OS Basics — Previously

We have covered all the fundamental OS components:

- Architecture and OS interactions
- Asynchronous concurrent processes
- Synchronization and deadlock
- Process scheduling
- Memory management
- File systems and I/O

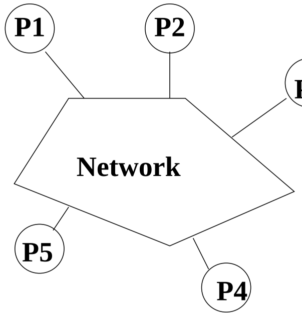
Distributed Systems — Today

Technology trends:

decade	technology	$$ per machine	sales	users per machine
50's	custom	10M	100	1000s
60's	mainframe	1M	10K	100s
70's	mini-computer	100K	1M	10s
80's	PC	10K	100M	1
90's	PC, portable, PDA	1K	1B	1/10
00's	appliances	0.1K	10B	1/100

Distributed Systems

A distributed system is a set of physically separate processors connected by one or more communication links.



Nearly all systems today are distributed in some way.

- Email
- File servers
- Printer access
- Backup over the network
- WWW

Tightly-coupled systems: "parallel processing"

- Processors share clock, memory, and run one OS
- Frequent communication

Loosely-coupled systems: "distributed computing"

- Each processor has its own memory
- Each processor runs an independent OS
- Communication should be less frequent

Distributed versus Centralized Systems

Issues:

- Communication
- Performance
- Reliability
- Security
- Programming models
- Transparency

Promise:

- Higher performance - more resources
- Higher availability - if one machine goes down use another
- Better reliability - through replication
- More security - make each small piece secure

Reality:

- Communication mechanisms and latencies important
- High performance - yes, with infrequent communication
- Worse availability - Lamport: "A distributed system is a system where I can't get any work done if a machine I've never heard of crashes."
- Worse reliability - can lose data if any machine crashes
- Worse security - anyone in the world can break into my system
- Programming model more complex
- Good news for OS researchers!

Issues in Distributed Systems

Communication:

- Users/processes on different systems can communicate.
- For example, mail, transaction processing systems like airlines, and banks, WWW.

Computational speedup:

- n processors potentially gives you n times the computational power
- Problems must be decomposable into subproblems
- Coordination and communication between cooperating processes (synchronization, exchange of results) is needed.

Reliability:

- Replication of resources yields fault tolerance.
- For example, if one node crashes, the user can work on another.
- Performance will degrade, but system remains operational.
- However, if some component of the system is centralized, a single point of failure may result
- **Example:** If a Linix lab workstation crashes, you can use another workstation. If the file server crashes, none of the workstations are useful.

Issues in Distributed Systems

Resource sharing:

- Resources need not be replicated at each processor (for example, shared files).
- Expensive (scarce) resources can be shared (for example, printers)
- Each processor can present the same environment to the user (for example, by keeping files on a file server)

Security:

- Authentication - do you have permission to do something
- Encription - preventing others from observing your actions
- Protection - orchestrating sharing just a specific thing