PROBLEM 1 (20 pts)
Read Sections 2.1–2.6 in Chapter 2 from P & D. Solve Problems 2, 15, 18 and 20.

PROBLEM 2 (30 pts)
Assume radio stations in Indiana have become wary of transmitting analog data—i.e., audio—and have decided to use AM and FM to transmit digital data. Of course, this may include digitized analog data such as MP3 (MPEG Layer-3 Audio, a popular audio compression standard), but, in general, can be any digital data such as news bulletin text and files. For each analog transmission method—AM and FM—describe your design for both the sender side and the receiver side. Note that each AM or FM station is assigned a carrier frequency by the FCC which varies from station to station in a given geographical area to avoid interference. Your design description, by necessity, must include specification of baud and bit rate which, in turn, depend on the selected carrier frequency \( f \). When calculating the data rate (bps) for a given \( f \), you may ignore the effect of noise.

Check the frequency range of AM and FM radio stations. Based on your architecture, can CD quality audio (1.411 Mbps uncompressed; compression can bring things down to around ~100 kbps) be broadcast using AM or FM carrier frequencies? What about DVD video (6.144 Mbps)? Considering that your car stereo may need to be replaced with a new digital one—a lesson may be drawn from the recent not-so-successful introduction of Satellite radio—based on the throughput spec that your design achieves, argue the pros/cons of converting to digital.

PROBLEM 3 (50 = 20 + 30 pts)
(a) As a continuation of Problem (b) in Assignment I, fix the client/server example code, get it to run, and show the output. Briefly comment what you needed to change and why. Use the UNIX script facility to record the run-time session. Hand in the output of the script as well as the client and server code. You must provide adequate documentation in your code.

(b) Extend the client/server application such that any UNIX command—you may ignore argument options—can be requested by the client to be executed on the server and its output returned to the client. The request format should be of the form

\[ \text{process-ID} \# \text{command} \#
\]

Your client program is to take as its command-line argument the name of the UNIX command to be executed. Test your program by first running the server application in the background, then executing four copies of the client in the background “simultaneously” with arguments date, ps, time, and hostname, respectively.