

Progetto Architettura 2

Fold Left

Gabriele Padovani

29 agosto 2020

1 Introduzione

Ho deciso di realizzare il progetto sull'unità didattica 5, su procedure ricorsive. In particolare ho pensato di realizzare una procedura chiamata Fold left, spesso utilizzata per processare liste nei linguaggi di alto livello.

2 Descrizione

Fold left é una procedura utilizzata in molti linguaggi funzionali che permette di eseguire un'operazione binaria su un array, in modo tale che la prima iterazione sia eseguita sui primi due elementi, la seconda iterazione sul risultato della prima e il terzo elemento, la terza sul risultato della seconda e il quarto elemento, e cosí via.

La procedura é detta Fold, perché l'array viene piegato su se stesso, eseguendo piú volte la stessa operazione. Left invece, perché viene eseguita da sinistra verso destra. Esiste anche una versione che é eseguita nel senso opposto, detta Fold Right.

L'operazione binaria eseguita puó essere, per esempio, una somma che permette di ottenere la somma dell'intero array, o anche una funzione che prende il massimo tra due valori, in modo tale da ottenere il massimo globale dell'array.

Tradotta in un'equazione ricorsiva, nel caso di una somma, questo procedimento diventa:

$$\begin{cases} r_1 = a_0 + a_1, & \text{in caso di indici uguali a 0 e 1} \\ r_x = r_{x-1} + a_x, & \text{in ogni altro caso} \end{cases} \quad (1)$$

Per il massimo invece:

$$\begin{cases} r_1 = \max(a_0, a_1), & \text{nel caso di indici uguali a 0 e 1} \\ r_x = \max(r_{x-1}, a_x), & \text{in ogni altro caso} \end{cases} \quad (2)$$

3 Implementazione

3.1 Main

In *main.asm*

In un linguaggio di alto livello normalmente é possibile passare come argomento una funzione, il che rende Fold Left molto comoda. In assembly questo non é possibile, quindi ho deciso di

utilizzare una variabile per scegliere quale operazione realizzare, 0 per addizione e 1 per massimo (nel caso si volesse aggiungere una nuova operazione é possibile usare gli interi successivi).

Al termine dell'esecuzione della procedura, stampo una stringa per indicare quale operazione é stata eseguita, e il risultato della Fold.

3.2 Procedura Fold Left

In *fold_left.asm*

Per quanto riguarda le procedure, la convenzione indica che i registri $\$ra, \$s_0, \$s_1 \dots$ sono **callee saved** (devono essere salvati dalla procedura chiamata), mentre gli altri sono registri **caller saved**, che sono salvati invece dalla procedura chiamante.

Nel mio caso, salveró $\$ra$, perché é **callee saved**, e i registri $\$ax$, perché sono **caller saved** (quando sto per chiamare max o addition).

3.3 Procedure Addition, Max e Mult

In *addition.asm*, *max.asm* e *mult.asm*

Per le procedure addition e max, dovró salvare $\$ra$ perché é **callee saved**, anche se queste ultime non chiamano nessuna procedura, e il salvataggio dello stato ha effetto nullo.