CS 392: Homework Assignment 6
Due: April 22, 11:55pm

Philippos Mordohai
Department of Computer Science
Stevens Institute of Technology
Philippos.Mordohai@stevens.edu

Collaboration Policy. Homeworks will be done individually: each student must hand in their own answers. It is acceptable for students to collaborate in understanding the material but not in solving the problems. Use of the Internet is allowed, but should not include searching for previous solutions or answers to the specific questions of the assignment.

Late Policy. No late submissions will be allowed without consent from the instructor. If urgent or unusual circumstances prevent you from submitting a homework assignment in time, please e-mail me explaining the situation.

Objective

The goal of this assignment is to implement pipes connecting two processes, one of which interacts with the user.

Simple Interactive Game using Pipes (100 points)

You will implement a simple game where the one process (the parent) randomly picks a number between 1 and 32, while the other process (the child) accepts guesses from the user and returns appropriate feedback. Specifically, the child process asks the user to enter a guess and writes the guess to the parent via the pipe. The parent compares the guess to the chosen number and returns one of three possible messages: “too low,” “too high” or ”bingo.” The game continues until the user guesses the chosen number. The bingo message signals to the child that the game is over. When that happens, the child exits with exit code 0, the parent waits for it and then also exits.

This game requires communication in both directions, hence two pipes. The tricky part is that reading from a pipe is a blocking call, so you should establish a protocol that specifies which process goes first. For example, you could have the parent write a welcome message to the child before it accepts the first guess. See the relevant example in the IPC notes.

Requirements

(1) The parent must pick a number between 1 and 32.

(2) The game continues until the chosen number is guessed correctly, as long as the user provides integer guesses

(3) There is no need to verify that the user's inputs are indeed integers. We will assume that the user wants to win.

(4) You are not allowed to use `popen()`.

(5) When the bingo message is received by the child it should exit with code 0. The parent must wait for the child to terminate and then also exit with code 0.

Hints

(1) The parent should use `rand()` to pick the number and `srand()` to change the random seed so that a different number is picked every time. See https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm

(2) For readability, you should write `parent()` and `child()` functions that will be called after `fork()`.

(3) Include the following headers and definitions.

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#define MSGSIZE 60
```

Deliverables

Submit your code in a zip file, including a makefile. The structure is up to you.