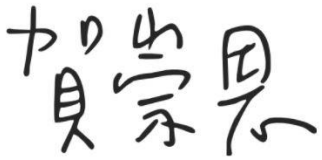
NTU Systems Programming HW5 Report
B09901027 賀崇恩



1. Please briefly tell us what bugs do you encounter during writing this homework. If no, tell us the most challenging part in this homework. (0.4pt)

我的 bug 大部分還是因為我在寫程式的時候會有的奇怪錯誤，比如用?: 的語法的時候搞不清楚條件是 true 對應到選左邊或右邊。不過最困難的部分應該是這是我寫過最需要自己搞得懂它在做什麼的程式，因為要很清楚程式 fork、哪些 code 是 child、哪些 pipe 是通往哪裡，這些東西的複雜程度都比以前演算法等程式還要大。

2. In this homework, you have to implement a fifo between two processes. Please tell us how you implement it in your code. Is your implementation good enough to prevent dead lock / race condition / busy waiting? If no, what is the better way to do it? (0.3pt)

我的實作方法如下的簡化版程式碼。
In player.c :
If(player_id <= 7) init player pssm;
Else make fifo, open fifo, read fifo
…
if player lose from zone A :
check fifo is created and opened via busy waiting;
write pssm to fifo;
如果 real player 先 access fifo，就會因為 agent player 還沒創 fifo 使得 real player busy wait 到 agent player 創 fifo 才能寫 fifo。這樣可以避免 real player 跟 agent player 對 fifo 的 race condition. 不過，這樣的做法是令 real player busy waiting 到 fifo 創好。一種處理這樣情況的方法是讓 real player 也可以創 fifo 然後在寫的時候 ignore SIGPIPE error.

3. In this homework, we use fifo to communicate between real players and agent

players. Is it possible to implement the communication with pipe instead of fifo? Why or Why not? Please briefly explain your answer. (0.3pt)

是，用 pipe 可以做到讓 real player 跟 agent 溝通，因為 real player 跟 agent 都有共同祖先(battle A)。不過實作起來應該會相對 fifo 麻煩很多，因為 pipe 在 fork 的時候會 duplicate 一份，之後還要把所有不需要的 pipe fd 關閉。