Outline for October 6, 2005

Reading: B. Miller, L. Fredriksen, and B, So, "An Empirical Study of the Reliability of UNIX Utilities," *Communications of the CACM* **33** (12) pp. 32–44 (Dec. 1990).

1. Puzzle of the day
2. Bad programming (continued)
   a. File access race condition (*xterm* race)
   b. Signalling race condition (*ftpd* bug)
   c. Environment variables (*vi* games)
   d. Not resetting privilege (Purdue games)
   e. Unknown interaction with other system components (*finger* port is *finger* and not *chargen*)
3. Good programming
   a. Understand what the program is to do
   b. Design the program (or programs) accordingly
   c. Implement it, checking at each step for possible problems
   d. Put the components together, testing interfaces
   e. Test the program in the environment in which it is to be used

Puzzle of the Day

The following item appeared in the current issue of the RISKS Digest[1]:

> Date: Sat, 01 Oct 2005 00:26:34 +0000
> From: Matt Roberds <mroberds@worldnet.att.net>
> Subject: Buffer overrun in television sets
>
> A recent discussion in news:sci.electronics.repair concerned late-model RCA television sets that would suddenly lose their sound. Two repair technicians stated that they could find nothing physically wrong with the sets, and that unplugging the set for a while seemed to cure the problem. One technician later posted this link:
>
> http://www.iwaynet.net/~nesda/SilentCTC.html
>
> According to that article, a device from one particular manufacturer that is used to insert closed captioning and other data into the video stream is generating data that has two bits more than the specification. These two extra bits were causing the microprocessor in the television to become confused. The article claims that Sony, Hitachi, and Philips sets are also affected.
>
> That article is dated June 2001, but the discussion in the newsgroup appears to indicate that this problem has occurred more recently than that.

What other types of devices may have this problem? How could you check for it? How could you protect against it without rebuilding the device?

1. *The RISKS Digest* **24** (6) (Oct. 5, 2005).