Operating Systems 1  
programming assignment

RAJ PATIL  
CS18BTECH11039

**PLAGIARISM STATEMENT**

I certify that this assignment is my own work, based on my personal  
study and research and that I have acknowledged all materials and sources  
used in its preparation, whether they be books, articles, reports, lecture notes,  
and any other kind of document, electronic or personal communication. I also  
certify that this assignment has not previously been submitted for  
assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

**Name: Raj Patil**  
**Date: 13th November 2019**  **Signature** 

# Report for file : CS18BTECH11039\_prg1.c

main objective : clearly defined in question statement

timed run

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ time ./a.out

---------

CS18BTECH11039\_prg1.c

---------

hey from (child 1)process # : 4736

hey from (parent)process # : 4735

hey from (child 2)process # : 4737

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 1)process # : 4736

hey from (child 2)process # : 4737

hey from (child 2)process # : 4737

hey from (parent)process # : 4735

hey from (child 2)process # : 4737

hey from (parent)process # : 4735

real 0m20.007s

user 0m0.001s

sys 0m0.003s

x-x-x-x-x-x-x-x-x-

output explanation

the time taken is as expected : a little over 20 seconds

the first 3 prints just as they are scheduled by the processor and are not under our control (their chronology)

the next 18 prints are from child 1 (operates for 9 seconds after initial 1 second sleep)

the next 2 prints are from child 2 the latter one being for killing child 1

then parent prints its message for getting a status update on child1

then child 2 gets killed after printing its last message after which parent prints its last message and exits

# Report for file : CS18BTECH11039\_prg5.c

main objective : independently compute some entity over some data in two children processes and pass it on to the parent for some useful conclusion

chosen specialised objective: compute the statistics ( just the mean in this case) of C's pseudo-random number generation facility

Exlanation: - (not chronological)(as the code is structured):

1. initialized an array of 100 doubles using run-time seeded rand function

2. declared variables:

- mean: final output calculated from separate data from children

- halfMean: name holder for the separate variables in each process

- NOTE: 3 exclusive variables are being created when forking with just the same name for them so the primary requirements of the question are being met.

3. created file descriptors for IPC ( pipe )

4. first fork : entering child 1:

- closing read descriptor

- computation of halfMean explained in comments

- updating write file descriptor with value of halfMean

5. by this time child 2 has done the same for the second half of the array and is waiting for child 1 to finish execution and terminate.

6. parent closes the write file descriptor now and reads in the value of the first halfMean after child1 has finished execution.

7. now child2 writes to the write descriptor and parent reads from it.

8. parent computes the mean and prints the output

Note : not summing up first and then dividing to compute the mean as this could result in exceeding the bounds of the defined variables field(overflow issues)

# RUNS :

one deterministic run to check the correctness of the procedure

where A[i] = i

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ ./a.out

----------

CS18Btech11039\_prg5.c

----------

first halfMean = 24.500000

second HalfMean = 74.500000

mean : 49.500000

This is as expected ( 0 + 99/2)

hence the program is working correctly

now doing what it was meant to do:

i.e. random runs

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ ./a.out

----------

CS18Btech11039\_prg5.c

----------

first halfMean = 1171186826.920001

second HalfMean = 1078905746.620000

mean : 1125046286.770000

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ ./a.out

----------

CS18Btech11039\_prg5.c

----------

first halfMean = 1053712021.900000

second HalfMean = 1109076059.480000

mean : 1081394040.690000

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ ./a.out

----------

CS18Btech11039\_prg5.c

----------

first halfMean = 1128400971.640000

second HalfMean = 999443480.840000

mean : 1063922226.240000

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ ./a.out

----------

CS18Btech11039\_prg5.c

----------

first halfMean = 1139895343.220000

second HalfMean = 1158039525.440000

mean : 1148967434.330000

rajp152k@Raj:~/links/sem3/Operating Systems 1/assignments/programmming\_assignments$ ./a.out

----------

CS18Btech11039\_prg5.c

----------

first halfMean = 1179166024.099999

second HalfMean = 1026370852.419999

mean : 1102768438.259999