

Benutzerdokumentation

zur maschinennahen Programmierung

TGI-Praktikum SS03

Gruppe 29

Manuel Mayr
Schulstraße 4
85356 Attaching
mayrm@in.tum.de

Alexander Gafriller
Schulstraße 4
85356 Attaching
gafrille@in.tum.de

29. Juni 2003

Inhaltsverzeichnis

1 Funktionsweise des Befehls	2
1.1 Funktionalität	2
1.2 Voraussetzung	2
1.2.1 Architektur	2
1.2.2 Betriebssystem	2
1.3 Signatur	3
1.3.1 Black-Box	3
1.4 Eingabebereich	3
2 Benutzung des Befehls	5
2.1 Bibliothek	5
2.2 Verwendung des Befehls	5
2.3 Beispielcode	5
2.4 Compilieren	6

Kapitel 1

Funktionsweise des Befehls

1.1 Funktionalität

Der Befehl `fast_sin` errechnet den Sinus einer 8-Byte langen Fließkommazahl gemäß der mathematischen Definition $y = \sin(x)$ mit $x \in \mathfrak{R}$. Der in der Programmiersprache C dazu äquivalente Datentyp ist `double`, wobei die Eingabewerte x hier natürlich aus dem Definitionsbereich kommen muss den der Datentyp `double` aufspannt. Die Eingabe muss für eine korrekte Funktionsweise im Bogenmaß übergeben werden.

1.2 Voraussetzung

1.2.1 Architektur

Der Befehl wurde für einen zum Intel 80386 kompatiblen PC programmiert. Für die korrekte Anwendung und Funktionsweise ist das Vorhandensein einer zur 80387 Fließkommazahleinheit (FPU) von Intel oder eines entsprechenden Emulators unbedingt vonnöten.

1.2.2 Betriebssystem

Der Befehl wurde sowohl auf dem Betriebssystem `WindowsXP` implementiert als auch getestet. Bei der Erstellung wurde allerdings darauf geachtet keine proprietäre Software zu verwenden, sondern nur jene die offenen Standards genügt. Dadurch dürfte der Portierungsaufwand auf eine andere Plattform, beispielsweise ein `Unix`-System, sehr gering sein.

1.3 Signatur

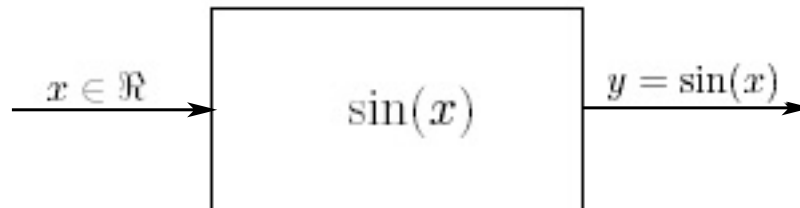
Der Befehl `fast_sin` hat folgende Signatur:

```
double fast_sin(double*);
```

Der Eingabewert den die Funktion fordert ist ein Zeiger auf einen 8-Byte-`double`-Wert, die Gründe hierfür liegen in der Implementation. Die Funktion liefert wiederum einen `double`-Wert als Ergebnis, der sich definitionsgemäß im Intervall $[-1, 1]$ aufhalten wird.

1.3.1 Black-Box

Diese Sichtweise spiegelt die sichtweise des Benutzers wieder, der zwar die Signatur kennt, aber nicht weiß wie die Funktion realisiert wurde.



1.4 Eingabebereich

Wie bereits in der Beschreibung der Signatur erwähnt, ist der Eingabewert ein Zeiger auf einen `double`-Wert. Der Wertebereich den dieser Datentyp aufspannt erstreckt sich im Intervall von $\pm 1.7 \cdot 10^{-308} \dots \pm 1.7 \cdot 10^{308}$ und bietet eine Genauigkeit von 15 Stellen.

Die Sinusfunktion als mathematisches Symbol hat keinerlei Definitionslücken, daher sind alle möglichen Werte (die sich in einer 8-Byte Fließkommazahl speichern lassen) als Parameter zugelassen. Es sei nur darauf hingewiesen, dass die Funktion das Argument intern auf das Intervall von $[0, 2 \cdot \pi]$ zurückführt.

Dabei können bei sehr grossen Zahlen signifikante Rundungsfehler auftreten. Zudem wird bei der Berechnung des Sinus angenommen, dass sich der Parameter im Bogenmaß befindet.

Kapitel 2

Benutzung des Befehls

2.1 Bibliothek

Der Befehl `fast_sin` wird in zwei Dateien zur Verfügung gestellt:

- (i) `sinus.h` enthält die Deklaration, und muss in den C-Quellcode eingebunden werden (siehe **2.2 Verwendung des Befehls**).
- (ii) `sinus.o` enthält als Objektdatei die gesamte Definition, sprich Implementation des Befehls.

2.2 Verwendung des Befehls

Um den Befehl verwenden zu können muss im C-Quellcode die Headerdatei `sinus.h` mit dem Befehl

```
#include "sinus.h"
```

eingebunden werden. Die Headerdatei `sinus.h` speichert man am besten zusammen mit der zugehörigen Objektdatei `sinus.o` im Verzeichnis der Applikation. Im Folgenden wird von einer solchen Konstellation ausgegangen und eine Beispieldatei kompiliert.

2.3 Beispielcode

Um den Beispielcode zu kompilieren wurde ein `gcc`-Compiler in der Version 3.2 verwendet.

```
/* **** */
/* Name der Datei: Beispiel.c */
/* Beispieldatei um die Funktionsweise der Sinus */
/* Funktion zu erlaeuern. */
/* **** */

#include <stdio.h>
#include <stdlib.h>

#include "sinus.h" /* Einbinden der Headerdatei
                   (liegt im selben Verzeichnis) */

/* Im Folgenden wird der Sinus von 6 berechnet
   und auf den Bildschirm ausgegeben */
int main(void)
{

    // Speicher fuer double-Wert allozieren
    double *d = (double*)malloc(sizeof(double));
    // Wert initialisieren
    *d = 6e0;
    printf("sin(%f) = %f\n", *d, fast_sin(d));

    return 0;
}
```

2.4 Compilieren

Zum Compilieren verwenden wir das obere Codestück. Wir wechseln zuerst in die Eingabeaufforderung und dann in das Verzeichnis, in dem die Codestücke gespeichert werden. Mit den Befehlen

```
gcc -c -o Beispiel.o Beispiel.c
gcc -o sinus.exe Beispiel.o sinus.o
```

wird eine ausführbare Datei erzeugt, die Sie mit

```
sinus.exe
```

aurufen können.