**Operating Systems Project - Exercise 1**

A screenshot of a computer program

Description automatically generatedTask 1

A black background with white text

Description automatically generatedThis output tells us a few things about mutual exclusion; since both instances of the file access stdout concurrently, we see mixed x-s and o-s. The rand()%3 and rand()%2 introduce a random factor to the sleeping time, which is why we cannot see a pattern in it.

Task 2

This output tells us how semaphores are used to control access to the output steam. In the code, it regulates so that only one process can print at a time by using sem\_wait to decrease the semaphore’s value and lock the section before printing and sem\_post to increase it and unlock the section. This is why the outputs are printed in pairs.

Task 3

A screen shot of a computer program

Description automatically generatedThis code simulates a game which is being played back and forth, by red posting “hei” semaphore and waiting for “hong” semaphore, and black posting “hong” semaphore and waiting for “hei” semaphore. This simulates the turn based gameplay of chess, which we can see is successful from the output. Each ‘player’ does 10 steps, with the last one being the end of the game where we can see that the red side won and the black side lost. The randomized processing time in the code simulates how in a real chess game, the players would take variable length time thinking on their moves. After the last move, each program closes and unlinks the semaphores.