1. Compile the three programs `s_daemon`, `s_sh`, and `s_tln` from the sources in my cs376 subdirectory. Make printouts of the sources for these program for discussion in class (you have already done this for the program `s_sh` which was covered in a previous lab).

2. The program `s_daemon` is the skeleton of a telnet daemon which will listen at a (non-privileged) port and allow up to 4 concurrent dialogs (logging to files if you use the “-log” option). To test it, first open two xterms. In the first, start the program with `s_daemon -log < cr >`
   Write down the tcp portnumber (say it is 3384) and then go to the other xterm and start telnet
   `telnet localhost 3384 < cr >`
   On the telnet xterm you should see a “Ready” prompt and on the daemon xterm you should see a notation that the daemon has (in the case of Unix) done a fork() and called the procedure `dialog_with_teln()` or (in the case of Windows NT) has started `dialog_with_teln()` as an auxilliary thread (which will share the parent’s environment). This will be a “cooked session” with a newer version Unix telnet but if you telnet from Windows NT it will probably be `raw` with no conversion of LF to CR-LF.

   Take a close look at the procedure/thread `dialog_with_teln()`. Most of the work is done here. In the case of Unix it fork()’s and exec()’s the small shell `s_sh`. In the case of Windows NT, the small shell is spawn()’d and an additional thread `pipe_to_socket()` is created. The reason for this complication is that we do not have complete freedom to mix descriptors in the select() call in Windows NT.

3. Finally, instead of using the system telnet, try using the small version `s_tln` instead. You should try
   `s_tln -v localhost 23 < cr >`
   first just to see the telnet arbitration (which is usually hidden). Then start the `s_daemon`, get the portnumber (say it is 3384) and the type
   `s_tln localhost 3384 < cr >`
   This will be a `raw` telnet session, so there will be no command line conversion of LF to CR-LF. It should be clear after giving a few commands that telnet arbitration is usually necessary for a productive working session. You can break to the telnet command prompt with a CONTROL-T. Then type “help” and “status” to see all command-mode options.