

Calcolatori Elettronici

Prova di Laboratorio - assembly MIPS: media_array_1D

12 Settembre 2012

Introduzione

Lo scopo di questa prova di laboratorio è lo sviluppo di un semplice programma nel linguaggio assembly del processore MIPS. Non è richiesta una particolare base di conoscenze algoritmiche, ma semplicemente un minimo di dimestichezza con la programmazione assembly.

Istruzioni

Cominciate facendo il login sulla macchina del laboratorio che vi è stata assegnata. Per il login occorre usare matricola e password dello *student portal*. Sul desktop troverete una cartella contenente i simulatori QtSpim e Mars. Lanciate ed utilizzate quello che preferite. Tutto il vostro codice (sia esso costituito da un singolo file, o da file multipli) andrà salvato nella cartella "mips" da creare sul drive H: .

Create un file `student-info.txt` con incluso il vostro nome e cognome e numero di matricola nella cartella "mips". Per maggior sicurezza, includete anche nome, cognome e matricola come commento, in testa ad ogni file sorgente. Alla fine della prova, i file saranno prelevati automaticamente dalla directory. Tutto quello che lascerete nella cartella mips sarà utilizzato per la valutazione. Salvare i vostri file altrove, o non indicare nome e cognome, porterà inevitabilmente all'annullamento della vostra prova. *Tutti i file all'esterno della cartella verranno cancellati automaticamente!!!*

Informazioni generali

La prova **non** è a correzione automatica. Tutti gli studenti autori di un codice che viene assemblato senza errori e risponde alle specifiche indicate alla sezione seguente saranno ammessi all'orale. In quella sede, il codice prodotto sarà esaminato e discusso col docente.

Le specifiche

Dovete scrivere un programma assembly che inizializza un array monodimensionale di interi con segno, e successivamente calcola in virgola mobile, singola precisione, la media aritmetica dei valori e la stampa in output.

Il main alloca in area stack un array A di 16 interi. Legge da tastiera due numeri m ed n. Poi richiama una funzione array_init, fornendole come parametri il puntatore all'array, m ed n. La funzione inizializza ogni elemento A[i] dell'array al valore $m*3 + n*5 + i$. Al ritorno dalla funzione, il main richiama la funzione media_array, che ha come unico parametro l'indirizzo base dell'array, e ritorna un float che rappresenta la media dei valori dell'array. Alla fine, tale valore viene stampato

Per maggior chiarezza, il seguente è un programma C dalle funzionalità analoghe al codice richiesto

```
void array_init(int *arr, int m, int n) {
    int i;

    for (i=0; i<16; i++)
        arr[i]=m*3+n*5+i;
    return;
}

float media_array(int *arr) {
    int i;
    int somma;
    float media;

    for (somma=0, i=0; i<16; i++)
        somma+=arr[i];

    media=(float)somma/(float)16
    return(media);
}

void main() {
    int A[16];
    int m,n;
    int i,element;

    printf("Dammi m: ");
    scanf("%d",&m);
    printf("Dammi n: ");
    scanf("%d",&n);
    printf("Inizializzo l'array...");
```



```
array_init(A,m,n);
printf(" fatto!\n\n");
```

```
printf("\nValore media =%f\n",media_array(A));
}
```

Facoltativo: chi è capace può provare a scrivere un programma che non faccia uso di codici di moltiplicazione (mult, multu, mul, ...)

Suggerimenti

potete leggere i interi in input da tastiera utilizzando la system call 5. Per la stampa di stringhe, di interi e di float, potete usare le syscall 4, 1 e 2, rispettivamente.

Il cast di un intero a float si può effettuare caricando l'intero in un registro floating-point e poi utilizzando le opportune istruzioni di conversione. Nella sequenza sottostante l'intero 32 viene caricato in \$t0, spostato in \$f2 e convertito in float in \$f4.

```
li $t0, 32          # $t0 = 32
mtc1 $t0, $f2       # $f2 = 32
cvt.s.w $f4, $f2    # $f4 = 32.0
```

Il seguente è un output di esempio:

```
Dammi m: 4
Dammi n: 7
Inizializzo l'array... fatto!

Valore media = 54.50000

-- program is finished running --
```

Valutazione

Scrivere un programma funzionante, che faccia uso di due funzioni come richiesto nelle specifiche e che segua le convenzioni di salvataggio dei registri è strettamente necessario per essere ammessi a sostenere l'orale. In quella sede, si entrerà nel dettaglio della struttura del codice, dando una valutazione migliore a soluzioni "pulite" e ben commentate.