

Università degli Studi di Verona
Dipartimento di Informatica
Corso di Laurea in Informatica

ELABORATO ASM
Architettura degli elaboratori

Marco Colognese VR386474

INDICE

- Presentazione _____ pag. 3
- Variabili utilizzate _____ pag. 5
- Funzioni _____ pag. 7
- Scelte progettuali _____ pag. 11

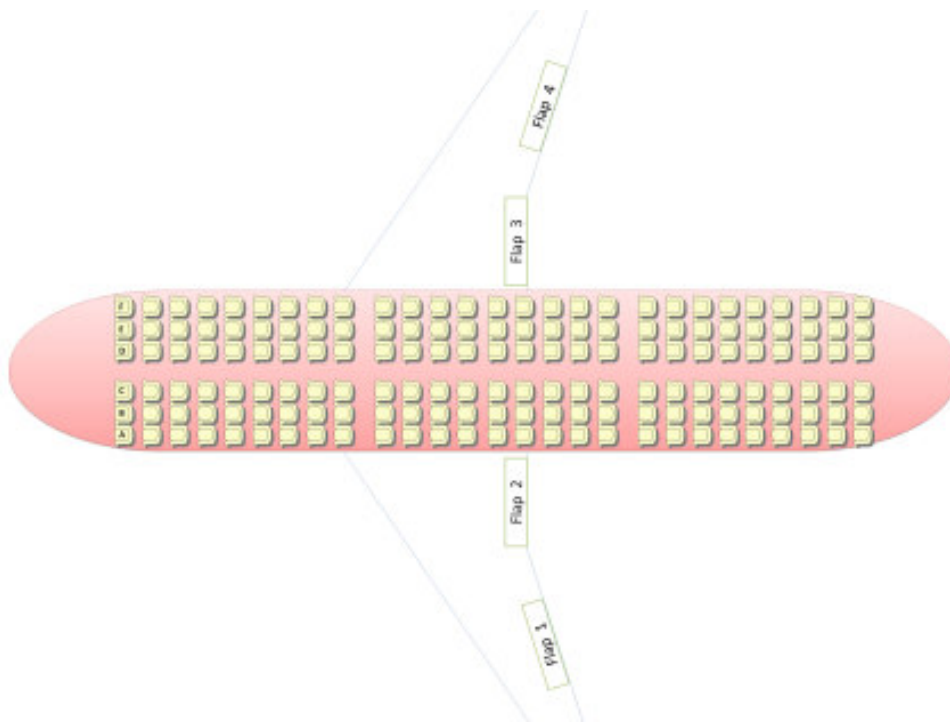
Presentazione Elaborato

E' stato sviluppato un programma Assembly che controlla l'inclinazione dei flap di un Airbus A320 in base al numero e alla distribuzione dei posti a sedere dei passeggeri. Può funzionare in 2 modalità di controllo in base al codice di partenza inserito:

- Controllo emergenza (9 9 2)
- Controllo dinamico (3 3 2)

Nel caso in cui il codice fosse errato (diverso dalle sequenze o incompleto) il programma permette di reinserire il codice dopo aver segnalato l'errore con un massimo di 3 tentativi. Se anche l'ultimo tentativo non va a buon fine si inserisce automaticamente la " Modalità Safe" e termina il programma.

L'aereo può contenere un massimo di 180 passeggeri, disposti su 6 file (A, B, C, D, E, F).



Controllo Dinamico

Se il codice inserito è 3 3 2 il programma stampa a video “Modalità controllo dinamico inserita”.

Verrà poi chiesto il numero totale dei passeggeri a bordo e, tramite ulteriori chiamate da tastiera, sarà richiesto l’inserimento del numero totale di passeggeri per ogni fila. A questo punto il programma controlla la coerenza tra il numero totale inserito e la somma dei passeggeri sulle sei file inserite successivamente. In caso di errore richiede l’inserimento dei sette valori precedenti. Il programma ora calcola le differenze di peso tra parte sinistra e destra del veicolo e imposta di conseguenza il bias per l'inclinazione dei flap nel seguente modo:

$$x = nA - nE$$

$$y = nB - nE$$

$$z = nC - nD$$

$$Bias_{flap_1} = \frac{x}{2} * k1 + \frac{y}{2} * k2$$

$$Bias_{flap_2} = \frac{y}{2} * k2 + \frac{z}{2} * k3$$

$$Bias_{flap_3} = -\frac{y}{2} * k2 - \frac{z}{2} * k3$$

$$Bias_{flap_4} = -\frac{x}{2} * k1 - \frac{y}{2} * k2$$

Dove, le costanti k1, k2 e k3 valgono 3, 6, 12 rispettivamente.

Il programma stampa il valore dei quattro Bias e termina.

Controllo Emergenza

Se il codice inserito è 9 9 2 il programma stampa a video “Modalità controllo emergenza inserita” e termina il programma.

Variabili utilizzate

Ogni variabile di tipo stringa (cioè di tipo Ascii) è accompagnata da un'altra variabile della forma `variabile_len` che rappresenta un numero intero che è la lunghezza della stringa contenuta nella variabile.

File:

progetto.s

- Nessuna

confronta.s

- `nuovo_ins` : messaggio da stampare in caso di codice errato per richiedere un nuovo inserimento (dichiarata di tipo Ascii)
- `dynamic_ins` : messaggio da stampare quando ci sarà l'inserimento della "modalità controllo dinamico" (dichiarata di tipo Ascii)
- `safe_ins` : messaggio da stampare quando ci sarà l'inserimento della "modalità controllo di emergenza" (dichiarata di tipo Ascii)
- `failure_ins` : Messaggio da stampare in caso di superamento dei tentativi per l'inserimento del codice. Verrà comunicato l'inserimento automatico della "modalità di emergenza" e terminerà il programma (dichiarata di tipo Ascii)

atoi.s

tastiera.s

itoa.s

- `car` : puntatore alla stringa da stampare (dichiarata di tipo byte)

dinamico.s

- `tot`: indica il totale dei passeggeri (dichiarata di tipo long)
- `car`: variabile utilizzata per andare a capo (dichiarata di tipo byte)
- `tot_somma`: indica la somma dei passeggeri delle file(dichiarata di tipo long)
- `x`: variabile per i bias (dichiarata di tipo long)
- `y`: variabile per i bias (dichiarata di tipo long)
- `z`: variabile per i bias (dichiarata di tipo long)
- `bias1`: variabile che contiene il valore del bias1 (dichiarata di tipo long)
- `bias2`: variabile che contiene il valore del bias2 (dichiarata di tipo long)
- `bias3`: variabile che contiene il valore del bias3 (dichiarata di tipo long)

- bias4: variabile che contiene il valore del bias4 (dichiarata di tipo long)
- meno: variabile usata per stampare l'operatore "-" (dichiarata di tipo long)
- resto_divisione: variabile utilizzata per stampare il resto della divisione per 2 (dichiarata di tipo long)
- tot_ins: messaggio da stampare per chiedere l'inserimento del totale dei passeggeri (dichiarata di tipo Ascii)
- fila_a_ins: messaggio da stampare per chiedere l'inserimento dei passeggeri della fila A (dichiarata di tipo Ascii)
- fila_b_ins: messaggio da stampare per chiedere l'inserimento dei passeggeri della fila B (dichiarata di tipo Ascii)
- fila_c_ins: messaggio da stampare per chiedere l'inserimento dei passeggeri della fila C (dichiarata di tipo Ascii)
- fila_d_ins: messaggio da stampare per chiedere l'inserimento dei passeggeri della fila D (dichiarata di tipo Ascii)
- fila_e_ins: messaggio da stampare per chiedere l'inserimento dei passeggeri della fila E (dichiarata di tipo Ascii)
- fila_f_ins: messaggio da stampare per chiedere l'inserimento dei passeggeri della fila F (dichiarata di tipo Ascii)
- errore_tot: messaggio di errore da stampare se inserito un numero totale di passeggeri diverso dalla somma totale delle file (dichiarata di tipo Ascii)
- failure_ins: messaggio da stampare in caso di superamento dei tentativi per l'inserimento del codice. Verrà comunicato l'inserimento automatico della "modalità di emergenza" e terminerà il programma (dichiarata di tipo Ascii)
- troppi_pass: messaggio di errore da stampare in caso sia stato inserito un numero totale di passeggeri superiore al limite di 180 (dichiarata di tipo Ascii)
- troppi_fila: messaggio di errore da stampare in caso sia stato inserito un numero di passeggeri nella fila superiore a 30 (dichiarata di tipo Ascii)

Funzioni

Progetto

questa funzione corrisponde al file main:

- eseguo un pop per verificare quanti argomenti sono stati inseriti dalla riga di comando
 - Se non sono 4, si imposta ECX a 1 (indica l'errore) e si invoca la funzione confronta
 - Altrimenti si procede con la lettura dei parametri
- Eseguo 1 pop a vuoto per arrivare a poter estrarre direttamente i parametri che mi interessano
- Legge il valore inserito:
 - Leggo il primo parametro
 - Sposto il primo parametro in ECX
 - Leggo il secondo parametro (Ho il primo parametro in EAX e il secondo in ECX)
 - $EAX = EAX * 10 \rightarrow EAX = EAX + ECX$
 - sposto il valore in ECX
 - Leggo il terzo parametro
 - $EAX = EAX * 10 \rightarrow EAX = EAX + ECX$
 - In EAX ho il codice inserito dall'utente. Non è diviso in 3 registri
- Azzerò ECX che indica se c'era stato un errore nell'inserimento dei parametri.
- Chiamata alla funzione confronta
 - Se la funzione ritorna indica che è stata inserita la modalità di controllo dinamico (altrimenti l'uscita dal programma sarebbe avvenuta nella funzione per il confronto)
- Chiamata alla funzione per il controllo dinamico. Se la funzione ritorna significa che sono stati inseriti correttamente tutti i valori, altrimenti il programma sarebbe terminato all'interno della funzione per la modalità di controllo dinamico
- Uscita dal programma

Confronta

Funzione che si utilizza per confrontare che il codice inserito dall'utente corrisponda a quello della modalità dinamica o emergenza. EDX conta i tentativi di inserimento. In seguito alla dichiarazione delle variabili globali e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- parte il conteggio di EDX
- Verifico attraverso ECX se è già stato commesso un errore nell'inserimento dei parametri da riga di comando
- verifico se il numero inserito corrisponde effettivamente ad un codice delle due modalità:
 - se è stato inserito il controllo emergenza, ne stampa il messaggio e il programma termina
 - se è stato inserito il controllo dinamico faccio partire la funzione dinamico per il controllo passeggeri
 - in caso di errore viene segnalato con un messaggio di errore e aumentato il conteggio di EDX dei tentativi. Se il numero di tentativi è inferiore alla soglia (3) viene richiesto un nuovo inserimento del codice attraverso la funzione atoi. In caso sia stata superata la soglia la funzione termina
- ripristino dei registri salvati sullo stack
- ritorno alla funzione chiamante

Atoi

Funzione utilizzata per trasformare un numero letto da tastiera, dunque Ascii, in un numero intero ricorsivamente. In seguito alla dichiarazione delle variabili globali e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- salva i registri sullo stack
- converte un carattere alla volta dal codice ASCII
- scorre tutta la stringa fino al carattere 0 di fine stringa
- ripristino dei registri salvati sullo stack ma nell'ordine inverso secondo la regola LIFO (Last In First Out) dello stack
- ritorno alla funzione chiamante

Nota: su EAX deve essere già presente il numero da convertire. Il parametro convertito viene restituito in EAX.

Tastiera

Funzione utilizzata per leggere un solo carattere da tastiera. In seguito alla dichiarazione delle variabili globali e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- salva i registri sullo stack
- legge un solo carattere da tastiera salvandolo sulla variabile car
- sposta il valore sul registro eax
- ripristino dei registri salvati sullo stack
- ritorno alla funzione chiamante

Dinamico

Funzione che si utilizza quando viene inserita la modalità dinamica, EDX conta i tentativi di inserimento. In seguito alla dichiarazione delle variabili globali e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- inizia il conteggio di EDX che viene salvato sullo stack
- parte il messaggio di richiesta dell'inserimento del numero totale dei passeggeri
- lettura valore e verifica (max 180). Cambiamento da Ascii ad intero e salvataggio in eax
- per ogni fila :
 - parte il messaggio di richiesta dell'inserimento del numero di passeggeri sulla fila
 - lettura valore e cambiamento da Ascii ad intero
 - verifica di non avere più di 30 passeggeri nella fila
 - somma il valore nella variabile contenente la somma dei passeggeri delle altre file
- verifica che il totale corrisponda con la somma di tutti i passeggeri di ogni fila

- in caso di errore viene segnalato con un messaggio di errore e aumentato il conteggio di EDI dei tentativi. Se il numero di tentativi è inferiore alla soglia (3), si ricomincia la funzione. In caso sia stata superata la soglia la funzione termina.
- Mentre vengono chiesti i valori delle file, vengono calcolate x , y e z
- Vengono calcolati e stampati a video uno alla volta tutti i bias
- ripristino dei registri salvati sullo stack
- ritorno alla funzione chiamante

Itoa

funzione che converte un valore intero in una stringa e stampa un numero decimale alla volta. In seguito alla dichiarazione delle variabili globali e della funzione, nella sezione istruzioni appaiono le seguenti operazioni:

- salva i registri sullo stack
- salva su ECX il numero di cifre da stampare
- attraverso continue divisioni per 10 carica una alla volta le cifre su car
- esegue le operazioni di stampa del valore contenuto su car ad ogni ciclo
- ripristino dei registri salvati sullo stack
- ritorno alla funzione chiamante

Nota: su EAX deve essere già presente il numero da stampare.

Scelte progettuali

Di seguito verranno elencate le nostre scelte progettuali che non vanno a modificare le direttive del progetto ma ci hanno permesso di apportare qualche modifica dal punto di vista dell'interfaccia e dell'efficacia del programma:

- L'inserimento dei valori viene richiesto singolarmente per ogni fila in modo da rendere la richiesta di input più chiara e per evitare un ulteriore controllo riguardante la pressione degli spazi tra un numero di fila e l'altro
- Abbiamo deciso di limitare a 3 tentativi l'inserimento del numero totale dei passeggeri e delle 6 file poiché nelle direttive non era stato precisato alcun limite. A noi non sembrava adeguato lasciare che in caso di continui errori il programma richiedesse ad oltranza un nuovo inserimento
- Nelle direttive viene specificata la capacità dell'Airbus ed abbiamo dunque scelto di introdurre 2 controlli:
 - Il numero totale dei passeggeri non può essere maggiore di 180 altrimenti viene chiesto un nuovo inserimento e viene incrementato il conteggio dei tentativi
 - Il numero totale dei passeggeri per ognuna delle 6 file non può essere maggiore di 30 (dal disegno le 6 file erano tutte uguali).
- Con il progetto e la relazione abbiamo allegato anche il file in linguaggio C che abbiamo creato prima di svolgere il progetto e dal quale abbiamo preso spunto per eseguire il progetto in linguaggio Assembly