

UNIVERSIDAD EMPRESARIAL

SIGLO 21



CARRERA: LIC. EN INFORMATICA

Trabajo Práctico número 3

Materia: Algoritmos concurrentes y paralelos

Profesor: PIRAY, EDUARDO ENRIQUE

Modulo: 3

Alumno: Soler, Diego Francisco.

Fecha: 01/07/2024

Objetivo:

El objetivo principal de esta actividad consiste en que el estudiante trabaje en posibles soluciones e implemente una solución a la problemática planteada, en la cual pondrá en juego las competencias de la materia. Puntualmente, trabajará con concurrencia y paralelismo, en memoria compartida y distribuida mediante procesos, subprocesos, hilos *pipes* y *sockets*.

Consignas:

Diseña y desarrolla una aplicación que permita simular el juego de cartas del siete y medio. La aplicación deberá operar con varios procesos mediante un modelo cliente/servidor, clientes = jugadores, y servidor = {distribuir las cartas, recoger las decisiones de los jugadores, hasta que se declare un ganador}.

La ejecución de la partida será mediante el comando: `iniciar_juego N`. N será el número de jugadores que participan en la primera partida. Como resultado de la ejecución, se publica el ganador de la partida, así como una tabla describiendo la situación de cada jugador (cuántos puntos posee, se plantó o abandonó).

El proceso `iniciar_juego` será el encargado de gestionar la partida, será el que repartirá las cartas por turnos, mantendrá el estado de la partida a partir de las decisiones de los jugadores, y decidirá finalmente el ganador de la partida.

El proceso jugador será el que implementará la lógica del jugador en la partida, de acuerdo a las cartas que reciba del repartidor (`iniciar_juego`), y de su propio estado (puntos), decidirá si continúa en el juego pidiendo más cartas, se queda ahí, o abandona (porque se pasó). Esta lógica de juego se deja abierta, de forma que se implemente como cada grupo lo decida, pudiendo llegar a ser simplemente aleatoria: según un número aleatorio, el jugador decide plantarse, pedir carta, o abandonar.

Recuerda que la generación de los N procesos jugadores deberá realizarse a través del uso de *forks*, y la comunicación repartidor/jugador se efectuará mediante *pipes*.

Descripción del Código

Este código implementa un juego de cartas similar al Blackjack, utilizando programación concurrente en C++. El juego cuenta con múltiples jugadores que reciben cartas en varias rondas y acumulan puntos, intentando no superar los 21 puntos. El juego sigue estas etapas principales:

1. **Definición de Constantes:** Se establecen constantes para el número de jugadores, valor máximo de una carta, límite de puntuación, y número de rondas.
2. **Clase Jugador:** Define un jugador con atributos como ID, puntuación, estado de plantarse, y pipes para comunicación interproceso.
3. **Clase Juego:** Maneja la lógica del juego, crea procesos hijos para cada jugador, y controla el flujo del juego a través de múltiples rondas.
4. **Método Iniciar Juego:** Inicia el juego creando procesos hijos para los jugadores, distribuye cartas, y maneja la lógica de plantarse o continuar jugando.
5. **Generación de Cartas:** Genera cartas aleatorias para los jugadores.
6. **Método Main:** Inicia el juego y maneja posibles excepciones.

Razón por la cual no se actualiza el último marcador

El último marcador puede no actualizarse correctamente por varias razones posibles en el código:

1. **Señal de Fin de Juego:** Después de enviar la señal de fin de juego (-1) a los jugadores, los pipes de lectura se cierran inmediatamente. Esto podría impedir que los jugadores procesen y respondan a la señal de fin de juego adecuadamente.
2. **Sincronización y Espera:** No hay sincronización adecuada para esperar a que todos los jugadores procesen la señal de fin de juego antes de cerrar los pipes. Es posible que los jugadores no tengan tiempo suficiente para actualizar y enviar sus últimas puntuaciones antes de que los pipes se cierren.
3. **Lectura de Respuestas:** La lógica de lectura de respuestas podría no estar manejando correctamente el caso cuando la puntuación no cambia debido a un jugador que ya se ha plantado o ha superado el límite de 21 puntos.