

## ***RELAZIONE SISTEMI OPERATIVI***

***Codice Corso: MFN0601***

***Data: 18/01/2020***

***Relatori: Cortosi Vittorio (Matricola: 884816), Dezzani Alessandro (Matricola: 885110)***

---

### ***Index***

- *Struttura*
- *Scelte Implementative*
  - *Old IPCs*
  - *Strutture utilizzate*
  - *Oggetti IPC utilizzati*
  - *Movimento delle pedine*
  - *Posizionamento pedine e bandierine*
- *Svolgimento del round*

## ***STRUTTURA***

---

*Il programma da noi creato è strutturato secondo una divisione in moduli, nel seguente modo:*

- *Master.c*
  - *Contenente il codice eseguibile (il main) ed eventuali funzioni utili/indispensabili inerenti alla scacchiera e alle pedine (movimento...).*
- *Semx.c*
  - *Modulo contenente funzioni per la gestione dei semafori (inizializzazione, estrazione risorsa, rilascio risorsa, attesa zero, etc...)*
- *Handling.c*
  - *Contenente codice per la gestione dei segnali*
- *Utilities.c*
  - *Contenente:*
    - *Tutte le macro richieste per l'implementazione del progetto*
    - *Tutte le strutture dati necessarie per lo svolgimento del programma (struct, static variable..)*
    - *Funzioni generali di utilità per lo svolgimento del programma.*
    - *Macro aggiuntive, oltre a quelle richieste.*
- *Makefile*
  - *File per il comando Make, contenente le specifiche ed i target di compilazione (-std=c89 -pedantic)*

## SCELTE IMPLEMENTATIVE

---

- *Old IPCs*
  - È stato deciso di implementare un metodo per la rimozione di Oggetti IPC rimasti appesi da esecuzioni fallite/terminate in modo anomalo:
    - L'implementazione prevede l'utilizzo di File Descriptor, ai fini di creare un file temporaneo in: `/tmp/`, chiamato: “**old\_ipcs.txt**”, che conterrà le `key_t` degli oggetti IPC aperti e non chiusi in un'esecuzione precedente.
  - Questo è stato possibile attraverso la definizione ed implementazione di tre functions:
    - `get_ipcs`
      - Quest'ultima function è utilizzata in fase di creazione di oggetti IPC. Quando un oggetto viene staccato, viene salvato (attraverso file descriptor) in `/tmp/old_ipcs.txt`.
    - `check_old_ipcs()`
      - Controlla che ci sia effettivamente il file, se esiste controlla che non sia vuoto, nel caso in cui non fosse vuoto, pulirà tutti gli oggetti IPC attraverso l'estrazione delle `key_t` trovate nel file.
    - `reset_ipcs_fd`
      - Una volta puliti gli oggetti IPC rimasti appesi, questa funzione nello specifico, pulisce il file.
  - Le ultime due funzioni sono richiamate in principio di tutta l'esecuzione, in modo tale di controllare prima di tutto se c'è qualche segmento di memoria aperto e lasciato appeso e pulire il file dopo appunto averlo processato
- *Strutture utilizzate*
  - È stato deciso di implementare tre strutture fondamentali:
    - Una che conterrà la matrice di **SO\_ALTEZZA** e **SO\_BASE**
    - Una che conterrà i dettagli per ogni giocatore (`pid`, `score`...)
    - Una che conterrà i dettagli per ogni FLAG (owner, posizione...)
- *Oggetti IPC utilizzati*
  - Gli oggetti IPC utilizzati sono stati i seguenti:
    - Un `SEM_SET` che conterrà solo semafori di sincronizzazione tra processi
    - Un `SEM_SET` di (`SO_BASE*SO_ALTEZZA`), che conterrà tutte le posizioni della scacchiera
    - Una `SHMEM` per la scacchiera
    - Una `SHMEM` contenente le strutture dei giocatori
    - Una `SHMEM` contenente le strutture delle bandierine
- *Movimento delle pedine*
  - Il programma, e le pedine, funzioneranno in maniera del tutto autonoma. Una volta mandato in esecuzione ed aver piazzato bandierine (da parte del master) e pedine (da parte dei giocatori), le pedine si muoveranno in maniera autonoma seguendo un algoritmo implementato.

- *Posizionamento pedine e bandierine*
  - *Sia le pedine che le bandierine sono posizionate in maniera del tutto randomica sulla schacchiera.*
- *Segnali Gestiti*
  - *Si è deciso di gestire solamente due segnali:*
    - **SIGINT**, in quanto durante l'esecuzione non si vogliono interruzioni da parte dell'utente in linea di comando (CTRL-C)
    - **SIGALARM**, poiché se il round dura più di **SO\_MAX\_TIME**, il master deve concludere il gioco.

## **SVOLGIMENTO DEL ROUND**

---

*Il round è svolto seguendo le indicazioni richieste dal documento ad eccezione dell'inizio di un nuovo round: una volta concluso il round e stampato lo stato della schacchiera (ASCII) e i punti dei giocatori, verrà chiesto all'utente se vuole che si svolga un altro round attraverso una semplice risposta (Y/N).*