

---

Due in class on Wednesday 7 February 2001

---

1. (Exercise 1.4 from *Distributed Operating Systems*) Discuss the relative advantages and disadvantages of the various commonly used models for configuring distributed computing systems. Which model do you think is going to become the most popular model in the future? Give reasons for your answer.

Minicomputer: Advantages — simple, allows sharing of resources among the minicomputers. Disadvantages — time-sharing is centralized so it doesn't scale well, not very transparent.

Workstation: Advantages — scales well, good local response time. Disadvantages — have to maintain multiple independent machines, load distribution is hard and may not be supported.

Workstation-server: Advantages — distributes some load to dedicated servers. Disadvantages — server can be a bottleneck, doesn't take advantage of idle workstations.

Processor-pool: Advantages — good load distribution, takes advantages of idle processors. Disadvantages — hard to implement, not suited for some high-performance or highly-interactive applications that demand dedicated independent machines.

Hybrid: Advantages — local computing power plus extra computing power (through the pool) when needed, good load distribution. Disadvantages — hard to implement.

Which will become the most popular? You could probably make a case for any of the last three. *I gave full credit for any convincing case made.*

2. (Exercise 1.8 from *Distributed Operating Systems*) What is groupware? Why is it considered to be a promising technology for software development?

Groupware is software that allows people to work together electronically. It includes email, calendars/scheduling, conferencing, etc. It is promising for software development because it allows geographically distributed programmers and others to work together on projects in an effective manner, by automating and easing routine tasks, and supporting collaboration (*many examples could be cited here*).

3. How does a client / server information system compare to a “true” distributed operating system?

A “true” distributed operating system, as defined in class, is an operating system that works in the workstation or processor-pool model from question 1, balancing processes across the machines, and being as transparent as possible to the user (so processor-pool model is a better fit than workstation model). In contrast, a client / server information system work in the workstation server model, usually with multiple types of servers, at various levels of abstraction, and does not provide the same degree of transparency. However, a lot of the basic concepts and goals, such as transparency, scalability, etc. apply to both the true distributed operating system and the client / server information system.

(Note that the goal of this question was to compare the two (i.e., how are they similar, and how are they different), so an answer that was mostly a detailed description of each, with very little comparison, did not receive much credit.)

4. (Exercise 2.11 from *Distributed Operating Systems*) What is meant by internetworking? What are the main issues in internetworking? Explain the following terms: (a) bridge, (b) router, and (c) gateway.

Internetworking means connecting two or more networks to form a larger network.

The main issues are how to connect the networks together (bridge, router, gateway, etc.), what communication medium to use, and how to manage the resulting network.

A bridge, a router, and a gateway are all similar in that they restrict some information to one side of the device, while letting other information pass through to the other network. A bridge separates networks that differ at the data link layer (e.g., Ethernet and token ring), a router separates networks that differ at the network layer, and a gateway separates networks that differ at higher levels (perhaps involving a large protocol conversion task).

5. (Exercise 2.19 from *Distributed Operating Systems*) What are the main differences between connection-oriented and connectionless communication protocols? Indicate which of the two protocols is preferable for the transmission of the following types of information: (a) voice, (b) video, and (c) bursty data.

In a connection-oriented protocol a connection is established before data is transmitted; in a connectionless protocol data can be sent without establishing a prior connection. Also, in a connection-oriented protocol data is transmitted as a single stream of bytes so data always arrives in the correct order; in a connectionless protocol the data is broken up into packets so can arrive at the destination out of order.

(Again, note that you were not asked to define these two protocols, but to explain the differences between them.)

Voice — for high-quality audio, connection-oriented is best, so that data doesn't get lost and delayed, which would result in choppy audio. However, for low-quality audio, connectionless may suffice.

Video — connection-oriented is best, for same reason, as above, but again, connectionless may suffice, especially if the network is of sufficiently high bandwidth.

Bursty data — connectionless is best here, instead of tying up network connections when no data is being transmitted.