UNIVERSITA’ DEGLI STUDI DI VERONA

ELABORATO ASM

ARCHITETTURA DEGLI ELABORATORI

TONINI FRANCESCO (VR408686)

CAPOGROSSO LUIGI (VR408776)

SOMMARIO:

DESCRIZIONE DEL PROGETTO

FUNZIONI

VARIABILI UTILIZZATE

SCELTE PROGETTUALI

DESCRIZIONE DEL PROGETTO

Riprendendo la situazione presentata nello scorso elaborato, è ora compito quello di ottimizzare un codice in linguaggio C che effettua il monitoraggio di un impianto chimico industriale mediante l’uso di Assembly.

Il programma deve essere lanciato dalla linea di comando rispettando la seguente sintassi:

$ ./controller testin.txt testout.txt

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

* 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 [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 considerano i seguenti valori di soglia:

* pH < 6.0: Acido
* 6.0 <= pH <= 8.0: Neutro
* pH >= 80: Basico

FUNZIONI

La funzione che è stata implementata per il miglioramento del codice in linguaggio C è ph\_asm().

Tale funzione riceve in ingresso due array, per la precisione bufferin, bufferout\_asm e viene richiamata all’interno del file controller.c come un extern module con la seguente sintassi:

extern void ph\_asm(char in[], char out[])

Questa, può essere definita come il “main” del nostro programma.

VARIABILI UTILIZZATE

Nel file sorgente “ph\_asm.s” sono state invece utilizzate le seguenti variabili:

* start: verifica se è stata o meno raggiunta la fine dell’array. In caso di responso negativo verrà chiamata la funzione checkInit (per il controllo del campo init)
* checkIinit: verifica il valore di init, ovvero il primo della riga. Se questo risulta essere diverso da 0, ovvero il sistema è acceso, viene chiamata la funzione per il controllo di reset. Se init invece risulta essere uguale a 0, reimposta gli output e richiama la funzione stampa
* checkReset: verifica il valore di reset, ovvero il terzo della riga. Se questo risulta essere diverso da 0, ovvero il sistema è acceso, viene chiamata la funzione per il controllo del pH. Se reset invece risulta essere uguale a 1, reimposta gli output e richiama la funzione stampa
* checkPH: verifica il valore di pH
* checkBasic: elabora pH basico
* checkAcid: elabora pH acido
* checkNeutral: elabora pH neutro
* clear: vengono modificati nck e vlv a seguito di una variazione di pH
* print: si occupa della stampa su bufferout e dell’incremento degli indirizzi

SCELTE PROGETTUALI