



Inst. für Angew. Informationsverarbeitung

Prof. Dr. Franz Schweiggert
Michaela Weiss
Wolfgang Kaifler

18.01.2011
Blatt 11

Systemnahe Software I (WS 2010/2011)

Abgabetermin: 25.01.2011

Organisatorisches:

- Nächste Woche müssen Vorlesung und Übung leider getauscht werden, d.h am Montag, 24.01. ist Vorstellung des nächsten Übungsblattes und am **Dienstag, 25.01 Vorlesung**
- **Melden Sie sich zur Vorleistung im Hochschulportal an**

1 Fragen (4 + 2 + 20 Punkte)

- Erklären Sie die Bedeutung von *extern* und *static*!
- Nennen Sie einige Vor- bzw. Nachteile sowohl von dynamischen als auch statischen Bibliotheken!

2 Code

Folgender Programmcode sei gegeben:

main.c

```
#include <stdio.h>
#include "module.h"

int main() {
    // Initialisierung
    init();
    // Ausgabe
    printf("In main() ist i: %i\n", i);
    return 0;
}
```

module.h

```
#ifndef MODULE_H
#define MODULE_H
    extern void init();
#endif
```

module.c

```
#include <stdio.h>
#include "module.h"

// globale Variable
int i;
// globale Variable initialisieren
void init() {
    i = 5;
    printf(" in init() ist i = %d\n", i);
}
```

Beim Compilieren tritt folgender Fehler auf:

```
kaiflerw@theseus ~/soft/uebungen/Blatt11 \ $ gcc -Wall main.c module.c
main.c: In function 'main':
main.c:8: error: 'i' undeclared (first use in this function)
main.c:8: error: (Each undeclared identifier is reported only once
main.c:8: error: for each function it appears in.)
[1]kaiflerw@theseus ~/soft/uebungen/Blatt11 \ $
```

Klar, die Variable i ist ja in `module.c` definiert ;-)

Lässt sich dennoch ein Zugriff auf i in `main.c` erreichen? Wenn ja, was müsste geändert werden. (`module.c` soll nicht verändert werden)?

Wenn nein, Begründung!

3 Taschenrechner

In dieser Aufgabe ist ein Taschenrechner zu programmieren, der eine beliebige Funktion entgegennimmt und den Funktionswert einer einzugebenden Stelle berechnet.

Beispiel:

```
Bitte Funktion eingeben (z.B. 2*x + 2 * cos (x)):
f(x) =
2*x + 2 * cos (x)
Bitte das x von f(x) eingeben:
0.5
```

Ergebnis:

```
f(x) = 2*x + 2 * cos (x) ; mit x = 0.5  
f(0.5) ==> 2.000000
```

Gehen Sie wie folgt vor:

- Erstellen Sie eine C-Funktion `void writeModul(char* x)`, die eine mathematische Funktion $F(x)$ übergeben bekommt und daraus ein Programm-Modul `myFunction.c` mit einer C-Funktion `float f(float x)` generiert.
- Erstellen Sie das Hauptprogramm, das die mathematische Funktion $F(x)$ als Zeile entgegennimmt, an `writeModul()` weiterreicht.
- Das Modul soll zur Laufzeit via `system()` zu einer dynamischen Bibliothek übersetzt und eingebunden werden,
- und das Ergebnis auf der Standardausgabe ausgegeben werden.
- Ein Makefile, welches mittels des Targets `clean` alle generierten Dateien wieder entfernt ist natürlich obligatorisch.

Hinweise:

- Eine Funktion `readline()` mit der eine ganze Zeile von der Standarteingabe gelesen werden kann wird zum download zur Verfügung gestellt
- Verwenden Sie für sin- bzw. cos-Berechnungen etc. die Mathe-Bibliothek

Viel Erfolg!