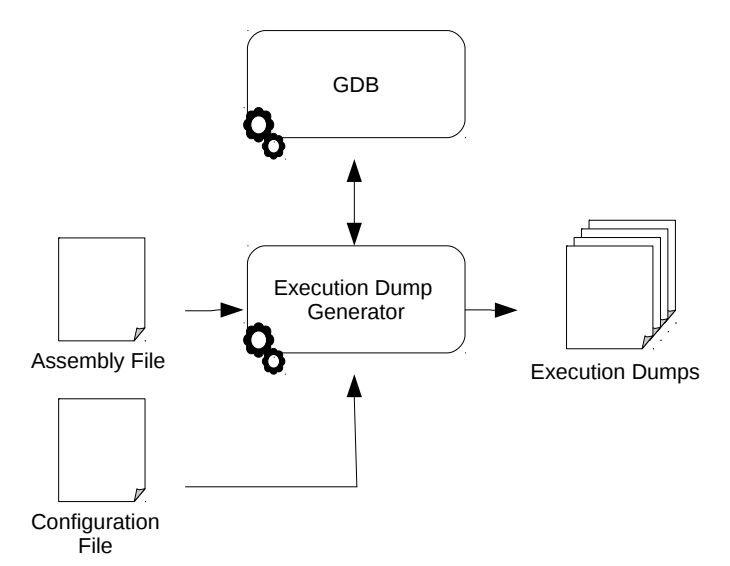
**Relazione Progetto Trojan Hardware Simulator**

**Registers Dump Generator**

Per la fase della gerazione del dump dei registri è stato utilizzato il debugger C GDB presente in ambiente Linux sul Raspberry. E’ stato fatto uso della possibilità di poter lanciare il debug su di un file C con l’opzione di poter aggiungere uno script associato che gestisse il processo di debug in automatico (--batch e –x script per indicare lo script).

Mentre lo script per la gestione del debug in GDB è strutturato in modo da abilitare i log dei risultati su di un file che di default si chiama “gdb.txt”. Successivamente viene indicato attraverso il comando “break main” di analizzare solo la funzione che si chiama “main”. Dopodiché viene fatto partire il debug seguito da un ciclo while(true) che ad ogni step lancia il comando “info registers” stampando il valore dei registri di nostro interesse a quel passo di esecuzione. Il ciclo, pur essendo infinito verrà interrotto quando si raggiungerà la fine del “main” dato che lo script verrà chiuso automaticamente.

Infine il comando per fare partire il debug in GDB viene lanciato da uno script bash con le relative opzioni sopra specificate.



Processo di generazione dei dump.

**Trojan Simulator Launcher**

Il Launcher del Simulatore si occupa di raggruppare tutti i componenti utili all’esecuzione della simulazione e di lanciarli nel corretto ordine.

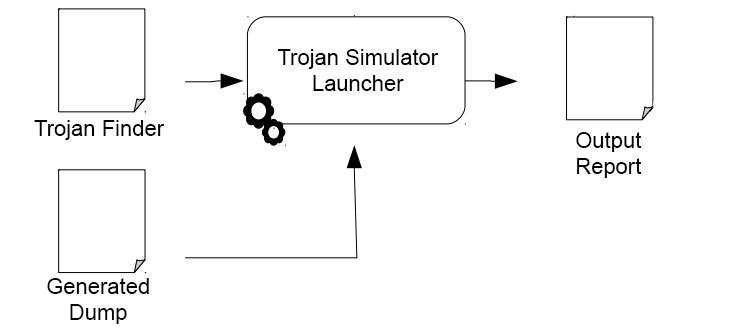
Il programma in Java è strutturato in modo da poter funzionare in futuro in caso lo si voglia eseguire su sistemi operativi diversi da Linux.

Per prima cosa si occupa di compilare il programma di ricerca dei Trojan tra i Trigger Generati.

Dopo esegue una serie di iterazioni di generazione di dump e di analisi sullo stesso, in numero specificate dall’utente come argomento del programma, altrimenti ne esegue 5.

All’interno di ogni iterazione il Launcher esegue lo script che genera il dump e successivamente lo fa analizzare dal programma in C di ricerca del Trojan. Si occupa anche di marcare all’interno del file output.txt, contenente il risultato del analisi, l’inizio di ogni iterazione con un numero progressivo associato.

Al termine delle iterazioni sarà presente il file output.txt contenente il risultato cercato e i file di dump rinominati in base alle iterazioni in cui sono stati generati.



Processo eseguito all'esecuzione del Trojan Simulator Launcher.

**Ordinamento Array in C**

Il programma in C di ordinamento di un array sarà il programma che andremo ad analizzare dal quale gereremo il dump dal debug della funzione main.

Il codice del programma consiste in un ordinamento utilizzando l’algoritmo BubbleSort. L’array da ordinare è di 10 elementi di tipo intero generati casualmente ad ogni sua esecuzione.