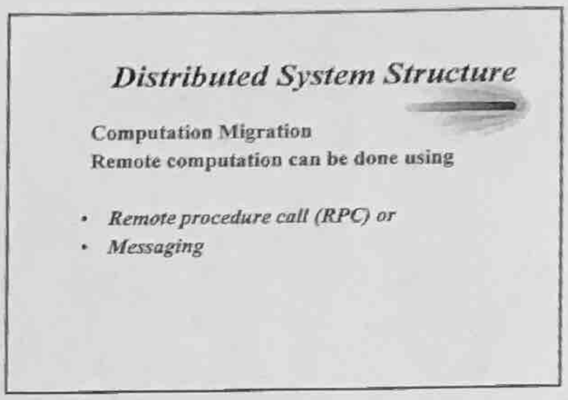


Distributed System Structure

Computation Migration

Transferring computation instead of data across the system.

(because sometimes the time it takes to transfer data is longer than the time it takes to execute the remote command.)

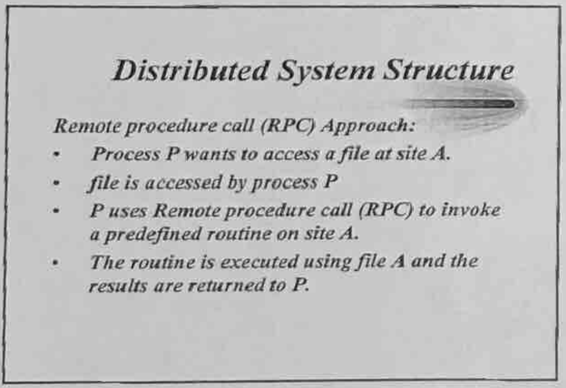


Distributed System Structure

Computation Migration

Remote computation can be done using

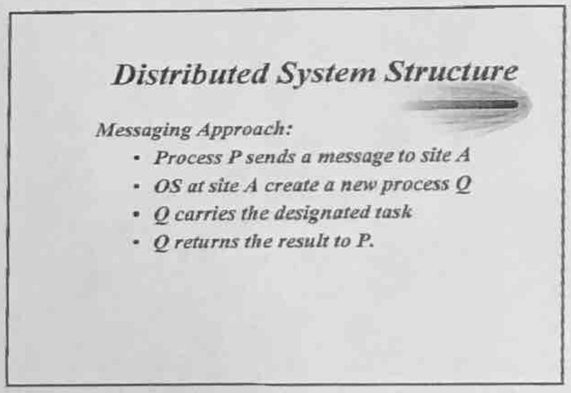
- ***Remote procedure call (RPC) or***
- ***Messaging***



Distributed System Structure

Remote procedure call (RPC) Approach:

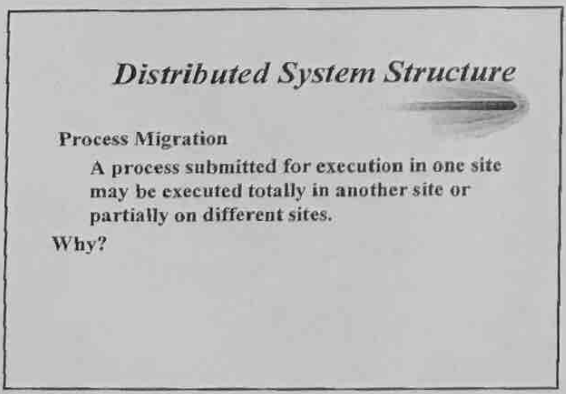
- *Process P wants to access a file at site A.*
- *file is accessed by process P*
- *P uses Remote procedure call (RPC) to invoke a predefined routine on site A.*
- *The routine is executed using file A and the results are returned to P.*



Distributed System Structure

Messaging Approach:

- *Process P sends a message to site A*
- *OS at site A create a new process Q*
- *Q carries the designated task*
- *Q returns the result to P.*

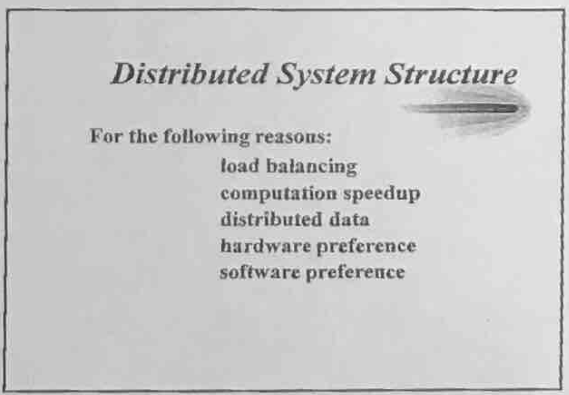


Distributed System Structure

Process Migration

A process submitted for execution in one site may be executed totally in another site or partially on different sites.

Why?



Distributed System Structure

For the following reasons:

load balancing
computation speedup
distributed data
hardware preference
software preference