

ETI-Praktikum: Assembler – Gruppe 7

Team:

Philip Daubmeier

Philip Lorenz

Ferdinand Mayet

Aufgabenstellung

- Berechnung der euklidischen Norm eines Vektors
- Signatur der Funktion:

$N \in \mathbb{N}_0, X \in \mathbb{R}^N, INCX \in \mathbb{N}$

DOUBLE PRECISION FUNCTION DNRM2(N, X, INCX)

INTEGER INCX, N

*DOUBLE PRECISION * X*

Euklidische Norm

- Die euklidische Norm berechnet die Länge eines Vektors
- Sie ist durch folgende Formel definiert:

$$\|x\| = \sqrt{(x_1^2 + \dots + x_n^2)} = \sqrt{\sum_{i=1}^n x_i^2} \quad x \in \mathbb{R}^n$$

Pseudocode

- Implementierung in Pseudocode:
pointer = x
for (i = n; i > 0; i--)
 // Wert des Zeigers quadrieren dann
 aufaddieren
 summe += (*pointer) * (*pointer)
 // Adresse des Zeigers verändern
 pointer += incx
// Wurzel ziehen
return sqrt(summe)

Vorgehensweise

- Definieren von Fehlerzuständen:
 - Ungültige Eingaben (z.B. $INCX \leq 0$, $X = \text{null}$, $N < 0$)
 - Überlauf während der Berechnung
- Implementierung des Pseudocodes mit Hilfe des Floating Point Unit (FPU)
- Dabei wurden folgende FPU-Befehle benutzt: FLDZ, FLD, FMUL, FADDP, FSQRT

Der FPU-Stack

- Der FPU-Stack besteht aus 8 Speicherzellen (ST0 - ST7) welche 80-bit Gleitkommazahlen enthalten
- Wird ein Element eingefügt (PUSH) landet es an der Position ST0 und alle anderen Elemente werden nach oben verschoben
- Entnimmt man ein Element (POP) wird ST0 vom Stack genommen und alle weiteren Elemente werden nach unten verschoben



FPU-Befehle im Detail

- FLDZ:
Lädt den Wert 0 in ST0
- FLD size source:
Lädt den Wert der Größe size aus der Quelle source in ST0
Beispiel: FLD QWORD [Speicheradresse]
- FMUL STx:
Multipliziert ST0 mit STx und speichert das Ergebnis in ST0

FPU-Befehle im Detail (Fortsetzung)

- **FADDP STx, ST0:**
Addiert ST0 und STx, speichert das Ergebnis in STx und poppt dann den FPU-Stack
- **FSQRT**
Berchnet die Wurzel von ST0 und speichert das Ergebnis in ST0

Testumgebung

- Programmaufruf ohne zusätzliche Parameter überprüft die Fehlerfälle und verifiziert die Korrektheit der Ergebnisse mit Zufallsvektoren
- Programmaufruf mit mehreren Parametern berechnet die euklidische Norm des angegebenen Vektors (eine Komponente pro Argument)

Zusammenfassung

- Signatur: *double DNRM2(int n, double *x, int incx)*
- Rückgabewerte im Fehlerfall:
 - *-1* falls die Eingabeparameter ungültig waren
 - *Infinity* falls das Ergebnis den maximal darstellbaren Wert überschreitet