Birzeit University - Faculty of Engineering and Technology
Electrical & Computer Engineering Department - ENCS5140 - Final exam
Real-Time Systems and Interfacing Techniques Laboratory - RT part - 1^{st} semester - 2023/24 - duration: 90 minutes

Instructor: Dr. Wasel Ghanem, Dr. Hanna Bullata, Mr. Mohammad Abu Zeina

Student Name/ID: _____

Playing card simulation

We would like to build a combined multi-processing and multi-threading application that simulates the behavior of a group of 4 players playing cards. The 4 players make 2 teams: Players 1 & 3 are part of the *odd team* while players 2 & 4 are part of the *even team*. Each team players are seated facing each other. As such, players 1 & 3 are facing each other while players 2 & 4 are facing each other.

The card deck is composed of 52 cards and these cards belong to one of the 4 following colors: Red, Green, Blue or Yellow. Each color has thus 13 cards numbered 1 to 13. For each card color, the card that contains the number 13 is of course higher than the remaining cards of the same color.

The simulation behaves as follows:

A. Initially player 1 owns the card deck and thus is responsible to shuffle the cards and then distribute 13 consecutive cards to each player starting by the player to his right (e.g. player 2).

B. When the cards are distributed, each player sorts the cards he has by color (e.g. 4 sorted arrays for the 4 card colors per player).

C. Depending on the sorted card colors, each player will make decisions based on the cards he has: If he doesn't have at least 7 cards of the same color, he'll announce that he'll pass for that round. Otherwise, he'll announce that he'll be able to make at least 7 or 8 or 9 or 10 or 11 or 12 or 13 sub-rounds depending on the number of cards of the same color he has.

The announcement by players occurs in the same order cards have been distributed to the players during that round.

D. Once all players have announced the number of sub-rounds they can make, the player who has announced the higher number of sub-rounds will declare the *round strong color* for the 13 sub-rounds that are about to start (e.g. he might say the Red color is the strong color). We'll call that player the *announcer* for that round. When the announcer declares the strong color it means that even the lowest card of that strong color becomes higher than any other card in the remaining 3 colors.

The announcer then puts on the table the highest card for one of the card colors he has. The selection of the first card color to start with is random.

E. Once step **D** above is done, the 3 remaining players should each put on the table a card according to the below logic table. The order of playing is round robin from the right, one player at a time. As such, the next player will see the cards that have been put on the table by the players that preceded him.

The playing logic is as follows:

- If a player does have a card of the same color as played by the announcer but of higher value, he should put on the table the highest card of that color.
- If the player does not have a card of the same color, then he puts on the table the lowest card of the round strong color.
- If he doesn't have any card of the round strong color, he just puts the lowest card of a random color he picks.

F. When the 4 players have put the 4 cards of a sub-round on the table, either the player who has put the highest card color as the first sub-round player wins the sub-round or the one who has put the highest strong color card. As such, the round winner will be the one to start the next sub-round with the highest card color of his choice. The same logic described above applies in the sense that the remaining 3 players should follow the card selection from the cards they have as described in step **E** above.

G. When the 13 sub-rounds have been played, the sub-rounds won by players 1 & 3 are summed together while the sub-rounds won by players 2 & 4 are summed together. We then check if the announcer was able to make the number of sub-rounds he promised to make before playing the round. If he was able to make at least the announced number of rounds, then the team to which the announcer belongs has won the round and we add to the team's score the number of won sub-rounds. Else, the team to which the announcer belongs has lost the round and we subtract from the team's score the number of announced sub-rounds and add the number of sub-rounds the other team was able to make to their score.

H. To start a new round, the player seated to the right of the one who shuffled and distributed the cards in the previous round is now responsible to shuffle the cards and then distribute 13 consecutive cards to each player starting by the player to his right.

I. Go to step **B** and continue playing.

J. The simulation ends if any of the following is true:

- One the teams has reached a score of at least 61. We'll call that team the game winner.
- One the teams has reached a score below -61. We'll call that team the game loser.
- The simulation has been running for more than one hour.

What you should do

- We would like to implement the above problem on Linux machines using a combined multi-processing and multi-threading approach. The teams are processes while the players of each team are threads that belong to a particular team process.
- In few bullet table items, describe in human language how you would implement the above described problem (20 steps max). Name the file `description.txt`. If you use chatgpt tool or other LLM tools to generate the description, you'll get a zero! **(15 points)**
- Using C-language, write the code for the team process in a file called `team.c`. No need to compile. Just focus on the code logic and make sure all included items behave as expected. **(25 points)**
- When the exam time is over, send the zipped folder that contains the above generated items as a reply to my message entitled **encs5140 final exam - RT part - 1stSem 2023/24**.