Brent Kuhn
4669
Programming Assignment 1

While the supplied example code for this assignment used only one child to compute the total of the given array, the scope of the assignment was to make the program use a user-input number of children to do the task. The first change necessary was to make the children have a two dimensional array for indexing and pipes (the variable children in the supplied code). All children would be indexed with their child value and the value of the end of the pipe (read[i][0] or write[i][0] in the supplied code). This allowed indexing using a for loop to iterate through all children in the code. All read and write pipes were written “from the parents point of view” such that when the parent writes, a write pipe is open and when the parent reads, a read pipe is open. As is typical with a simple parent/child program, the parent creates the children, and opens a write pipe to the child using the write end of the write pipe, and the child reads the write pipe using the read end of the write pipe. The parent keeps track of the totals which are read from the read pipes connected to the children. All children print their corresponding values using printf strings, and the parents string is concatenated using strcat as the data is available.

This code should execute for any given child number and input array in the order of childNumber spaceSeparatedArray (as can be seen below). One possible “Known issue” is that the handler for separating arrays is not entirely “even” as the last element will be less than or equal to the others in size due to the logic used to separate them, however during class professor Luis stated that for simplicity the code only needed to work for 3 children and 6 values to get a perfect grade.

Below is the output of the code using two sets of test values executed on ubuntu 16.04 LTS (tested on two machines).

```
brent@brent-MS-7A63:~/git/os-concepts/code1$ ./4669 2 2 3 5 7 1 3
I am parent with pid 10698
I am child with pid 10699: received 2 3 5 sending partial sum 10
I am child with pid 10700: received 7 1 3 sending partial sum 11
I am the parent of: child 10699 who returned partial sum 10 child 10700 who retu
rned partial sum 11 which has a total sum of 21
```

```
brent@brent-MS-7A63:~/git/os-concepts/code1$ ./4669 3 1 2 3 4 5 6
I am parent with pid 11056
I am child with pid 11057: received 1 2 sending partial sum 3
I am child with pid 11058: received 3 4 sending partial sum 7
I am child with pid 11059: received 5 6 sending partial sum 11
I am the parent of: child 11057 who returned partial sum 3 child 11058 who retur
ned partial sum 7 child 11059 who returned partial sum 11 which has a total sum
of 21
```

FLORIDAPOLY

Florida Polytechnic University – Computer Science Department
Programming Assignment 1 - Process API

Submission deadline: Jan 22th 11:59 pm

Deliveries:

Code: a .c program following the instruction of the section:Format.

Report: Brief report explaining how you address the problem and screenshots of the results. Include the paragraph in your report, and sign (failing in include the paragraph and signing will result and total grade of zero in this homework.

I certify that I coded this program by myself and this code doesn't correspond to the intellectual work of someone else.
Signature: _____

Submission: Zip your program and the report in a zip file with your name and upload the file to Canvas.

Item	Value
Program	75
Report	25
Total	**100**

Format: A .c program will read the number of children and the elements of the array as command-line argument as show in Example 1, and will output the result as in Example 2. The program should work for 1, 2, and 3 children.

Here, a link about parsing the command line arguments in C.
https://www.cs.swarthmore.edu/~newhall/unixhelp/C_commandlineargs.php

Note: Use your student ID number as your program name.