Due: Wednesday, October 16, 2024, 11:59 PM

CSCI 356: Programming Assignment 3
Inter-process Communication

Program Specification

In this project, you will be creating a program that uses **inter-process communication** to simulate a modified card game (War) tournament by using ***pipes***.

You will write a program that uses fork to create two child processes that will play rounds of War. The child processes will wait until the parent "signals" (Note: you don't actually need to use UNIX signals to do this, you can use your IPC methods) that they should choose their next "draw." The children should repeat this wait-draw-(suit)?-wait process until they are signaled to exit. Note, that in the event of an initial tie, the parent will make an additional request for suit from each child. Lets says that Spades beats Hearts, Hearts beats Diamonds, and Diamonds beats Clubs. Of course you can infer that Spades beats Diamonds and Clubs, etc… If the suits are also the same, then a tie will be declared with no winner.

The parent process should output each child's process id and, for each round, output each child's "draw" and the result of the round. After the tournament is over, the parent should output the aggregate results.

For Honor's section: no ties are allowed, and once a card has been drawn, it cannot be drawn again for duration of game. Note: you should indicate in your author block if you are in the Honor's section.

For example:

$./war 3

Child 1 PID: 113

Child 2 PID: 114

Beginning 3 Rounds…

Fight!

Round 1:

Child 1 draws 8

Child 2 draws King

Child 2 Wins!

Round 2:

Child 1 draws Jack

Child 2 draws Ace

Child 2 Wins!

Round 3:

Child 1 draws 5

Child 2 draws 5

Checking suit…

Child 1 draws 5 Diamonds

Child 2 draws 5 Clubs

Child 1 Wins!

Results:

Child 1: 1

Child 2: 2

Child 2 Wins!

Note: You must create your program so that it uses **inter-process communication with pipes**. I suggest that you design your WAR protocol to be as simple as possible. In other words, use 1-10 for numeric cards and maybe 11 - 14 to represent Jack, Queen, King, Ace and another for suit, maybe 1 – 4 or 15 – 19 to represent Spades, Hearts, Diamonds and Clubs instead of strings, etc…

Files you are required to upload to the Drop-box:

- war_pipes.c

As always, the source code must be indented properly and contain comments that describe important parts of the algorithm. The use of proper indentation is considered good programming style, and you are expected to use good programming style.

Add submission

Submission status

Submission status	No submissions have been made yet
Grading status	Not graded
Time remaining	8 days 9 hours remaining