

Project 5: Fork/Wait, command line arguments and file handling

4/25/2022

40 Possible Points

Attempt 1



IN PROGRESS

Next Up: Submit Assignment



Add Comment

Unlimited Attempts Allowed

Details

In this project you will be writing a C program that forks off a single child process to do a task. The main process will wait for it to complete and then do some additional work.

Your program should be called *mathwait.c* and it will be called with a filename followed by a series of numbers. So for example:

```
./mathwait tempfile.txt 32 9 10 -13
```

Optionally, your program should also take in one option:

-h : This should output a help message indication what types of inputs it expects and what it does. Your program should terminate after receiving a -h

After processing and checking for -h, your program should then do a call to *fork()*. The parent process should then do a *wait()* until the child process has finished.

What the child process should do:

The child process will take all the numbers from the command line arguments and put them into a dynamic array of a large enough size for those numbers.

Once this is done, you should then open the file you were given for writing and then write all the numbers to the file. However, whenever the child writes to the file, it should write it in the following format:

Child: PID: Data



<https://umsystem.instructure.com/courses/97754/modules/items/4445932>

Assignment

<https://umsystem.instructure.com/courses/97754/modules/items/4445932>

It should then process this array to see if *any two* of the numbers sum up to 19.

Your process should then output any pairs that sum up to 19 in the file, so in our file we would output:

Child: 817: Pair: 32 -13 Pair: 9 10

Note that the pairs can be in any order, as long as you list all the possible pairs. Once complete, the child process should close the file, free the dynamic array and terminate. It should give EXIT_SUCCESS if it found at least one pair that summed up to 19 and an EXIT_FAILURE if it found none.

What the parent process should do:

After forking off the child process, the parent process should do a wait call waiting for the child to end. It should then check the status code returned from the child process and write that to the file. For example, assuming its process ID was 816 and it got EXIT_SUCCESS:

Parent: 816: EXIT_SUCCESS

For this project, you only need one source file (mathwait.c) and your Makefile.

Choose a submission type

