



## Laboratorio di Architettura degli Elaboratori

A.A. 2016/17

### *Elaborato Assembly*

Si ottimizzi un codice in linguaggio C che effettua il monitoraggio di un impianto chimico industriale mediante l'uso di Assembly inline. Ricevendo come input il pH di una soluzione contenuta in un serbatoio e, una volta impostate le soglie minima e massima per il funzionamento ottimale, il programma fornisce in uscita lo stato della soluzione in termini di acido (*A*), basico (*B*) e neutro (*N*). Il sistema deve portare sempre la soluzione allo stato neutro, accettando un transitorio di 5 cicli di clock; pertanto si richiede che al sesto ciclo di clock in cui il sistema sia allo stato *A*, venga aperta una valvola con soluzione basica (*BS*), e analogamente se allo stato *B* si apra la valvola con soluzione acida (*AS*). Il sistema deve inoltre fornire in uscita il numero di cicli di clock da cui si trova nello stato attuale.

Il programma deve essere lanciato da riga di comando con due stringhe come parametri, la prima stringa identifica il nome del file `.txt` da usare come input, la seconda quello da usare come output:

```
$ ./controller testin.txt testout.txt
```

Il programma deve leggere il contenuto di `testin.txt` contenente in ogni riga i seguenti valori:

INIT, RESET, PH

- **INIT** [1]: valore binario, quando vale **1** il sistema è acceso; quando vale **0** il sistema è spento e deve restituire una linea composta da soli **0**.
- **RESET** [1]: quando posto a **1** il controllore deve essere resettato, ovvero tutte le uscite devono essere poste a **0** e il sistema riparte.
- **PH** [3]: valore del pH misurato dal rilevatore. Il range di misura è compreso tra **0** e **14** con risoluzione di **0,1**. Il valore è espresso in decimi di pH e sempre riportato in 3 cifre, ad esempio 065 corrisponde a 6,5.

Il programma deve restituire i risultati del calcolo in `testout.txt` in cui ogni riga contiene:

ST, NCK, VLV

- **ST** [1]: indica in quale stato si trova la soluzione al momento corrente (acida – *A*, basica – *B* o neutra - *N*)
- **NCK** [2]: indica il numero di cicli di clock trascorsi nello stato corrente.
- **VLV** [2]: indica quale valvola aprire per riportare la soluzione allo stato neutro nel caso in cui la soluzione si trovi da più di 5 cicli di clock in stato acido (*BS*) o basico (*AS*).

Si considerino i seguenti valori di soglia:

- pH < 6.0 : Acido
- $6.0 \leq \text{pH} \leq 8.0$  : Neutro
- pH > 8.0 : Basico



## Laboratorio di Architettura degli Elaboratori

A.A. 2016/17

Assieme al presente documento sono forniti:

- un file `controller.c` contenente il sorgente C che dovrà essere editato per inserire la parte assembly. **Il file può essere modificato solo nelle parti segnalate tramite commento!** Files contenenti modifiche esterne a queste sezioni saranno penalizzati e potranno comportare anche un voto insufficiente. È possibile utilizzare assembly inline e/o funzioni assembly richiamate da C, in quest'ultimo caso i files creati devono essere consegnati all'interno della cartella principale.
- un file `testin.txt` di esempio. In fase di valutazione sarà utilizzato un file di test diverso.
- un file `trueout.txt` da utilizzare per verifica del corretto funzionamento del programma. Sarà sufficiente usare il comando `diff testout.txt trueout.txt` per vedere eventuali differenze tra i due files (quello generato dal programma e quello corretto).

In fase di correzione, saranno valutati meglio progetti che ottimizzeranno meglio il codice, ovvero programmi che porteranno ad un maggiore guadagno in termini di velocità di esecuzione. Durante l'esame orale è possibile che venga richiesto di operare delle modifiche al codice sul momento.

### Modalità di consegna:

Tutto il materiale va consegnato elettronicamente tramite procedura guidata sul sito Moodle del corso. Indicativamente 15 giorni prima della data di consegna sarà attivata una apposita sezione denominata "consegna\_ASM\_mmmaaaa" (mmm=mese, aaaa=anno); accedendo a quella pagina sarà possibile effettuare l'upload del materiale. La consegna del materiale comporta automaticamente l'iscrizione all'appello orale.

Il giorno seguente la data ultima di consegna (entro le ore 12.00) verrà pubblicato sul sito Moodle del corso il calendario provvisorio degli orali; in caso di impossibilità a presenziare alla discussione orale nell'orario assegnato è necessario comunicarlo al docente via email entro le successive 24 ore. Entro la sera del giorno seguente sarà pubblicato il calendario definitivo.

Materiale da consegnare:

Si richiede ad ogni gruppo di caricare un singolo archivio .tgz denominato `asm_cognome1_nome1_cognome2_nome2.tgz`<sup>1</sup> contenente tutti i files di seguito elencati senza sottocartelle:

1. Il file `controller.c` modificato, contenente la versione definitiva del progetto da usare per i test. Non rinominare il file fornito ma modificarlo opportunamente
2. Tutti gli altri files sorgenti necessari al progetto (ad esempio funzioni assembly)
3. Makefile per la compilazione (non fornito, ogni gruppo deve scrivere il proprio)

---

<sup>1</sup> Per generare correttamente un file `filename.tgz` in linux seguire la seguente procedura:

- posizionare tutti i files da comprimere in una cartella
- rinominare la cartella "filename"
- uscire dalla cartella e lanciare il comando `tar cvfz filename.tgz filename`



## Laboratorio di Architettura degli Elaboratori

A.A. 2016/17

4. Un file `Relazione.pdf` con una relazione del progetto che affronti nel dettaglio almeno i seguenti punti:

- le variabili utilizzate e il loro scopo;
- le modalità di passaggio/restituzione dei valori delle funzioni create;
- il diagramma di flusso o lo pseudo-codice ad alto livello del codice prodotto;
- la descrizione delle scelte progettuali effettuate

Si ricorda che è possibile effettuare più sottomissioni, ma ogni nuova sottomissione cancella quella precedente. Ogni gruppo deve consegnare una sola volta il materiale, ovvero un solo membro del gruppo deve effettuare la sottomissione!