CS 603 (Park) **Assignment I** **Due: Jan. 30 (Fri.), 1998**

PROBLEM 1

Read Chapters 1 & 2 from Tanenbaum.

PROBLEM 2

(a) Write a client/server application using sockets which emulates `finger` and `date` without arguments. The server (daemon) waits on port 19991, and upon receipt of a request from client C, it executes `finger` or `date` depending on the request, shipping back the output produced to the client. It then returns to waiting/serving the next request. Thus the server is an iterative server.

Use UDP for message transport where your packet consists of a header containing the client's IP address, port number, client's time stamp, and payload length where the payload contains either the string "finger" or "date". Notice that both the header and payload of your packet constitute the payload of the UDP packet. When determining the source address of the client at the server, use the information present in your own header rather than that of UDP. On the server, the server is to use the same packet format as the client where now the client's IP address and port numbers occupy the header part. In the time stamp part of the header, copy over the time stamp contained in the client's request packet so that it can be used to calculate the RTT/response time at the client.

Test your application in the following set-up. Run the server on xinu_i and run five clients on xinu_{i+1}, xinu_{i+2}, xinu_{i+3}, xinu_{i+4}, xinu_{i+5}. Each client takes two command-line arguments X and T where X indicates the command to be requested and T is a frequency parameter. Upon start-up, the client will send multiple requests to the server at T second intervals for a total request count of 10. For each response received, the client is to print the answer on the terminal as well as the RTT or response time.

Use `script` to record the entire interaction and test your system for $T = 50, 100, 500, 1000$ ms. Give an interpretation of your results.

(b) Repeat part (a), however, now with a concurrent server design. What, if any, performance difference do you observe across the two implementations?

PROBLEM 3

Problem 2.4, 2.7, 2.8, 2.10 of Tanenbaum.