

# DISTRIBUTED SYSTEMS

\* need a brief look at multicomputer systems for some basic concepts  
[cluster computing]

each node has processor, memory,  
network interface

→ hundreds or thousands can be  
interconnected

- "world's most powerful computers"

\* interconnection technology  
+ topology

\* communication based on message-  
passing

I send(dest, &mptr)

receive(src, &mptr)

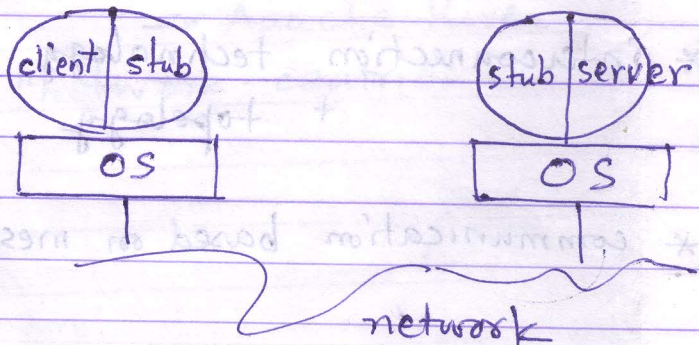
blocking  
versus

non-blocking calls

II

RPC (remote procedure call)

(Fig. 8-21)



how can a "procedure call"  
- both work across two different  
address spaces??

see "implementation issues" if  
you are interested  
(p. 539)

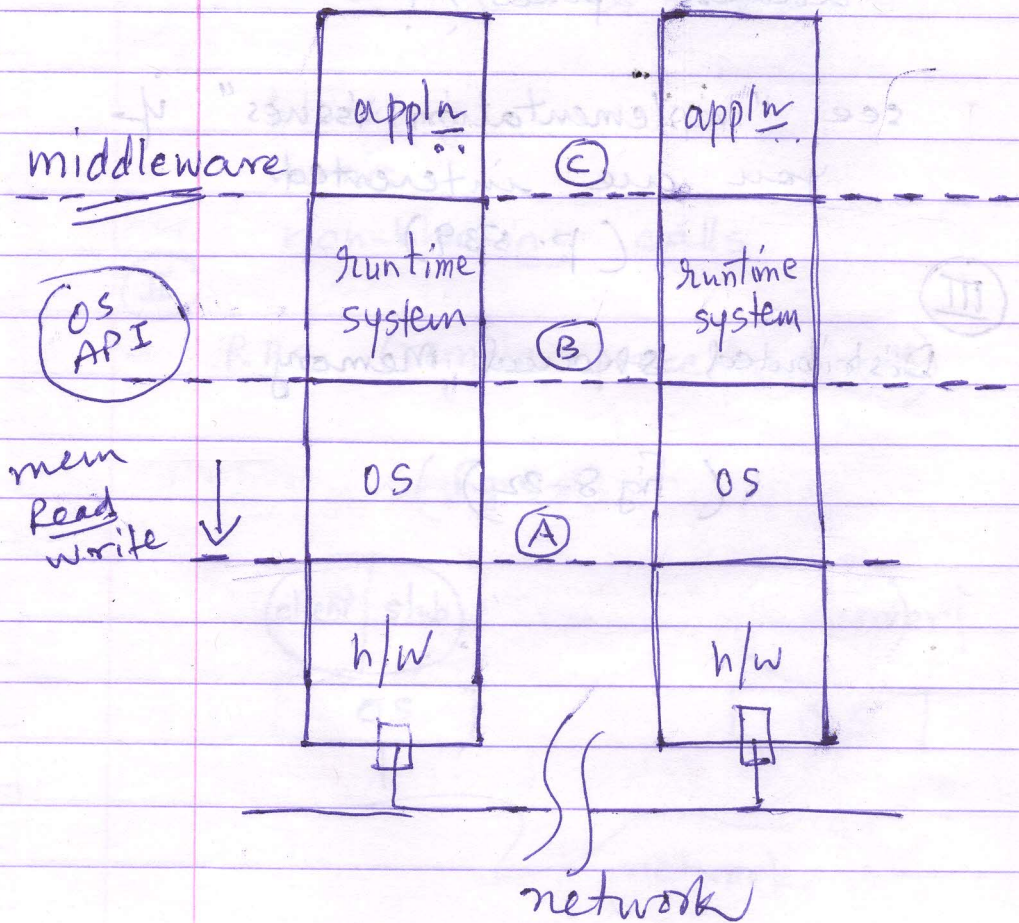
III

Distributed shared memory

(fig 8-22)



Various layers where shared memory can be implemented -



pp: 549-550

Introduction to Distributed Systems

see Fig. 8-27

Comparison of three kinds  
of multiple CPU systems

### 8.3.3 Document-based Middleware

web pages

hyper-links

URLs

DNS

- to access a web page }
1. browser requests DNS for IP address
  2. DNS replies with IP address
  3. browser makes TCP conn to port
  4. " requests for file
  5. server sends " " <sup>html</sup>
  6. TCP conn is closed
  7. browser displays text
  8. " " images

### 8.3.4 File-system based m/w

"single global file system"

modeled as [ upload/download  
remote access

Fig 8-34

"mounting"

Directory hierarchy

- meaning of "mounting" a remote f/s on to the local directory hierarchy

Fig 8-35

- do all users see the same global directory hierarchy?

Naming transparency -

Location independence

(stronger condition)



"semantics" of file sharing

sequential consistency - all processes  
reading/writing  
a file see the  
same ordering  
of reads & writes

for efficiency, we need to  
replicate files. But then  
sequential consistency becomes  
difficult to achieve. Impact  
on efficiency.

session semantics - changes to an  
open file seen only by resp. process.  
When file is closed, changes seen by  
other processes

"... getting the semantics of  
(distributed) shared files right  
is a tricky business with no  
elegant and efficient solutions"

Dijkstra

Hoare

Per Brinch Hansen

MULTICS → UNIX → MINIX

→ LINUX



Internet

(TCP/IP)



### 8.3.5 Shared object-based m/w

"language-level" objects

→ "run-time objects"

CORBA [Common Object Request-  
Broker Architecture]

- \* references to objects which have been created may be stored in a directory
- \* client-server architecture with RPC type mechanism
- \* lack of scalability
- poor performance on large scale

Globe - distributed object system ...

designed to scale to billions of  
nodes and billions of objects

### 8.3.6 Coordination-based m/w

Linda (mid 1980s) tuple space

To the user, the tuple space  
looks like a big, global  
shared memory -----

----- tuples are pure data; ----- do  
not have associated methods

Sun Microsystems → Jini

→ Apache River

"network-centric"