CS 422 (Park) **Assignment II** **Due: Sep. 21 (Tue.), 1999**

PROBLEM 1 (20 pts)

(a) We occasionally make the analogy between networks and road systems including highways. Based on our discussion and present understanding of networks, list their commonalities and differences.

(b) Give a description of FDM and TDM in the framework and language of road systems.

Note: Your write-ups must be typed using your favorite word processing tool.

PROBLEM 2 (20 pts)

(a) Design a 1024-channel baseband (TDM) framing standard (similar to T1 for telephony) suited for carrying super-CD quality audio assuming a frequency range of 20Hz–20.02kHz. Assume a quantization level of 16 bits per audio channel and two framing bits (for synchronization) per frame. What is the data rate of your standard? What is the data rate per audio channel? What is the frame duration (i.e., time duration of each frame)?

(b) As a continuation of part (a), since the frame size for the super-CD audio standard is "not too small," we may need to adopt a particular bit encoding scheme to facilitate ease of clock recovery and synchronization (beyond the two framing bits). What is the resulting bit rate of the standard when you use NRZI with 4B/5B encoding? Make sure to indicate which parts of the frame are subject to encoding.

PROBLEM 3 (20 pts)

Give a 3-page summary overview of the following aspects of UNIX system programming: signals (`sigaction`, `signal`, `kill`, `pause`) and interprocess communication (`pipe`, `mkfifo`, `semget`, `shmctl`). Organize your description around the main system calls (a subset is indicated in parenthesis). Make sure to explain the notions *atomicity*, *blocking/non-blocking*, *synchronous/asynchronous I/O*, *mutual exclusion and semaphores*, and *interrupts*, and in which system calls they play a role. The `man` pages should be the primary source of reference.

PROBLEM 4 (30 pts)

You will find a client/server application under ~park/pub/cs422; the client program is `client.c` and the server program is `server.c`. Reverse-engineer the code and explain, in a high-level way, what the server and client are doing. Then, give a line-by-line explanation of the code—you may group several lines into one unit, as you see fit, and give a modular description. The important criterion is that your answer makes clear that you understand all aspects of the code. If you find inefficiencies, indicate what and why, and suggest possible changes. Include in your answer a clear explanation of what trick is being used to direct the output of the server back to the client and why this is nontrivial.

Compile the program and test the client/server application by first running the server process `server.bin` in the background followed by the client process `client.bin` in the foreground. Use the UNIX `script` facility to record the run-time session. Submit a copy of the session log along with your write-up.

PROBLEM 5 (10 pts)

The purpose of this problem is to introduce "snooping" raw data from the "wire" on the private LAN consisting of two Xinu machines, two laptops, and Ethernet hub. Execute the following command on a Xinu machine to capture a packet from the network:

```
% /usr/local/etc/sudo /usr/local/etc/snoopwrap-elxl0 -c1 -o /var/tmp/login-ID
```

The command will capture a packet from the Ethernet LAN and save it in the file `/var/tmp/login-ID`, where `login-ID` is your login ID. You will have to generate your own traffic on the private network by pinging a machine on the private net (`/usr/sbin/ping 10.0.0.X`) or by using telnet to communicate with a machine on the private net (`telnet 10.0.0.X`). Execute both and print out the log files in octal format. Can you find any commonalities, in terms of the bit patterns, between the two?