

Algoritmos concurrentes y paralelos

Trabajo Práctico 2

21-11-2022

AGUSTIN ZAPPONE

DNI: 38327363

Legajo: VIN012766

a. 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*.

A modo resumen, se puede decir que este TP estuvo lleno de desafíos.

El programa arranca mostrando las partidas ganadas, veces plantado y veces retirado de cada jugador.

Luego procede a pedir que se ingrese una opción:

```
Jugador 0 | Partidas ganadas: 0 | Veces plantado: 0 | Veces retirado: 0
Jugador 1 | Partidas ganadas: 0 | Veces plantado: 0 | Veces retirado: 0
Jugador 2 | Partidas ganadas: 0 | Veces plantado: 0 | Veces retirado: 0

Jugador 0 - Puntaje: 0.000000
Jugador 1 - Puntaje: 0.000000
Jugador 2 - Puntaje: 0.000000
Banca - Puntaje: 0.000000

Jugador 0, elija una opcion:
1.Recibir carta
2.Plantarse
3.Retirarse

Opcion: 
```

Si se elige la opción 1, el programa procederá a pedir que se ingrese una apuesta:

```
Jugador 0, elija una opcion:  
1.Recibir carta  
2.Plantarse  
3.Retirarse  
  
Opcion: 1  
ingrese su apuesta:
```

Sin embargo, el programa tiene algunos defectos.

Al terminar la partida, el programa entra en un loop infinito y no procede a la partida siguiente. Aunque termina mostrando a los ganadores de la partida actual.
Si la banca gana, no llega a mostrarlo y entra en el loop infinito antes de eso.
No he encontrado forma de arreglar este problema. Pero después de varios intentos de arreglar otros problemas, me he dado cuenta del uso e implementación de los pipes (lo que ha solucionado muchos otros problemas).