**ECT362 Homework #6**

This assignment looks at how interprocess communications (IPC) can be used to provide mutual exclusion among concurrent Linux processes. The problem to be solved is a variant of the producer consumer problem, first posed by Dijkstra. Consider a producer which is responsible for gathering data about up to the minute stock prices at the stock market to create a ticker. Also consider a consumer which gets the stock prices from the producer via the ticker. The stock market ticker is a critical resource in which only one producer or consumer may access at a time. Likewise, care must be taken to ensure that the consumer does not try to consume ticker information before it exists.

Create a design for a producer and consumer process that share information using an anonymous pipe for communications. Use the code examples worked in class from LSN 13 as a starting point in your design. The producer process is responsible for obtaining randomly generated 3-character stock symbols and their associated randomly generated valuations. The consumer process is responsible for retrieving the stock information from the anonymous pipe. The output of the code must indicate all interactions with the shared ticker and the nature and element of the interaction (produces X / consumes Y). Your program should run for a user determined amount of time, read in as a command line argument (you will want to use the `time()` function).

Full credit for this assignment will be based upon a report that includes the program design (10%), the final source code implementation (75%), the source code documentation (10%), and the professionalism of your report and code (5%). The software design can be generated using pseudo code, activity diagrams, or a combination of both and should not be included in the program, but in the associated report. The report must explicitly describe the selection process of the storage mechanism(s) used for the ticker information (symbol and value) and the design of the timer controlled program termination. The report must also fully explain how the Linux pipe mechanism is able to provide mutual exclusive use of the ticker.

The implemented code must execute for full credit. The code needs to be thoroughly documented. The report must document the “why” behind all of the coding decisions (trial and error is not effective software engineering). Any issues encountered, key design methods used, or any supplemental material referenced must also be included for full credit.