CIS 41B – Assignment 5: processes, sockets

Write a program that measures the data transfer speed between processes. Then write a conclusion based on the resulting program output.

Here are the key parts of main:

• Call a highLevel function that:

• Creates a child process that uses a multiprocessing object to pass data to the main process.

• The main process and child process will pass data back and forth in a loop, with a timer that starts right before the loop begins and ends right after the loop ends.

• The function returns the ratio: number of one-way data transfer / time difference.

• Call a lowLevel function that:

• Creates a child process that uses a socket to pass data to the main process.

• The main process and child process will pass data back and forth in a loop, with a timer that starts right before the loop begins and ends right after the loop ends.

• The function returns the ratio: number of one-way data transfers / time difference.

• The data that the main and child processes pass back and forth is an integer:

• The main process initializes the integer with 0 and increments it to 1, then sends it to the child.

• The child increments the integer and sends it back to the main process.

• The main and child processes keep incrementing the integer at each turn, in a loop which loops 10,000 times.

[4pts extra credit:

• Instead of incrementing an integer, append to a list.

• The main process initializes an empty list by appending 0 to the list, then sends the list to the child.

• The child appends a 1 to the list and sends it back to the main process.

• The main process keeps appending 0 while the child keeps appending 1 to the list, in a loop which loops 300 times.

• If you choose to do the extra credit, it's recommended that you start with the integer data first. When everything works, then change the data to a list.]

• Call the lowLevel and highLevel functions a couple times and print the return values along with the following system information:

OS: Windows

Num of cores: 8

<name of multiprocessing object> nnn

Socket nnn

<name of multiprocessing object> nnn

Socket nnn

<name of multiprocessing object> nnn

Socket nnn

Pseudocode for the steps of highLevel or lowLevel function:

• create a child process and the data transfer mechanism (for example, the socket)

• test that the connection is okay by:

• send an integer 1 to the child

• get the returned data from the child

• verify that the returned data is 2

raise an exception to print an error message and end the program if data is not as expected

• re-initialize the integer

• start the timer

• loop 10,000 times

• add 1 to the integer

• send the data to the child

• receive the data from the child

• record the end clock time

• send a 0 to the child so it can terminate

• wait for the child to end

• verify that the returned data is as expected  
 raise an exception to print an error message and end the program if data is not as expected

10 return the ratio: number of one-way data transfers / time difference

Pseudocode for the steps of the child process. (If socket is used, the child is the server)

• receive the data

• loop as long as the data is not 0

• increment the integer

• send the data back

• receive the next data