6.033 Handout 12: Assignment 5
March 11 through March 13

For Lecture, Monday, March 10

Last lecture on networking. Read Saltzer's "Case study: synchronized file transfer; failure retry", reading #18.

For Recitation, Tuesday, March 11

Your one-page reading report should address the following question regarding the end-to-end argument (Saltzer et al., "End-To-End Arguments in System Design", reading #16).

> The End-to-End Argument is an important principle in system design. Pick one of the systems we have read about already in 6.033 (Therac-25, X Windows, Birrell's threads or RPC package, Ethernet, etc) and give two examples of places where the designers did or did not obey the principles of the End-to-End Argument.
>
> To illustrate what we are looking for consider this question from 1995: "Translation between different byte-order schemes is sometimes a significant source of overhead, especially when it occurs at multiple points between two communicating parties (D. Cohen, "On holy wars and a plea for peace", reading #21). The X Window system deals with both big-endian and little-endian clients by providing two different network ports. Big-endian clients send requests and data to one port, while little-endian clients send requests and data to the other. The server may, of course, be implemented on either a big-endian or a little-endian machine. This approach is unusual. Discuss this design decision in terms of the end-to-end argument." *Remember, this is the 1995 question! You want to answer the question in the first paragraph!*

For your entertainment we also included reading #17 (D. Cohen, "On holy wars and a plea for peace"). It is not necessary to understand this article in full detail. You only need to understand the basic issues, and what big-endian and little-endian are.

For Lecture, Wednesday, March 12

This lecture starts a new topic in 6.033: *naming*. Naming is typically treated superficially in most systems subjects, although it is one of the most fundamental issues in systems. Many properties in computer systems and languages can be phrased as naming questions. We will see how naming is done in virtual memory systems, file systems, the Domain Name System (the Internet's name system), and the WWW. We will also see that scoping and inheritance in programming languages are naming issues. To get prepared read Saltzer Chapter 5.A, reading #19.

System aphorism of the week
A system takes longer to debug than you expect, even when you take this fact into account.

6.033 Handout 12, issued 3/4/97