# Progetto SO 2023/24

# Reazione a catena

## INTRODUZIONE

Il progetto è stato sviluppato utilizzando l'utility **make**, che ha permesso di compilare separatamente i vari processi richiesti. Ho organizzato il progetto in diversi moduli, ciascuno con una specifica responsabilità:

* **Master**
* **Config**
* **Attivatore**
* **Alimentazione**
* **Atomo**
* **Inibitore**

In aggiunta, nella cartella **src** ho incluso alcune utility per supportare lo sviluppo:

* **Log utils** 🡪 Usato per scrivere sul terminale con diversi colori
* **Semaphore utils** 🡪 Per i semafori
* **Structure utils** 🡪 Per le strutture dati

Ho idealizzato anche una piccola interfaccia grafica che tiene traccia di tutti i processi e della memoria condivisa, inoltre ha due pulsanti per inviare il segnale al master di accendere o spegnere l’inibitore.

## COMPILAZIONE

Il progetto può essere compilato in due modi:

* Make run 🡪 per compilare ed avviare il master senza la gui
* Make runGui 🡪 per compilare ed avviare il master con la gui

I file object vengono messi all’interno della cartella objects, mentre i file eseguibili vengono messi dentro bin.

## INTERFACCIA GRAFICA

L’avvio o meno dell’interfaccia grafica non influisce sul funzionamento, è importante però attivarla per mandare il segnale di accendere o spegnere l’inibitore. Ho deciso di sviluppare questa piccola interfaccia, anche se non richiesto, per aiutarmi durante la fase di sviluppo e di testing. Ho scoperto così che mi è stata davvero utile riuscendo a scovare alcuni bug all’interno del progetto.

## CONFIGURAZIONE

Come file di configurazione ho utilizzato i file .ini utilizzando la libreria inih per facilitarmi la lettura e l’accesso.

## APPUNTI SUL FUNZIONAMENTO

Tutti i processi vengono attivati dal processo master con execve che inizializza le strutture e fa iniziare la simulazione.   
Lo scambio di dati fra tutti i processi avviene grazie ad una memoria condivisa regolamentata da un semaforo che il master inizializza.  
Ho dovuto inoltre implementare due code di messaggi, rispettivamente:

* Fra Atomo e Attivatore
* Fra Atomo e Inibitore

Per quanto riguarda il processo attivatore esso invia agli atomi attualmente funzionanti / 2 il messaggio di sciindersi. Se c’è solo un atomo in funzione allora invierà il messaggio solamente a lui.

Ho prestato particolare attenzione allo scambio di messaggi fra i processi atomo ed il processo inibitore. Inizialmente stavo comunicando con gli atomi in modo casuale poiché stavo semplicemente scrivendo nella coda senza assicurarmi che quel determinato messaggio debba arrivare ad un determinato atomo e non al primo che prende il messaggio dalla coda. Accorgendomi di questa cosa ho fatto in modo di passare il PID di atomo nel messaggio in modo che al suo ritorno il messaggio venga captato solamente dall’atomo avente quel determinato PID.

Come richiesto il processo inibitore comunica all’atomo se sciindersi o meno e quanta energia eliberare, ho utilizzato queste due logiche:

* Sciindersi: c’è un 10% di probabilità che l’atomo venga messo subito fra le scorie e spento
* Energia da eliberare: se l’energia potenziale (energia attuale + energia che l’atomo vuole eliberare) è maggiore di 2/3 della soglia massima di energia (ENERGY\_EXPLODE\_THRESHOLD) allora l’atomo libererà tutta la sua energia.