

MECI FOTBAL - LOVITURI LIBERE

1 Introducere - prezentarea problemei

Se consideră un meci de fotbal aflându-se la momentul loviturilor libere, astfel că doar un jucător va putea executa o lovitură liberă într-un moment de timp. Ceilalți vor aștepta la rând până când jucătorul curent va marca sau va rata lovitura. Un jucător poate urma la executarea unei lovituri libere numai în momentul în care cel dinaintea sa a executat lovitura.

1.1 Pas 1. Definire problemă

Condițiile sunt următoarele:

- Un singur jucător poate merge într-un moment de timp pentru a executa lovitura liberă.
- Un goalkeeper poate interveni sau poate încasa un gol.
- Un alt jucător poate executa o lovitură liberă doar în momentul în care goalkeeperul a intervenit sau nu.

2 Analiza problemei

2.1 Pas 2. Analiza cerințelor

Fiecare jucător este un fir de execuție diferit. Fiecare goalkeeper este un fir de execuție diferit. Secvențe posibile:

- Primul jucător merge la punctul cu var pentru a executa.
Al doilea jucător merge la punctul cu var pentru a executa.
Primul jucător execută lovitura liberă.
Goalkeeper-ul intervine(primește gol).
Al doilea jucător execută lovitura liberă.
Goalkeeper-ul intervine(primește gol).
- Primul jucător merge la punctul cu var pentru a executa.
Primul jucător execută lovitura liberă.
Goalkeeper-ul intervine(primește gol).
Al doilea jucător merge la punctul cu var pentru a executa.
Al doilea jucător execută lovitura liberă.
Goalkeeper-ul intervine(primește gol).

Secvențe imposibile:

- Primul jucător merge la punctul cu var pentru a executa.
Al doilea jucător merge la punctul cu var pentru a executa.
Primul jucător execută lovitura liberă.
Al doilea jucător execută lovitura liberă... - nu îndeplinește condițiile 1 și 3
- Primul jucător execută lovitura liberă.
Al doilea jucător execută lovitura liberă.
Goalkeeper-ul intervine(primește gol).
Goalkeeper-ul intervine(primește gol). - nu îndeplinește condiția 2 și 3.

3 Definirea structurii aplicației

În cazul problemei prezentate, taskurile sunt:

- task de tip Player (tipărind informațiile referitoare la execuția loviturii libere a jucătorului): Task Player
- task de tip Goalkeeper (tipărind informațiile referitoare la intervenția portarului sau la încasarea unui gol): Task Goalkeeper

4 Definirea soluției în vederea implementării

4.1 Pas 4. Soluție de implementare

Aleg mecanismele de sincronizare și comunicare între taskuri, prezentând organigramele taskurilor.

- aleg 3 semafoare binare, cu valorile inițiale :
player-asteapta=0, player-suteaza=0, goalkeeper-goal-nogoal=1.

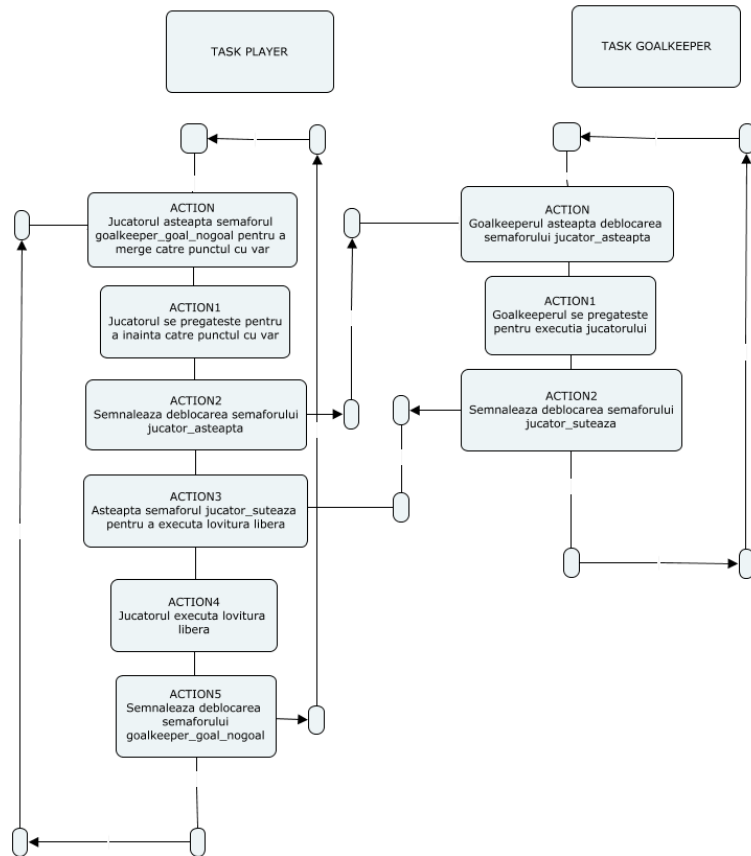


Figure 1: Soluție implementare - organigrame taskuri

5 Implementarea soluției

5.1 Cod program

```

#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <stdlib.h>

#define buff_dim 5 // Numarul executantilor loviturilor de la 11 metri
int a=1; // Variabila utilizata pentru contorizarea jucatorului ce urmeaza a executa

//Semafoarele utilizate
sem_t player_asteapta;
sem_t player_suteaza;
sem_t goalkeeper_goal_nogoal;

//Taskurile utilizate
void* Task_Goalkeeper();
void* Task_Player();

void* (*Taskuri[])() = {Task_Player, Task_Goalkeeper };

```

```

//Tasku-ul ce acopera executarea loviturilor de catre jucatori
void* Task_Player() {

int i;
for (i = 0; i < buff_dim; i++)
{
    sem_wait(&goalkeeper_goal_nogoal);
    printf("Jucatorul %d merge la punctul cu var pentru executarea loviturii de la 11
    metrii.\n", a);
    a = a + 1;
    sem_post(&player_asteapta);
    sem_wait(&player_suteaza);
    int x;
        x = rand() % 2;
    if (x == 0)
        printf("Goalkeeperul intervine\n\n");
    else if (x == 1)
        printf("Goalkeeperul incaseaza un gol.\n\n");
    sem_post(&goalkeeper_goal_nogoal);
}
}

//Task-ul ce acopera interventia sau incasarea unui gol a goalkeeperului in fata jucatorilor
void* Task_Goalkeeper() {

int i;
for (i = 0; i < buff_dim; i++)
{
    sem_wait(&player_asteapta);
    printf("Jucatorul executa lovitura de la 11 metri.\n");
    sem_post(&player_suteaza);
}
}

int main(void){
pthread_t Task[2];
int i;

// Initializarea semafoarelor
if(!sem_init(&player_asteapta,0,0)){
    else printf("Eroare la initializarea semaforului player_asteapta \n");
if(!sem_init(&player_suteaza,0,0)){
    else printf("Eroare la initializarea semaforului player_suteaza \n");
if(!sem_init(&goalkeeper_goal_nogoal,0,1)){
    else printf("Eroare la initializarea semaforului goalkeeper_goal_nogoal \n");

// Crearea firelor de executie
for(i = 0; i<2; i++)
    if( pthread_create(Task+i, NULL, (void*)(*(Taskuri+i)), NULL) ){

        perror("pthread_create");
        return EXIT_FAILURE;
    }
}
}

```

```

}

for(i=0; i<2; i++)
    if(pthread_join(*(Task+i), NULL) != 0){

        perror("pthread_join");
        return EXIT_FAILURE;
    }

printf("\nMain: Task-urile s-au incheiet ... iesire din executie ... \n");

sem_destroy(&player_asteapta);
sem_destroy(&player_suteaza);
sem_destroy(&goalkeeper_goal_nogoal);

exit(0);
}

```

6 Testarea aplicației si validarea soluției propuse

Din rezultatul execuției se poate observa clar o afișare corespunzătoare pașilor precedenți , dar și a cerințelor aferente. Rând pe rând , fiecare jucător merge către punctul cu var pentru a executa, așteptând ca cel din față să termine , goalkeeperul intervenind sau încasând un gol în urma execuției. (e.g. în Fig. 2)

```
Jucatorul 1 merge la punctul cu var pentru executarea loviturii de la 11 metrii.
Jucatorul executa lovitura de la 11 metri.
Goalkeeperul incaseaza un gol.

Jucatorul 2 merge la punctul cu var pentru executarea loviturii de la 11 metrii.
Jucatorul executa lovitura de la 11 metri.
Goalkeeperul intervine

Jucatorul 3 merge la punctul cu var pentru executarea loviturii de la 11 metrii.
Jucatorul executa lovitura de la 11 metri.
Goalkeeperul incaseaza un gol.

Jucatorul 4 merge la punctul cu var pentru executarea loviturii de la 11 metrii.
Jucatorul executa lovitura de la 11 metri.
Goalkeeperul incaseaza un gol.

Jucatorul 5 merge la punctul cu var pentru executarea loviturii de la 11 metrii.
Jucatorul executa lovitura de la 11 metri.
Goalkeeperul incaseaza un gol.

Main: Task-urile s-au incheiet ... iesire din executie ...

...Program finished with exit code 0
Press ENTER to exit console.□
```

Figure 2: Rezultatul a 5 lovituri libere