

Some projects

This handout lists a few projects suggested by people at UNC for the distributed systems course. You are not required to choose a project from this list and are, due to the large number of students in the class, encouraged to fashion your own.

A Distributed backup system for the Macs

Presently, every Mac user has to perform his/her own backups. The Macs are networked. There must be a better way.

A SLIP driver for the Macs and/or PCs

SLIP is Serial Line Internet Protocol. It allows machine connected via serial lines, such as dial up links to exchange IP packets. Joe Hewitt is “playing a small advisory roll in a little project on campus to write a SLIP TCP/IP driver for a mac or pc that will allow email and, possibly later, news to be read from remote.” Mail to hewitt@cs.unc.edu for details.

Instrumentation of the Cosmic Environment

The *Cosmic Environment* is a parallel/distributed processor programming environment from Caltech. It compiles C program into object code for real machines, such as Ametek 2100, Cosmic Cube, Intel iPSC, as well as into the code for a simulated environment, which is running on UNIX. The Cosmic Environment (CE) lacks instrumentation, such as facilities to count number and sizes of messages without changing the code. Another useful feature would be load visualization tools for each virtual processors, something like the one on SUNTOOLS. Mail to Ryutarou Ohbuchi, ohbuchi@cs.unc.edu, for details.

FAX to your office

Design a system to “capture” FAXed documents and direct them to workstations and/or PC's. Mail to Norm Vogel, vogel@cs.unc.edu, for detail.

Electronic voice mail

Presently, the CS department has a Watson card equipped PC-XT that will store messages on a hard disk. What would it take to make this useful to the CS department? Ask Norm Vogel for more information.

InteCom IBX call processing services

The operating system of the IBX switch supports OAI, Open Application Interface, which allows the attachment of an “application processor” to the IBX communication control channel. Possible projects involving this capability include providing a UNIX compatible interface to the IBX functions or implementing “call processing services.” More details about OAI are available from Norm Vogel.

Finding downed machines

Several months ago, Joan Curry (bdrc!jwc@mcnc.mcnc.org) and Dean Brock “specified” (and partially coded) a system that allowed users to specify actions to be taken when machines crash or uncrash. They would like someone to finish it.

Distributed IDLVIEW

To quote Rick Snodgrass (snodgrass@cs.unc.edu): “I have a project that involves making idlview, a debugger that works with dbx (and dbxtool) and knows about idl, truly autonomous. It uses sockets right now, but the communications are not at all robust, nor functional. The advantages I see to this project are that the non-communication portions are in very good shape, so the students can focus on the communications, and that it involves real-world issues of reliability and system design.”

Ethernet monitor in X-windows

The Sun NIT (Network Interface Tap) can be used to read *all* packets sent on an Ethernet segment. Design an X-window (or SunView) based system to display various parameters of interest about Ethernet traffic. Interesting information would include packets per second segregated by machine, sender or receiver, protocol, length, etc.

The Right Time

To the machine `john` is attached a clock receiving WWV, a radio station broadcasting the real time. Use this clock to give all machines the correct time.