Architetture e Sistemi Operativi (Corso A e B) A.A. 2022-2023
Terza prova di verifica intermedia – Programmazione Assembler ARMv7

Esercizio obbligatorio

Si chiede di realizzare in Assembler ARMv7 le funzioni descritte di seguito

- **`int myatoi(char *p);`**

  che restituisce il valore intero corrispondente ad una stringa (terminata dal carattere NULL) che rappresenta un numero intero positivo. In caso di errore, per esempio dovuto alla presenza di caratteri illegali, la funzione restituisce –1;

- **`int map(char **v, int n);`**

  che calcola la somma dei valori ottenuti dividendo per 4 gli interi rappresentati dal vettore di *n* stringhe *v* (ad esclusione degli eventuali -1 ottenuti dalla traduzione con myatoi di stringhe non traducibili in interi positivi).

Si richiede di sviluppare un codice funzionante in cui entrambe le funzioni siano realizzate in modo da poter essere utilizzate da codice esterno, cioè rispettino tutte le convenzioni tipiche della programmazione ARMv7 (tra cui le regole sull'uso dei registri). Le due funzioni dovranno essere realizzate in due file testuali separati "**map.s**" e "**myatoi.s**" di cui potete scaricare una versione da riempire con le direttive all'assemblatore solite già impostate. Una volta complete dovete caricare i due file nel Google Forms e sottomettere entro la deadline. La correttezza delle vostre funzioni potrà essere valutata mediante il codice "**main.c**" allegato al compito e che potete scaricare. Usando il cross compilatore potete compilare il codice con questo comando

```
$ arm-linux-gnueabihf-gcc main.c map.s myatoi.s -static
```

Nota 1: nella codifica ASCII, i caratteri da 0 a 9 corrispondono ai codici esadecimali da 0x30 a 0x39.
Nota 2: qualora usiate l'istruzione MUL dst, src1, src2 ricordarsi che per questa istruzione il campo src2 può essere solo un registro non un immediato.

Esercizio opzionale

Si scriva il codice della funzione map come una funzione *higher order* con firma alternativa

`int map(char **v, int n, int (*f)(char *p));`

dove il terzo parametro rappresenta la funzione da calcolare su ciascuno degli elementi del vettore, tenendo conto che una funzione, di indirizzo noto e contenuto in un registro *Rfun*, può essere chiamata utilizzando l'istruzione BLX Rfun invece della BL che prende solo un'etichetta. Implementate questa funzione come file "**map_ho.s**" e sottomettetelo su Google Forms. Potete testare questa seconda opzionale versione della map con il codice main "**main_ho.c**" sempre allegato al compito.