# CENG 305 – Operating Systems – Fall 2020

## **Project – 2** (Due 29/12/2020)

**Subject:** Playing a rock paper scissors game with **3 threads**.

**Aim:** The aim of this project is to use and create multiple threads inside a process and gain some experience about basics of parallel programming.

This is a teamwork assignment; you are going to work the same group as in project 1.

Cheating and studying together with the other project groups is strictly prohibited. A group should never show their codes to another group whether in seeking aid or giving it. A student may ask a fellow student general questions which do not pertain to a particular programming assignment but may not ask any programming aid.
Disciplinary action to be taken against cheating is arranged by the Rules and Regulations Governing Student Disciplinary Actions in Institutions of Higher Education.

**Algorithm:**

1. Main process will create 3 threads pthread_create()
2. 1. Thread will select an item randomly
3. 2. Thread will select an item randomly too
4. 3. Thread will select an item randomly too
5. Then they will compare their items
6. Score of the winning thread will be increased
7. Game will continue until one of the threads reaches 5 points.

The game has only two possible outcomes: a draw, or a win for one player and a loss for the other. A player who decides to play rock will beat another player who has chosen scissors, but will lose to one who has played paper; a play of paper will lose to a play of scissors. If both players choose the same shape, the game is tied and is usually immediately replayed to break the tie.

In addition to the previous project, now you must keep the counter for each item. The counter must be a shared object and it stores how many items (scissors, rock, paper) are selected during the game. You need to take necessary precautions for data consistency for this data and any other data in your project if it is necessary.

**Requirements:**

1. A 1 or 1.5 page report. The format of the report should be the same with **Group_XX_project-2_report.docx**. Change XX with your group ID.
   a) The should contain a summary of your project, implementation details, how you managed to synchronization between threads, what is the most changeling part.
   b) The screenshots of outputs of your program after running your program 3 times successively. The scores of each game should be different since they are based on totally random decisions. Take necessary actions to produce random decisions for every game. That is, ensure that the score for two consecutive games will always be different.
   **c) Convert it to a PDF file with the same name. (-20 points)**

2. Your program should adopt POSIX pthread implementation we covered in the class.

3. Writing and reading directly from a file is strictly prohibited. (gets 0 point)

4. You will use C programming language in Linux environment. Your program consists of a single file called **mtgame.c** . Your program will not take any input. It should be compiled with **gcc –lpthread** .

5. You will upload your homework to aybuzem.aybu.edu.tr after compressing them in a single ZIP file name **groupXX_project2.zip**. The compressed ZIP file should include:
   a) mtgame.c C source file.
   b) a file which includes compilation commands that you used (gcc –lpthread … -o ….. etc.), that is, a text file that has just one line.
   c) Project report in PDF format.

6. The codes that give compilation error or run-time error will get **0.**
7. Only one member of a group is required to submit the project. (-10 points)

**Example:**
Here are some example executions for the project.

```
>> gcc mtgame.c –lpthread -o mtgame
>> ./mtgame

The game has launched
3 threads will be created
The game starts
--
--
1. Turn, 1.Thread: ROCK, 2.Thread: ROCK, 3.Thread: SCISSORS
Draw, Score: 0 – 0 - 0
--
--
2. Turn, 1.Thread: PAPER, 2.Thread: ROCK, 3.Thread: SCISSORS
Draw, Score: 0 – 0 - 0
--
--
3.Turn, 1.Thread: PAPER, 2.Thread: PAPER, 3.Thread: SCISSORS
3. Thread win, Score: 0 – 0 - 1
--
--
4.Turn, 1.Thread: ROCK, 2.Thread: PAPER, 3.Thread: ROCK
2. Thread win, Score: 0 – 1 - 1
--
--
5.Turn, 1.Thread: SCISSORS, 2.Thread: PAPER, 3.Thread: SCISSORS
Draw, Score: 0 – 1 - 1
--

…
…
…
--
23. Turn, 1.Thread: PAPER, 2.Thread: PAPER, 3.Thread: SCISSORS
3. Thread win, Score: 3 – 2 - 5
--
--
3.Thread has won the game with score: Score: 3 – 2 – 5 in 23 Turns.
--
--
how many times items were selected: SCISSORS: 33, PAPER: 29, ROCK: 41
--
--
1.Thread terminated
2.Thread terminated
3.Thread terminated
Threads are joined by main process
Game finished
```

Please do not try to put some fake results in your report. TA and I will test, run and analyze your projects separately on our computers.
Good luck!