CSE344 – System Programming

Midterm Project Report

1) How Did I Solve This Problem?

In this homework, I checked the first command line arguments. To satisfy the requirements I used to getopt function. Thanks to getopt function I got arguments in the proper way and I got the command line argument, for both client and server files. After the get input file. I read the file and store it into a request struct which contains both pid value of the matrix, pid of the client and the pipe used to access data in serverZ. After the file readed I create a fifo which is based on the pid name, and send data to serverY and then it starts to wait until the response comes.

After the client send data to serverY, serverY starts to read. While the serverY is initializing , workers and serverZ are initialized too. Pid of serverZ is kept for the handle sigint signal. Active workers of serverY kept in unnamed semaphore and active workers of serverZ is kept in named semaphore, to access both in serverY and serverZ. According to semaphores, request is forwards to proper worker. ServerY sends request to workers via pipe, serverZ sends requests to workers via shared memory. After the calculations are done, worker, responds to client's request. To satisfy the required outputs, several semaphores are used. To synchronize the child of serverZ, I used stop and continue signals. To control whether the worker is available or not.

2) My Design Decisions

I use error check first mechanism.
I use both fork and both fork execve methods to create processes.
I use pipes, shared memory and semaphores.
I use childStatus, request structs.

3) Requirements That I Achived

I think I achieved almost all the requirements. However, I may not have been able to achieve some requirements. Note: When the serverY is become daemon, handling SIGINT is broken somehow, so I couldn't figure that out. Also getting cofactor, cause a problem to requests.

4) My Files

Become_daemon.h - Header file to make serverY and serverZ daemon.

Fifo_seqnum.h - Header file for struct request.

Makefile - The makefile.

serverZ.c - ServerZ implementation.

serverY.c - ServerY implementation.

client.c - Client implementation.

5) Some Outputs From Program

```
Tue Apr 19 06:51:38 2022
(Server Y log, p = 2, t = 7) started
Tue Apr 19 06:51:38 2022
Instantiated server Z
Tue Apr 19 06:51:38 2022
Z output:
Tue Apr 19 06:51:38 2022
(Z:Server Z log, t7, r = 2) started
Tue Apr 19 06:51:43 2022
Worker PID#66298, is handling client PID# 66486, matrix size 2x2, pool busy 1/2
Tue Apr 19 06:51:43 2022
Worker PID#66299, is handling client PID# 66487, matrix size 2x2, pool busy 2/2
Tue Apr 19 06:51:43 2022
Forwarding request of client PID#66488 to serverZ, matrix size 2x2, pool busy 2/2
Tue Apr 19 06:51:43 2022
Z:Worker PID#66300, is handling client PID# 66488, matrix size 2x2, pool busy 1/2
Tue Apr 19 06:51:43 2022
Forwarding request of client PID#66489 to serverZ, matrix size 2x2, pool busy 2/2
Tue Apr 19 06:51:43 2022
Z:Worker PID#66301, is handling client PID# 66489, matrix size 2x2, pool busy 2/2
SIGINT received, terminating Z and exiting server Y. Total requests handled: 0, 0 invertible, 0 not. 2 requests were forwarded.
Z:SIGINT received, exiting server Z. Total requests handled 0, 0 invertible, 0 not.

```

```
Client PID#73962 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73963 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73964 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73967 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73966 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73969 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73970 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73968 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:30 2022
Client PID#73971 (test.csv) is submitting a 2x2 matrix
Tue Apr 19 06:52:36 2022
Client PID#66490: the matrix is invertible, total time 0.000058 seconds, goodbye.
Tue Apr 19 06:52:36 2022
Client PID#66493: the matrix is invertible, total time 0.000075 seconds, goodbye.
Tue Apr 19 06:52:36 2022
Client PID#66492: the matrix is invertible, total time 0.000060 seconds, goodbye.
```