

Sistemas Operativos

Tarea 2:

Profesor: Viktor Tapia
Ayudante Cátedra: Francisco Olivares
Ayudantes Tareas: Gabriel Valenzuela – Giorgio Pellizzari

15 de octubre, 2017

1 Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje C. Se exigirá que el código se presente de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de su programa junto a cualquier indicación que sea necesaria y el makefile del código.

2 Tarea

En esta ocasión se le solicita programar el famoso juego de cartas UNO™, el cual constará de 4 jugadores. Para esto su programa deberá ser capaz de crear 3 procesos hijos (*fork*) los cuales cumplirán con el rol de jugadores (los 3 hijos más el padre). Las jugadas deberán ser comunicadas entre jugadores mediante el uso de pipes, y deben interactuar con el mazo de cartas, el cual deberá estar en memoria compartida.

3 Reglas

3.1 Objetivo del juego

El objetivo de UNO es deshacerse de todas las cartas que se “roban” inicialmente, diciendo la palabra “UNO” cuando queda la última carta en la mano. El primero que elimina todas las cartas de su mano, gana.

3.2 La preparación del juego

El mazo consta de 108 cartas que se deben barajar. Se deben repartir 7 a cada jugador. La primera carta se revela y se dispone como la zona de juego.



3.3 Desarrollo del juego

El primer jugador pone una carta de su mano. Una carta sólo se puede superponer en una carta del mismo color o del mismo tipo. Si se pone una carta que no cumpla esto, se debe robar una carta. Las cartas negras se pueden superponer sobre cualquier carta. Si un jugador no puede poner la carta oportuna, tiene que tomar una carta del mazo. Puede jugar esta carta ahora mismo si existe la posibilidad. Si no, pasa a ser el turno del siguiente jugador. Quién pone su penúltima carta, debe decir “UNO” y señalar que tiene sólo una última carta en la mano. El ganador de la ronda es quien pone su última carta. El juego termina cuando se acaba el mazo o cuando existe un ganador.

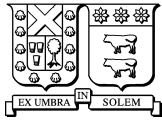
3.4 Cartas

Los colores existentes son: azul, rojo, verde, amarillo.

- Cartas 1-9: 2 de cada color.
- Cartas 0: 1 de cada color.
- Cartas +2: 2 de cada color.
- Cartas de reversa: 2 de cada color.
- Cartas de salto: 2 de cada color.
- Cartas de colores: 4 total de color negro.
- Cartas +4: 4 en total de color negro.

3.5 Cartas Especiales

- Carta +2: Cuando se pone esta carta, el siguiente jugador debe tomar dos cartas y no puede jugar en su turno (se salta). Esta carta sólo se puede superponer en una carta con el color apropiado u otras cartas “+2”. Si se revela al principio del juego, las mismas reglas aplican.
- Carta de reversa: Con esta carta se cambia la dirección del juego. Es decir que el próximo turno será de la persona que se encuentre en lado contrario y el juego continuará en ese sentido (como cambiar la dirección de las agujas del reloj). La carta sólo se puede superponer en una carta con color correspondiente o en otra carta de reversa. Cuando esta carta se toma en el principio del juego, el primer jugador comienza y entonces el jugador de la dirección contraria continúa.
- Carta de salto: Después de poner esta carta, el siguiente jugador no podrá jugar en su turno (lo salta). La carta sólo se puede superponer en una carta con color correspondiente o en otra carta de salto. Cuando esta carta se toma en el principio del juego, el jugador se “salta” su turno y le toca al próximo jugador.
- Carta (negra) de colores: Con esta carta el jugador decide qué color sigue en el juego (este no puede ser negro). Una carta de elección de colores se puede poner sobre cualquier carta y se puede jugar cualquier carta sobre esta (que cumpla con el color elegido). Si se toma una carta de elección de colores en el principio del juego, el jugador deberá elegir un color.
- Carta (negra) +4: el demonio. El jugador decide qué color sigue en el juego (este no puede ser negro). Además, el siguiente jugador debe robar cuatro cartas y no puede jugar en su turno (se salta). Una carta de +4 se puede poner sobre cualquier carta y se puede jugar cualquier carta sobre esta (que cumpla con el color elegido). Si se toma una carta de +4 en el principio del juego, las mismas reglas aplican.



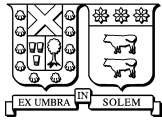
3.6 Consideraciones

- Tanto el mazo como la carta boca arriba deben estar en memoria compartida para el acceso de todos los jugadores y se debe mostrar su estado en todo momento.
- Los turnos de los jugadores deben comenzar en el orden $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow \dots$.
- En el turno de cada jugador deberán verse las cartas que quedan en su mano respectiva.
- Cuando a un jugador le quede una carta en la mano, los demás jugadores deberán tener conocimiento de esto. Para ello, a cada jugador le deberá aparecer un aviso declarando a cual jugador o jugadores les queda sólo una carta.
- Las jugadas deben ser comunicadas a los jugadores mediante el uso de pipes. Pueden hacerlo del método que prefieran mientras cumplan esta regla.
- Toda regla que usted conozca del juego y que no aparezcan en este documento deben ser **ignoradas**.
- La información a mostrar puede entregarla en cualquier formato mientras se entienda correctamente y se puedan realizar las jugadas.
- Frente a errores a la hora de ingresar jugadas, el programa debe ser capaz de resolicitar la información.

4 Consideraciones para la entrega

- Se deberá trabajar de forma individual. Se deberá entregar en Moodle a mas tardar el día 05 de Noviembre del 2017 a las 23:55 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje C o C++. Se asume que usted sabe programar en alguno de estos lenguajes.
- No se puede aplicar materia no vista en clases a excepción de investigación para memoria compartida (esto considera capitulo 1, 2 y 3).
- Los archivos deberán ser comprimidos y enviados juntos en un archivo .tar.gz en el formato TAREA2_ ROL.
- Las preguntas deben ser hechas por *Moodle*
- Si no se entrega Makefile o README, o si su programa no compila, la **nota es 0** hasta la corrección.
- Se **descontarán** 50 puntos por:
 - Mala redacción en el README.
 - No funcionamiento de Makefile.
 - Solicitar software externo o norcoreano¹ para el funcionamiento de la tarea.
 - No respetar el formato de entrega.

¹Creditos a Profesor Andrés Moreira



5 Calificación

Las tareas, además de ser revisadas por la evaluación correspondiente, podrán sufrir cambios si el alumno es interrogado por la tarea.

5.1 Reglas del Interrogatorio

- Se interrogarán 30 alumnos al azar los cuales serán anunciados el día 6 de Noviembre.
- Se realizarán 3 preguntas relacionadas a la entrega (Códigos, Algoritmos, etc).
- Si el alumno responde 3 preguntas correctamente, su nota se multiplicará por 1.1.
- Si el alumno responde 2 preguntas correctamente, su nota se multiplicará por 1.
- Si el alumno responde 1 preguntas correctamente, su nota se multiplicará por 0.8 y se pondrá bajo sospecha de copia.
- Si el alumno no logra responder las preguntas, su nota se multiplicará por 0.