

**Project #1**  
**Signals under Unix/Linux**  
**Due: March 26, 2021**

**Instructor:** Dr. Hanna Bullata

## Guessing Game Simulation

We would like to create a multi-processing application that simulates a guessing game between 2 processes using signals. A parent process will create 2 children processes. We'll call the first child process  $P_1$  and the second child process  $P_2$ .

The behavior of the whole system should be as follows:

1. Upon creation, the children  $P_1$  and  $P_2$  will be sensitive to signal `SIGUSR1`. The parent process will be sensitive to both signals `SIGINT` and `SIGQUIT`.
2. The parent process will ask  $P_1$  and  $P_2$  to pick a random integer number in the range `[1...100]`.
3. Once they pick their numbers,  $P_1$  will write its picked number in the file `P1.txt` while  $P_2$  will write its picked number in the file `P2.txt`.
4. Once  $P_1$  finishes writing the number it picked in the file `P1.txt`, it informs the parent process.
5. Once  $P_2$  finishes writing the number it picked in the file `P2.txt`, it informs the parent process.
6. Upon being informed by  $P_1$ , the parent process will read the picked number in file `P1.txt`.
7. Upon being informed by  $P_2$ , the parent process will read the picked number in file `P2.txt`.
8. The parent process declares the round winner by comparing the numbers picked by  $P_1$  and  $P_2$ . It increments a counter associated with the round winner process. If by chance  $P_1$  and  $P_2$  have both picked the same number, the parent process increments the counters of both  $P_1$  and  $P_2$ . Afterwards, the parent process deletes the files `P1.txt` and `P2.txt`. This ends the current round.
9. Go to step 2. above unless the counter of either  $P_1$  or  $P_2$  has reached 10.
10. The parent process declares the winner of the game then kills  $P_1$  and  $P_2$  and exits. If both counters are 10, then both  $P_1$  and  $P_2$  are winners of the game.

## What you should do

- Write the code for the parent process in addition to processes  $P_1$  and  $P_2$ .
- Compile and test your program.
- Check that your program is bug-free. Use the `gdb` debugger in case you are having problems during writing the code (and most probably you will :-). In such a case, compile your code using the `-g` option of the `gcc`.

- Send the zipped folder that contains your source code and your executable(s) before the deadline. If the deadline is reached and you are still having problems with your code, just send it as is!