

**CORSO DI ARCHITETTURA DEGLI ELABORATORI
PROGRAMMAZIONE ASSEMBLY: PROGETTO**

24 MAGGIO 2011

CONSEGNA Lunedì 11 Luglio 2011

Lo studente programmi nel linguaggio Assembly 8088 un micro 8088, corredato di 1MB di memoria e adeguate interfacce di I/O, al fine di realizzare quanto sotto descritto. Si utilizzi il linguaggio assembly descritto nel libro “Andrew S. Tanenbaum, *Architettura dei Calcolatori, Un Approccio Strutturato*, Pearson Prentice Hall, 5 ed., 2006.” e l’assemblatore a questo allegato o scaricabile dall’indirizzo internet <ftp://ftp.cs.vu.nl/pub/evert>

Si svolgano i quesiti che seguono. E’ SUFFICIENTE SVOLGERE ALMENO 1 QUESITO PER UNA VALUTAZIONE POSITIVA DELL’ELABORATO mentre i rimanenti concorrono al raggiungimento del massimo dei voti.

Si faccia uso di eventuali brevi commenti per indicare nel codice il quesito a cui si riferisce e si utilizzino gli stessi nomi per funzioni o procedure come in questo testo.

Si utilizzi, come visto a lezione, **lo stack per il passaggio degli operandi** (siano essi valori o puntatori ad aree di memoria) **tra sotto procedure e lo stack o il registro AX per ritornare il risultato**. Gli operandi denotati da “*” sono dei puntatori (es. vettore*).

QUESITO 1

Implementare la procedura:

void CREAALBERO (A*, dim)

che riceve in ingresso il puntatore alla struttura *A* di tipo albero binario con *dim* nodi e **costruisce** tale struttura in modo da associare ad ogni nodo gli indirizzi dei **due** nodi figlio da esso dipendenti.

Ogni nodo è formato da 3 parti: 2 puntatori ai due nodi figlio e 1 area di memorizzazione a 16 bit:

Puntatore al primo figlio	Puntatore al secondo figlio	Area di memorizzazione (16bit)
---------------------------	-----------------------------	--------------------------------

Il puntatore al figlio può contenere l’**indirizzo effettivo** della locazione di memoria che lo ospita, o uno **spiazzamento rispetto la radice dell’albero** – lo studente scelga la modalità che preferisce.

Le foglie, nodi che non hanno figli, avranno i due puntatori marcati con “0000...0”.

QUESITO 2

Il file di testo *input.txt* contiene una sequenza di numeri **decimali** distanziati da uno spazio bianco. Ogni numero è compreso tra 0 e 255.

Esempio, *input.txt* contiene: “10 2 34 120 3 4 ...”

Si implementi la procedura:

stato CARICA (A^* , dim)

che:

- 1- riceve in ingresso il puntatore alla struttura A di tipo albero binario con dim nodi, costruita come indicato al quesito 1;
- 2- copia dim numeri decimali da *input.txt* nella struttura A **convertendoli in intero a 16 bit**, memorizzando un intero in ogni nodo di A ;
- 3- restituisce lo stato di esecuzione (0 in caso di successo, 1 in caso di errore – ad esempio errori nella apertura, chiusura, lettura del file).

QUESITO 3

Similmente a quanto richiesto dal quesito 2, si implementi la procedura:

stato SALVA (A^*)

che:

- 1- riceve in ingresso il puntatore alla struttura A di tipo albero binario con dim nodi, costruita come indicato al quesito 1;
- 2- crea il file di testo *output.txt* e in esso copia ordinatamente il contenuto di A convertendolo in numeri **decimali** separati da spazi bianchi (*output.txt* se già esiste sarà svuotato);
- 3- restituisce lo stato di esecuzione (0 in caso di successo, 1 in caso di errore – ad esempio errori nella apertura, chiusura, scrittura del file).

QUESITO 4

Si implementi la funzione:

boolean COMPARANODO (N^* , t , s)

che riceve in ingresso il puntatore a un nodo N di albero binario, costruito come indicato al quesito 1, e gli interi a 16 bit t e s . La funzione compara il dato contenuto nel nodo N , all'interno **dell'area di memorizzazione a 16 bit**, con t e se è uguale lo sostituisce con s ; restituisce 0 in caso di successo, 1 in caso di errore.

QUESITO 5

Si implementi la funzione:

occorrenze TROVAESOSTITUISCI (A^* , t , s)

che riceve in ingresso il puntatore alla struttura A di tipo albero binario, costruita come indicato al quesito 1, e gli interi a 16 bit t e s . La funzione cerca in A tutte le occorrenze di t e le sostituisce con s ; restituisce il numero di occorrenze trovate.

La procedura è chiamata in maniera ricorsiva ed esegue il seguente algoritmo:

```
TROVAESOSTITUISCI (A*, t, s)
{
    occorrenze=0
    if (la radice di A = t)
        [sostituisci con s e incrementa occorrenze di 1]
    if (A ha il primo figlio)
        occorrenze = occorrenze + TROVAESOSTITUISCI (FIGLIO1*, t, s)
    if (A ha il secondo figlio)
        occorrenze = occorrenze + TROVAESOSTITUISCI (FIGLIO2*, t, s)
    return occorrenze
}
```

In sostanza TROVAESOSTITUISCI invoca se stessa sui due sotto alberi figlio fino ad arrivare alle foglie.

Si faccia uso della funzione COMPARANODO, sviluppata al quesito 4, per comparare e sostituire il contenuto di un nodo.

NOTE PER LO SVOLGIMENTO DEL COMPITO

- **Quanto non è specificato in questo testo fa parte delle scelte progettuali dello studente. Si riportino le scelte, commentandole, nella relazione.**
- Le date della discussione orale saranno pubblicate nella pagina <http://iafelice.web.cs.unibo.it/archi/>

NOTE SULLA VALUTAZIONE DEL COMPITO

La valutazione del progetto dipende strettamente dai seguenti vincoli:

- Il progetto deve essere consegnato entro le 24.00 della data indicata nella prima pagina.
- Saranno attribuiti dei punti ai seguenti aspetti generali:
 - 1) **Presenza di commenti.** Il codice deve essere commentato in modo da agevolarne la comprensione. Ogni funzione deve essere preceduta da commenti che ne chiariscono funzionalità, argomenti ed eventuali variabili interne.
 - 2) **Codice funzionante.** Il progetto deve essere **assemblabile e funzionante altrimenti NON sarà corretto.** Indicare nella relazione i **quesiti svolti** e il **sistema operativo usato.**
 - 3) **Codice ottimizzato e strutturato.** Saranno valutate positivamente implementazioni che ottimizzano il codice. L'ottimizzazione deve riguardare anche i commenti: chiari e concisi. Il codice deve essere opportunamente strutturato facendo uso di cicli e chiamate a sotto procedure.
- **Relazione** di max. 1 pagine per facilitare la lettura del codice e **comprendere le scelte progettuali.** Evitare copertine, indici, glossari e quanto NON-NECESSARIO o di abbellimento.
- **Lavoro di gruppo.** Si consiglia di svolgere il progetto in gruppi di lavoro. I gruppi devono essere composti al **massimo da 3 persone.** Ogni elemento del gruppo deve conoscere scelte progettuali e dettagli implementativi. E' utile il confronto tra gruppi. Tuttavia ogni gruppo deve fare le proprie scelte progettuali evitando di copiare integralmente quelle degli altri: saranno valutati **negativamente** progetti simili.

MODALITA' DI CONSEGNA

Consegna on-line alla pagina: <http://iafelice.web.cs.unibo.it/archi/>