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.

Example 1: program execution.

```
./luis 2 2 3 5 7 1 3
```

In this example the number of children is 2, and the elements of the array are {2, 3, 5, 7, 1, 3}

Example 2: of program outocome.

Assuming parent has pid 2, child 1: 3, and child 2: 4. We want to see the following output

```
>>I am the parent with pid: 2 sending the the array: 2, 3, 5, 7, 1, 3
>>I am the child with pid: 3, adding the array 2, 3, 5 and sending partial sum10
>>I am the child with pid: 4, adding the array 7, 1, 3, and sending partial sum 11
>>I am the parent of pid: 3, and pid: 4.  I have pid: 2 and got partial results 10, 11, final result is 21.
```

Note: A C program compiled using the ember system (TA's will run the program using just the ember system, programs not running in that system will have automatically a grade of zero).

Inter Process Communication (IPC)

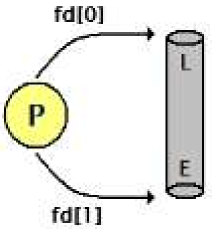
There are several mechanisms for IPC those include: Sharing Memory models, Message Passing model, Sockets, and pipes among others.   For this programming assignment we will use pipes.

Tutorial for pipes 1: https://users.cs.cf.ac.uk/Dave.Marshall/C/node23.html
Tutorial for using pipes 2: http://tldp.org/LDP/lpg/node11.html

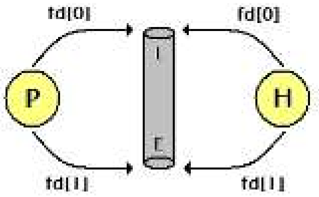
Basic ideas about pipes.

1. When the program is executed a process is created (parent process)

2. The parent process creates a pipe (fd) instruction where fd is an array of two descriptors, fd [0] points to the read end of the pipe and fd [1] to the write end of the pipe).

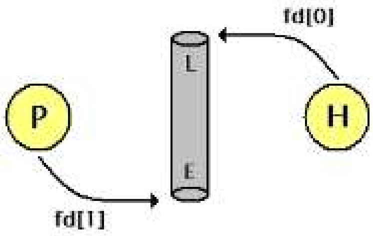


3. Then the parent must create the child process (fork instruction). When the child is created this "inherits" the same descriptors of the parent.

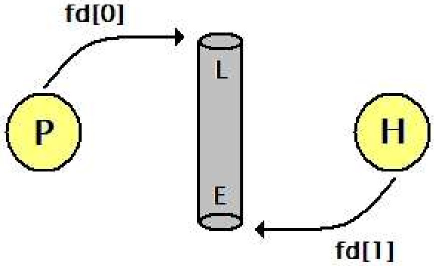


4. This way you can begin to establish communication between both processes.

5. If the PARENT wants to send data to the CHILD ti must close his descriptor fd [0] and the CHILD close its descriptor fd [1] .



6. If the CHILD wants to send data to the PARENT it must close its descriptor fd [0] and the PARENT close its descriptor fd[1] .



CODE EXAMPLE USING: ONE-DIRECCTIONAL AND BIDIRECTIONAL COMMUNICATION (Copy paste this link in your browser)

https://translate.google.com/translate?sl=auto&tl=en&js=y&prev=_t&hl=en&ie=UTF-8&u=https%3A%2F%2Fwww.programacion.com.py%2Fescritorio%2Fc%2Fpipes-en-c-linux&edit-text=&act=url