

HW2: IPC, Multi-threading and Synchronization

March 31, 2016

Policy

- This homework is due on **April 15**. Late submission of the homework will be punished(33% score deduction per day).
- Collaboration: Discussion with your fellow students are allowed; but copying the answers or sharing your answers with other students is NOT acceptable and will result in a grade of zero.
- Submission: Please send email with your **source code**(if it's too large, you can give a download link) and report(includeing your programming design, core codes and results.) to the TA(zonghua@shanghaitech.edu.cn). Any homework with no report will be published with 20% reduction of your scores.

Introduction

In this assignment, we will explore concurrent programming with threads and locks operating on a hash table. Please note that all of you are supposed to do the assignment on Linux (not xv6). It is ideal to do this assignment on a computer which has a processor with multi-cores. Since most of the recent laptops do have multicore processors, don't worry about that.

Description

In this homework you should create two processes. One process works as a server. It concurrently builds a hash table and then waits for commands(includeing add, delete and search) from the client. Another process works as a client which sends commands to the server and receives results from it. The server adds the received commands into a command queue but will not run them until it receives "exec".

Hint: Command "search" can be done concurrently, while "add" and "delete" cannot.

Requirements

1. Concurrently build a hash table with 1 to 100000 as keys and 5 buckets. If you are not familiar with hash table, you can read materials on wikipedia(https://en.wikipedia.org/wiki/Hash_table).
2. Implement 3 operations: add an element to the hash table, delete an element from the hash table and search for an element on the hash table.
3. Concurrently run "search" operations while serially run "add" and "delete" operations.
4. Use fork function to build two processes, the parent process builds the hash table and executes received commands called server while the child process sends commands and waits for results from the server called client. You can use pipe to implement inter-process communication.