CSCI 473: Network Programming
Yet another change

On Friday, February 1, 1991, from 10:00 AM to noon, Eduardo Biagioni, a graduate student at UNC-CH, is giving a tutorial over the MCNC video network on Unix interprocess communication, the topic what we are now covering in CSCI 473. It's often beneficial to hear two people present similar material and for that reason, I'm substituting Ed's lecture for the scheduled Friday makeup class in CSCI 473. I realize that some students will be unable to come at 10:00 AM. The lecture be videotaped and students who can't make the lecture may view it at their leisure.

I lecture may be viewed in the MCNC video classroom, *i.e.*, Rhodes ???. The MCNC network supports two-way video so you will be able to ask questions of Ed during the lecture. This lecture is part of a series given by contributors to a book called the *Unix System Handbook*. The purpose of the series is to give the contributors feedback on what they have written. Ed has written the chapter on Unix interprocess communication. (I turned down an offer to write that chapter, but that's another story.)

The ``official'' abstract for the talk.

> An overview of the different methods of interprocess communication available on UNIX machines. Not covered are high-level protocols and data transfer programs such as uucp or sendmail, nor low-level network management.
>
> Inter-process communication covers several independent mechanisms, all of which are used to communicate among different processes in Unix, running either on the same computer or on independent computers. Many of these mechanisms are based on analogous hardware mechanisms: Signals are analogous to traps and interrupts in a computer, pipes and sockets resemble asynchronous byte-wide communication over wires and busses, message passing and semaphores are available to synchronize processors in parallel computers, and shared memory is analogous to a multi-ported memory connected to several processors.
>
> Mechanisms which can be visualized without reference to hardware include RPC, which is similar to making a system call to a different machine, and datagrams, packets dropped into an unreliable network which may arrive out of order or disappear completely. The client/server model is like having remote program libraries that provide particular functions.
>
> The purpose of the talk is to provide an overview of the kinds of programs that use IPC and of the kind of problems that IPC addresses.