Machine Problem 2 Report

Lab members: Martin Fracker, Chris Findeisen

Date: Feb. 20, 2015

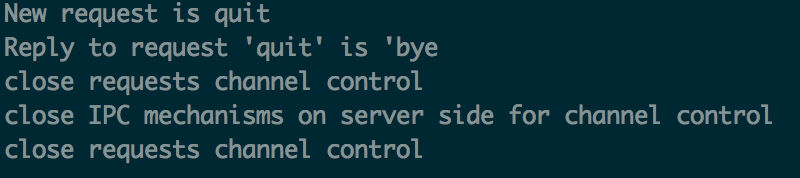
**Introduction**

This week’s lab, we focused on inter-process communication using request channels. We accomplished this by leveraging fork and exec unix commands. Then we analyzed the performance of running two separate processes versus localizing the logic into a single process.

**Procedures**

We began by creating the skeleton code for client.c. This consisted of forking the process and using printf statements to test whether or not we had successfully branched our program into parent and child logic. Next, we coded the client logic into the parent branch; and the child logic consisted of loading the dataserver program into the child process. In order to compare a function call versus using a request channel, we used a function taking a string and returning the same string, and compared the time to execute the function with the time to go through the request channel.

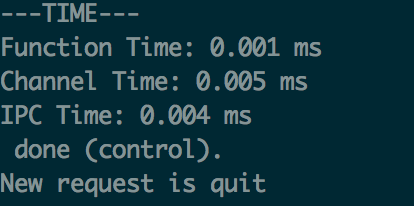
In the following pictures, we can see the client process communicating successfully with the dataserver process:



**Results**

We found that the function call was four times faster than the request channel. This

result is reflected in the following picture.



**Conclusion**

It was not surprising to find that the function call was faster than the request channel. The cause of this apparent overhead is the context switching overhead incurred when we have system logic split between multiple processes.