CENG 305 – Operating Systems – Fall 2020

Project – 1 (Due 24/11/2020)

Subject: Playing a rock paper scissors game with processes.

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

This is a teamwork assignment; you are going to work in a group of 5 or 6 students. If you could not find any group, we will assign you in a random group.

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 a new child process with fork()
2. Child process will select an item randomly
3. Parent process will select an item randomly too
4. Then they will compare their items
5. Score of the winning process will be incremented
6. Game will continue until one of the process 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.

For process communications you should handle inter-process communication (IPC) between processes by yourself. Please pay an extra attention for communication part which is the most important portion for this assignment.

Requirements:

1. A 1 or 1.5 page report. The format of the report should be the same with Group_XX_project-1_report.docx. Change XX with your group ID.
   a) The should contain a summary of your project, implementation details, which IPC is used, 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 one of the IPC mechanisms that 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 **game.c** . You program will not take any input. It should be compiled with **gcc**.

5. Your program should call **fork()** system call.

6. You will upload your homework to aybuzem.aybu.edu.tr after compressing them in a single ZIP file name **groupXX_projectPart1.zip**. The compressed ZIP file should include:
   a) game.c C source file.
   b) a file which includes compilation commands that you used (gcc … -o ….. etc.), a text file that has just one line.
   c) Project report in PDF format.

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

Example:
Here is some example executions for the project.

```
>> gcc game.c -o game
>> ./game

The game has launched
Child process is created
The game starts
--
Turn 1, Parent: ROCK, Child: ROCK
Draw, Score: 0 – 0
–
Turn 2, Parent: SCISSORS, Child: ROCK
Parent win, Score: 1 – 0
--
Turn 3, Parent: ROCK, Child: PAPER
Child win, Score: 1 – 1
–
Turn 4, Parent: PAPER, Child: SCISSORS
Child win, Score: 1 – 2
–
Turn 5, Parent: ROCK, Child: ROCK
Draw, Score: 1 – 2
–
…
…
…
--
Turn 17, Parent: PAPER, Child: ROCK
Parent win, Score: 5 – 4
--
Parent has won the game with score: 5 – 4 in 17 Turns.
--
Child has terminated
Parent waits child with wait()
Parent has termianted
```

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!