



UNIVERSITÀ
di **VERONA**

Dipartimento
di **INFORMATICA**

-ELABORATO ASM- ARCHITETTURA DEGLI ELABORATORI

-SECONDO SEMESTRE 2021-2022-

**Kovacs Alex 'VR471323'
Faedo Marco 'VR472571'
Bonotto Sally 'VR476772'**

-ELABORATO ASM-

INDICE:

Traccia

1

Variabili

2-3

Funzioni

4

Diagramma di flusso

5

Scelte progettuali

6

-ELABORATO ASM-

Descrivere un programma che simuli il sistema di telemetria del videogame F1.

Il sistema fornisce in input i dati grezzi di giri motore (rpm), temperatura motore e velocità di tutti i piloti presenti in gara per ogni istante di tempo.

OBIETTIVO:

Si scriva un programma in assembly che restituisca i dati relativi al solo pilota

indicato nella prima riga del file, in base a delle soglie indicate.

Vengono definite tre soglie per tutti i dati monitorati:
LOW,MEDIUM,HIGH.

Le soglie per i dati monitorati sono così definite:

- Giri Motore

- LOW: rpm ≤ 5000
- MEDIUM: $5000 < \text{rpm} \leq 10000$
- HIGH: rpm > 10000

- Temperatura

- LOW: temp ≤ 90
- MEDIUM: $90 < \text{temp} \leq 110$
- HIGH: temp > 110

- Velocità

- LOW: speed ≤ 100
- MEDIUM: $100 < \text{speed} \leq 250$
- HIGH: speed > 250

Il file di output mi dovrà restituire:

I tempi del giro, la soglia di temperatura, la soglia dei giri motore e la soglia della velocità.

E la struttura dell'ultima riga mi restituisce:

giri del moto massimi, temperatura massima, velocità massima e velocità media

- 1 -

-ELABORATO ASM-

VARIABILI/ETICHETTE:

Nel file "telemetry.s" sono state utilizzate le seguenti variabili/etichette:

- **caricamento_registro:** carico il numero che mi identifica il pilota all'interno del registro ebx e lo includo nello stack.

- **controllo_ID_pilota:** espello dallo stack il pilota, incremento il registro della conta dei piloti e se sia arriva al numero 20 sono arrivato all'ultimo e quindi salto poi dentro alla funzione fail. All'interno di

questa variabile troviamo poi un **ciclo_confronto** in cui sposto il carattere puntato nel file di input nella parte alta del registro e il carattere della stringa del pilota nella parte bassa. Abbiamo poi inserito un **if_terminatori** in cui controllo se i caratteri di terminazione (\n) coincidono, se sono uguali ci sarà un salto che mi permette di uscire e andare nella variabile **end_if_terminatori**, invece se sono diversi ci sarà un jump che mi tornerà nel controllo_ID_pilota e inizia a confrontare con un altro id pilota.

-end_controllo_ID_piloti: ora nel registro eax troviamo l'ID del pilota, in esi ho il puntatore alla prossima riga, e azzerò ecx. Al suo interno inserisco la variabile **controllo_fino_alla_virgola**, infatti comparerò il termine ascii 44 (che sarebbe la virgola) con la parte alta del registro.

-verifica_rmp_alti: confronto il valore 10,000 con il registro che contiene i giri del motore, se quest'ultimi sono maggiori di tale valore, li assegno con il grado alti e salto all'interno della variabile **assegna_rmp_alti** in cui inserisco il valore 'Hig' all'interno del registro edx, copiando poi ciò che sta nel registro nel file di output e infine salto all'interno di **fine_assegnamento_rmp** e vado a confrontare la temperatura.

- 2 -

-ELABORATO ASM-

- verifica_rpm_medi: se i rpm sono minori di 10.000 vado all'interno di questa variabile e li comparo con il valore 5000, se sono maggiori di 5000 vado in **comparazione2**, all'interno di questa variabile, il valore deve essere maggiore di 5000 e minore uguale di 10,000, se si verifica questo eseguo le stesse istruzioni scritte precedentemente.

- verifica_rpm_bassi: se i giri del motore sono inferiori a 5000, entro nella variabile **assegna_rmp_bassi** e poi salto all'interno di **fine_assegnamento_rmp**.

-verifica_temperatura_alta: confronto il valore 110 con il registro che contiene la temperatura, se quest'ultima è maggiore di 110 vado all'interno della variabile **assegna_temperatura_alta**, nella quale assegno 'hig' nel registro edx, copiando poi tutto nel file di output e andare in **fine_assegnamento_temperatura** e vado a confrontare la velocità.

-verifica_temperatura_media: se invece la temperatura è minore di 110 vado all'interno di questa variabile, comparandola con il valore minimo 90, se è maggiore di questo valore e minore di 110 salto

all'interno di **assegna_temperatura_media** ed eseguo le stesse istruzioni.

-verifica_temperatura_bassa: se il valore è minore o uguale di 90 invece, entro in **assegna_temperatura_bassa** e poi salto all'interno di **fine_assegnamento_temperatura**.

-verifica_velocità_alta: confronto il valore 250 con il registro che contiene la velocità, se è maggiore di 250 entro in **assegna_velocità_alta** e copio poi tutto nel file di output.

- 3 -

-ELABORATO ASM-

-verifica_velocità_media: se invece il valore è inferiore o uguale a 250 vado all'interno di questa variabile e lo comparo con il valore 100, perché per assegnare il valore medio deve essere compresa tra 100 e 250 compreso. Se si verifica questo, entro in **assegna_velocità_media**.

-verifica_velocità_bassa: se confrontando il valore è minore di 100 salto in quest'ultima variabile in cui assegno il valore di velocità bassa.

FUNZIONI:

-fail: questa funzione mi restituisce 'Invalid' se la stringa del pilota non corrisponde con il file di input.

- copy_funzione: carico nello stack i registri ebp,ecx,eax, poi muovo il valore di esp (Il registro ESP è un registro specializzato di tipo puntatore che contiene l'indirizzo della cima della memoria) in ebp e azzerò ecx in quanto lo devo usare come indice derefenziato. Nel **copy_loop:** estraggo il carattere, comparo il carattere 0 (null) con la parte alta del registro e quando arrivo al terminatore capisco che sono arrivato alla fine, incremento poi il registro edi, copierò il carattere 44 (che sarebbe la virgola) e se sono arrivato qui salto all'interno del **return_copy**, altrimenti torno ancora nel **copy_loop** perché significa che non ho trovato nè il terminatore nè la virgola. Nel **return_copy** tutto quello che sta nello stack lo introduco nei registri.

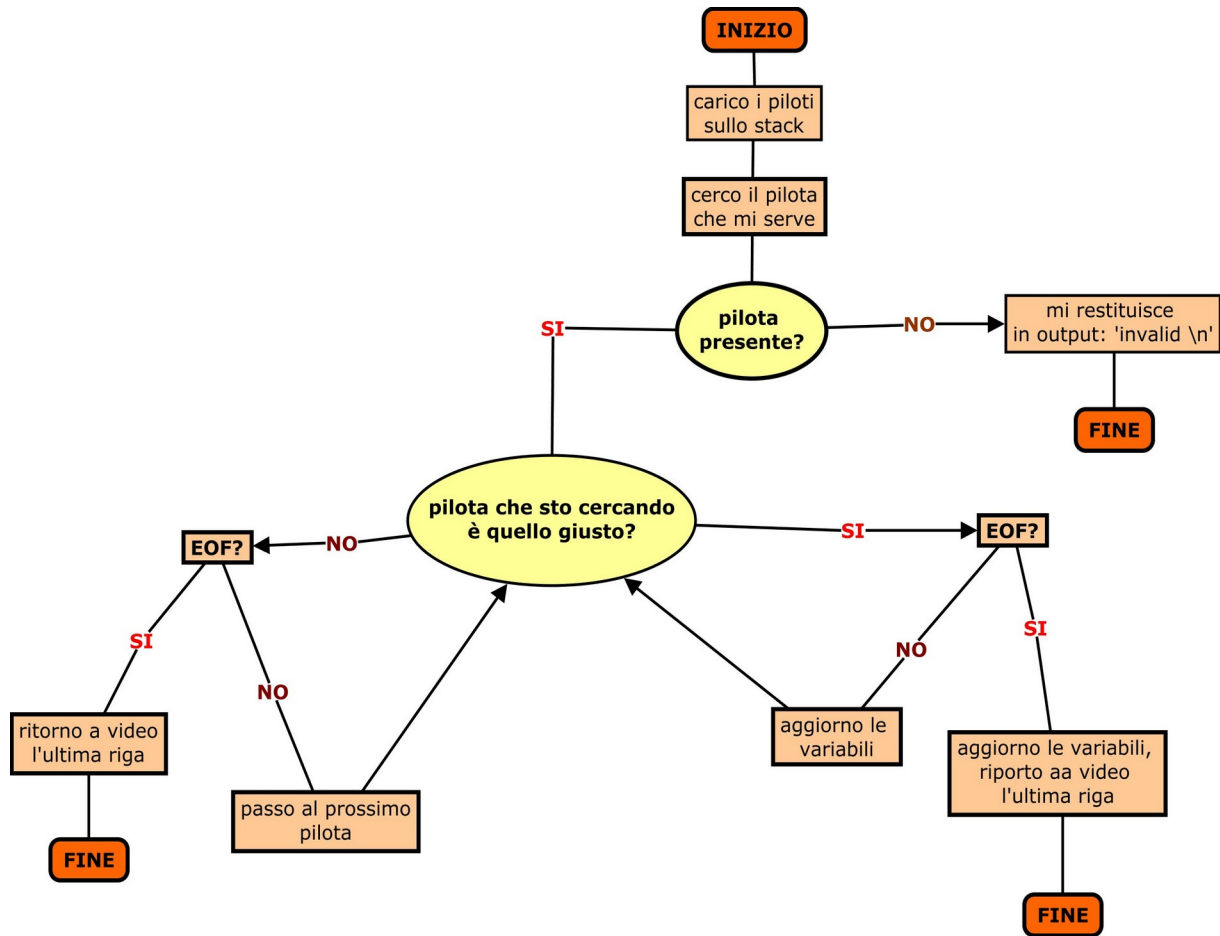
-atoi: mi permette di convertire una stringa ad un intero, carico nello stack i registri ebp ed eax, eseguo le operazioni, vengono ripristinati i

registri salvati sullo stack; e ritorno alla funzione chiamante in cui mi salvo l'id del pilota in ebx per poterlo confrontare, ripristino poi eax e il base pointer.

- 4 -

-ELABORATO ASM-

DIAGRAMMA DI FLUSSO:



- 5 -

-ELABORATO ASM-

SCELTE PROGETTUALI:

Per trovare i valori massimi e la velocità media ci siamo appoggiati a 5 variabili:

tmp_max, temperatura_max, velocità_max, somma_velocità e numero_righe.

Ad ogni confronto con le variabili inizializzate a zero, il loro valore andava a modificarsi se le condizioni incontrate erano vere; per la variabile della somma invece non c'è stato nessun confronto in quanto siamo andati a sommare tutte le velocità trovate in quanto il confronto del pilota avveniva prima di arrivare a questo punto, per il numero di righe è stato eseguito lo stesso procedimento. Queste ultime due variabili sono state utilizzate come appoggio per trovare la velocità media così da poter costruire infine l'ultima riga.