Unix System

LemIPC

Contents

Administrative details 2

Subject 3

Technical constraints 4

Prohibited functions 5

Authorized functions 6

Administrative details

- Your sources shall be turned-in on the `PSU_year_lemipc` directory
  ex: PSU_2013_lemipc for the 2013-2014 scolar year
- Your project must compile with a Makefile.
- The project executable will be `lemipc`.
- This project has to be done in a group of two.

Subject

"Electronic Arts" (c) (hereinafter referred to as "EA"), small company specialized in import / export of maple syrup, has decided to widen its circle of activities in order to increase its power gain. To do this, they chose to get into the emerging market for video game applications.

EA does not intend to do it the easy way, they want to make a big splash, releasing the first innovative, immersive, addictive, and (most importantly) profitable game.

They now appeal to you to make this masterpiece which will leave a permanent trace in the history of video games.

Your mission is to develop a program meeting the following specificities:

- The goal is to fight players (spread over several teams) on a two dimensions game board.
- The last team remaining on the board is the winner.
- To kill a player, a minimum of two players from the same team (and different from the first) need to be in contact with the player to kill. It means on a square next to the one where the player you want to kill stands on. (including diagonally)
- The players will be able to move horizontally and vertically.
- When a player realizes he is surrounded by at least 2 players of the same opponent team, he must leave the board game (yes, because it is well known that NPC are honest in games, they scrupulously respect the rules ... yep, ask the mustachioed plumber on his kart if he cheats sometimes ... well nope!)
- A square cannot host more than one player simultaneously.
- Your binary will be launched this way: ./lemipc `PATH` `TEAM_NUMBER` with `PATH` being the path you give ftok and `TEAM_NUMBER` the team number of the current champion. `TEAM_NUMBER` will be greater than 0.

Technical constraints

- You are given a lempic.h file which you MUST store at the root of your repository. You are expected to read its content and follow the instructions inside carefully.
- Each client is a process and there should be only one executable, which means that the first player to start will create shared resources (shm, msgq, semaphores)
- Similarly, when a player leaves the game, he must check if he is the last player on the board. Indeed, the last process to quit is responsible for cleaning all IPCs created by the first process.
- The board must be stored in a shared memory segment (shm). Each player can see the contents of the board as he wishes, but to change it, it must respect the shared resource constraints and competitive access (semaphores).
- A player can only communicate with other players by msgQ.
- You must implement a team strategy of some kind
- On the board, a player can see if a square is empty or if it contains a player. In this case, it's the number of the team which is available. It is not possible to differentiate two players of the same team.
- You must display what's going on, on the board:
    - The interface must be done in text mode, and in this case, only the first player (the one who creates the board) displays the contents (the process then must continue to operate even after his death).

Prohibited functions

- None.

Authorized functions

- The C library.
- The system calls related to IPCs.