REDUNDANCY AND CODING

1. Ways to Apply Redundancy

Section 8.D in the course notes discusses various ways to apply redundancy, including:

--channel coding (we’ll discuss in some depth today)
--component redundancy (we have discussed in the past)
--system redundancy (we’ll discuss a bit today)

2. Channel Coding

--Coding here does not mean "geeking out in an Athena cluster to finish a 6.170 homework assignment" but rather _transforming a message in preparation for transmission over a channel_.

--The channel could be the Internet, radio waves over 802.11 range, radio waves through outer space, a CD, the hard disk, etc.

--Channel coding can yield several different goals: error detection, erasure recovery, and error recovery.

--The simplest erasure-recovery code is "send every message twice". Such a code can be surprisingly effective in some contexts. (RAID-1, which you read about in appendix 8-A, is an example of such a code. In this case, "sending a message" means "writing a given sector".)

--An example of an error-recovery code is a _Hamming code_. Pages 8-29 through 8-31 detail this code.

--We’ll now quickly cover a more involved erasure-recovery code, called Reed-Solomon codes.

3. Reed-Solomon Codes (simplified view)

6.033 IN CONTEXT

4. Let’s talk a bit about how the material in this class relates to computer science and EE.

FAULT-TOLERANCE WAR STORIES

5. What’s N-version programming? How well do you think it works?

6. Why doesn’t memory come with error checking hardware any more (as mentioned in Appendix 8-B) yet disks do?

7. What can we learn from the national archives story in Appendix 8-B?

8. Single points of failure. What do people think about the electric generator story in Radin, Poland, described in Appendix 8-B?