Università degli Studi di Cagliari

Sistemi Operativi

Modulo di Laboratorio 14E

Esercizio 01: regole di gioco

- Scrivere in C con l'utilizzo delle ncurses e dei meccanismi di `fork` e `pipe` un **programma multiprocesso** che implementi il gioco "**vespa e contadino**"
- Il gioco si compone di **tre tipologie di oggetti sullo schermo**:
  - Il **contadino** (carattere **#**), controllato dall'**utente** tramite le **frecce direzionali**, inizia la partita dal **centro** dello schermo
  - La **vespa** (carattere **^**), controllata dalla **logica di gioco** e che si muove in **maniera casuale ma lineare**, inizia la partita dall'**angolo superiore sinistro** dello schermo
  - Delle **trappole** (carattere **x**), controllate dalla **logica di gioco** e che compaiono in posizioni casuali dello schermo a intervalli di tempo prestabiliti
- Ogni qualvolta il contadino è "punto" dalla vespa, il contadino **perde una vita**
- Ogni qualvolta la vespa "cade" in una trappola, il contadino **guadagna una vita**
- Il gioco **termina quando il contadino finisce le vite**

Esercizio 01: regole di gioco

- Definiamo in maniera più approfondita le regole appena presentate, in particolare:
  - Richiedendo che la **vespa si muova in maniera** “**casuale ma lineare**”, si intende che questa **può cambiare direzione**, ma lo fa **ogni N passi** nella direzione precedente, e non ad ogni suo passo (N deve essere casuale in un dato intervallo)
  - Le **trappole** presenti sullo schermo in ogni intervallo sono **esattamente 3**:
    - Bisogna assicurarsi che le trappole abbiano **posizioni distinte e non sovrapposte**
    - Ad ogni nuovo intervallo **le nuove trappole devono occupare posizioni diverse** dalle precedenti trappole appena rimosse
  - Il contadino ha **3 vite iniziali**, e può arrivare ad accumulare al **massimo 6 vite**
  - Il **giocatore vede** il **numero di vite** in suo possesso ed il **tempo di gioco**
  - A fine gioco viene mostrato un **messaggio di** “**game over**” **indicante anche il tempo** per cui il contadino è riuscito a sopravvivere alla vespa

Esercizio 01: dettagli implementativi

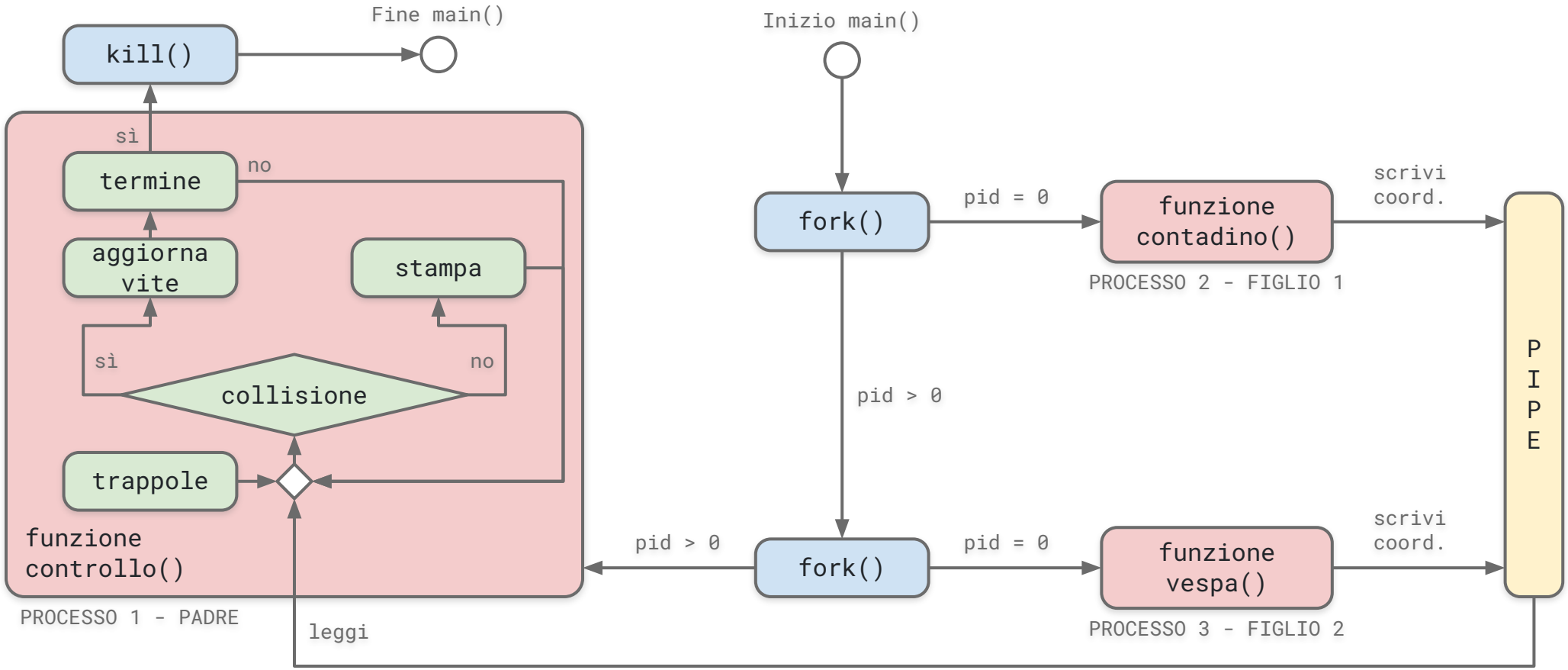
- Il programma richiesto deve far uso di **3 processi distinti** che comunicano tra di loro:
  - Il processo padre **riceve le comunicazioni** sulle nuove posizioni degli oggetti di gioco, **genera le trappole**, **stampa gli oggetti** stessi, **verifica le collisioni** e **controlla la terminazione**
  - Un primo processo **figlio acquisisce l'input utente e gestisce la posizione del contadino**, comunicandola al padre ogni volta che la aggiorna
  - Un secondo processo **figlio gestisce la posizione della vespa**, **generando gli spostamenti casuali ma lineari** e comunicando la posizione aggiornata al padre
- Definire delle **funzioni dedicate per la logica del padre e dei processi figlio**
- Utilizzare una **singola pipe per le comunicazioni** tra processi padre/figlio
- Gli eventi di gioco devono essere adeguatamente **temporizzati**

Esercizio 01: pseudocodice

```
#include <stdio.h>
#include <curses.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    Inizializzo_pipe();
    Creo_processo_figlio(Vespa);
    Creo_processo_figlio(Contadino);
    Funzione_Controllo();
    Termino_processi();
    Stampo_messaggio(TempoDiGioco);
    Exit();
}

Contadino() {
    while(1) {
        Leggo_tasti_cursore();
        Aggiorno_coordinate_oggetto();
        Comunico_coordinate_con_pipe();
    }
}
```

```
Vespa() {
    while(1) {
        Genero_movimento_lineare_casuale();
        Aggiorno_coordinate_oggetto();
        Comunico_coordinate_con_pipe();
        Sleep(Temporizzazione);
    }
}

Funzione_Controllo() {
    while(Vite > 0) {
        Leggo_coordinate_dalla_pipe();
        Rilevo_tipo_oggetto();
        Pulisco_posizione_precedente_oggetto();
        Stampo_nuova_posizione_oggetto();
        Genero_e_visualizzo_trappole();
        if(Collisione) {
            TipoCollisione = Definisci_collisione();
            Aggiorna_e_stampa_vite(TipoCollisione);
        }
    }
}
```

Esercizio 01: diagramma di flusso



Fine Modulo 14E