

Progetto MIPS/SPIM

Descrizione

Questa tipologia di progetto consiste in

- Sviluppare un programma in linguaggio MIPS assembly
- Simulare il programma in QtSPIM
- Preparare una relazione del progetto (max 20 pagine)

Inoltre,

- Si può presentare il progetto individualmente o in coppie
- Al momento dell'esame, occorre presentare una copia cartacea della relazione e una copia digitale (CD o DVD, comprensiva di codice)

Infine,

- Lo studente potrà scegliere tra i progetti presenti in questo documento, oppure proporre progetti differenti. Nel secondo caso, il Professore valuterà la coerenza e la fattibilità della proposta.
- In ogni caso, il progetto scelto va comunicato al Prof. Leonardo Pasini via mail (leonardo.pasini@unicam.it), mettendo in cc anche il sottoscritto (nicola.paoletti@unicam.it)

Nella pagina seguente, sono elencati alcuni progetti disponibili e un template esemplificativo per la relazione.

Lista dei progetti disponibili

Progetto 1 – Reverse string

Implementare una procedura in MIPS che presa in input una stringa, stampa la stringa inversa, ovvero con ordine dei caratteri invertiti. Es. Input: **pip**po, Output: **opp**ip. Input: **anna**, Output: **anna**.

Progetto 2 – Calculator

Implementare una procedura in MIPS che simula una calcolatrice. Ovvero prende in input una stringa o un codice che identifica l'operazione da eseguire (addizione, sottrazione, moltiplicazione, divisione, elevamento a potenza), poi prende in input gli operandi (interi), e stampa il risultato.

La procedura termina solo quando l'utente immette un apposito codice/stringa.

Progetto 3 – Analisi di procedure MIPS

Implementare una procedura in MIPS per confrontare le performance delle due implementazioni (iterativa e ricorsiva) dell'elevamento a potenza (vedere slides e codice).

Il tempo di esecuzione (somma pesata del numero dei passi) è calcolato come segue (vedi App. A di [http://cad6.csie.fju.edu.tw/comorg97/download/MIPS_Assembly_Language_Programming_\(2003\).pdf](http://cad6.csie.fju.edu.tw/comorg97/download/MIPS_Assembly_Language_Programming_(2003).pdf)):

- **mul** ha peso **33**.
- Istruzioni di salto condizionato quali **bge**, **bgt**, **ble**, **blt** hanno peso **2**.
- **li** e **la** hanno peso **2**.
- **Tutto il resto (lw, sw, syscall, ...)** ha peso **1**.

In input: base e esponente. In output si restituisce il tempo di esecuzione nel caso ricorsivo e nel caso iterativo.

La procedura va testata con diversi input, in particolare va tenuta fissa la base e vanno confrontati i tempi con esponenti crescenti. I risultati vanno discussi nella relazione.

Progetto 4 – String replace

Implementare una procedura in MIPS che prende in input una stringa **s**, un carattere **c1** e un carattere **c2**. Restituisce (stampa) la stringa **s** in cui ogni occorrenza di **c1** è sostituita da **c2**. Es. Input: **pip**po, **p**, **c**.

Output: **cic**co.

Progetto 5 – Funzione esponenziale

Implementare una procedura in MIPS che effettua il **calcolo della funzione esponenziale**. Il programma prende in input un numerico $n \geq 0$ e restituisce l'esponenziale di n , calcolato attraverso lo sviluppo di Taylor, per cui

$$e^n = \sum_{i=0}^{max} \frac{n^i}{i!}$$

Max è il numero massimo di iterazioni da eseguire.

Suggerimenti:

- Avrete bisogno di utilizzare i registri floating-point
- Non è necessario ogni volta calcolare x^i e $i!$. Infatti ad ogni iterazione, per il primo basta moltiplicare ad ogni passo x^{i-1} per x , e per il secondo basta moltiplicare $(i-1)!$ per i .

Progetto 6 – Print memory

Implementare una procedura in MIPS che stampa il contenuto della memoria in un dato intervallo. In input prende due indirizzi di memoria (due numerici) **a1** e **a2**. Supposto che **a1** <= **a2**, per ogni word di memoria a partire da **a1** fino a **a2**, stampa una riga della forma:

word [a1;a1+3]: *contenuto*
word [a1+4;a1+7]: *contenuto*

...

La procedura si ferma non appena l'indirizzo corrente è >= **a2**.

Progetto 7 – String concatenation

Implementare una procedura in MIPS che date in input due stringhe **s1** e **s2**, restituisce la stringa concatenata **s1s2**.

Progetto 8 – Substring

Implementare una procedura in MIPS che dati in input una stringa **s1**, un numerico **beginIdx** e un numerico **endIdx**, restituisce la sottostringa di **s1** compresa tra gli indici **beginIdx** e **endIdx**.

Es. Input: pippo, 1, 3.

Output: ip

Vanno effettuati controlli sugli indici, ovvero

- **beginIdx** < **endIdx**
- **0** <= **beginIdx** < length(**s1**)
- **0** < **endIdx** <= length(**s1**)

Progetto 9 – Numbers to letters

Implementare una procedura in MIPS che dato in input un intero, restituisce il valore in lettere (separato da spazi).

Es. Input: 321. Output: tre due uno.

Progetto 10 – Sezione aurea

Implementare una procedura in MIPS che effettua il **calcolo approssimato della sezione aurea ϕ** .

$$\phi = \frac{1 + \sqrt{5}}{2}$$

ϕ è calcolata a partire dai numeri di Fibonacci nel seguente modo:

$$\phi = \lim_{n \rightarrow +\infty} \frac{F_n}{F_{n-1}}$$

in cui F_n è l'n-esimo numero di Fibonacci. I numeri di Fibonacci sono dati dalla successione:

$$F_n = F_{n-1} + F_{n-2}$$

con

$$F_0 = 0 \text{ e } F_1 = 1$$

La procedura prende in input un numero massimo di passi (**max**) e restituisce ϕ come

$$\phi = \frac{F_{max}}{F_{max-1}}$$

NOME PROGETTO

Progetto di Architettura degli Elaboratori – Prof. Leonardo Pasini

Corso di Laurea in Informatica – Scuola di Scienze e Tecnologie – Università di Camerino

Anno Accademico: 2011/2012

Candidati:

Candidato 1 – Matricola n. 1

Candidato 2 – Matricola n. 2

...

Specifica del problema

Descrivere il problema da risolvere. La lunghezza di questa parte dipende naturalmente dal progetto scelto.

Descrizione dell'algoritmo

Descrivere in modo dettagliato e più formale l'algoritmo da implementare. In questa parte, si consiglia di inserire lo pseudocodice dell'algoritmo e la sua codifica in un linguaggio ad alto livello a scelta (C, Java, ...). Si spieghino e si giustifichino in modo dettagliato le scelte adottate.

Implementazione

In questa parte va aggiunto il codice MIPS assembly implementato e commentato in modo chiaro ed esauriente. Sono consigliati commenti all'interno del codice, e commenti più ampi nella relazione.

Simulazione

In questa sezione andranno gli screenshot commentati di una simulazione-tipo.