ECES 338 Assignment #2 Due: February 12, 1999

Spring 99 (100 points)

Ozsoyoglu,G.

In this assignment, you will work with pipes.

(1) **Unnamed Pipes using C:** Write a C program to create a two-consumer-two producer paradigm where the producers (say, P1 and P2) and the consumers (say, C1 and C2) communicate using a single unnamed pipe. You will use unbuffered read()/write() and pipe() system calls in this program. For pipes, please read chapter 5 of the Unix textbook. The producers and consumers work at different speeds; each value produced by a producer is passed to (one of) the consumers via the pipe. A producer process should produce its values by reading them from its own input file. And, a consumer should append the values that it consumes to its own output file. After processing, say, 100 values, from the input files of the two producers, the output files of consumers C1 and C2 will have these values, randomly distributed into one of the output files. Use the sleep() function call with random values to simulate the producers and consumers working at different speeds. Turn in your code and a sample output. Please make a note of your recitation section and your login id.

(2) **Unnamed Pipes with dup() or dup(2):** Write a program that creates an unnamed pipe, forks a producer and a consumer. The producer performs execl() on, say, ls; the consumer performs execl() on, say, cat; and the output of ls is piped over the named pipe to cat. You need to use dup() or dup2() calls for associating the standard output of ls and the standard input of cat with the pipe. Turn in your code and a sample output. Please make a note of your recitation section and your login id.

(3) **Named Pipes through the Unix Shell:** Create a named pipe from the shell with the mknod command. Do an ls to list the pipe as a file. Redirect the output of a command (e.g., ls) into the pipe, and redirect the input of a command (e.g., cat) from the pipe. Script the session, and turn it in.

(4) **Named Pipes using C:** Write a program to create a named pipe, fork a producer to produce into the pipe and a consumer to consume from the pipe. Your main process should then do execl() on ls, which lists the contents of the created pipe as a regular file while the content of the pipe is changing. Script the session, and turn it in along with your code. Please make a note of your recitation section and your login id.