

# Angewandte Informatik

Prof. Dr. Peter Gerwinski

29. November 2012

## 3 Bibliotheken

### 3.1 Der Präprozessor

**#include**: Text einbinden

- **#include** <stdio.h>: Standard-Verzeichnisse – Standard-Header
- **#include** "answer.h": auch aktuelles Verzeichnis – eigene Header

**#define** VIER 4: Text ersetzen lassen – Konstante definieren

- Kein Semikolon!
- Berechnungen in Klammern setzen:  
**#define** VIER (2 + 2)
- Konvention: Großbuchstaben

## 3 Bibliotheken

### 3.2 Bibliotheken einbinden

Inhalt der Header-Datei: externe Deklarationen

**extern int** answer (**void**);

**extern int** printf (**\_\_const char** \***\_\_restrict** **\_\_format**, ...);

Funktion wird „anderswo“ definiert

- separaten C-Quelltext mit an `gcc` übergeben: `foo.c`
- vorcompilierte Objekt-Datei: `foo.o`  
Erzeugen mit: `gcc -c foo.c`
- vorcompilierte Bibliothek: `-lfoobar`  
Erzeugen mit: `ar cr libfoobar.a foo.o bar.o`
- Verzeichnis für vorcompilierte Bibliotheken: `-L ../lib`  
(`-L .` für aktuelles Verzeichnis)

### 3.3 Bibliothek verwenden (Beispiel: OpenGL)

- Include-Dateien:

```
#include <GL/gl.h>
```

```
#include <GL/glu.h>
```

```
#include <GL/glut.h>
```

- Compiler-Aufruf:

```
gcc -Wall -O cube.c -lGL -lGLU -lglut -o cube
```

### 3.3 Bibliothek verwenden (Beispiel: OpenGL)

Selbst geschriebene Funktion übergeben: *Callback*

```
void draw (void)
```

```
{
```

```
    ...
```

```
}
```

```
...
```

```
glutDisplayFunc (draw);
```

→ OpenGL ruft immer dann, wenn es etwas zu zeichnen gibt, die Funktion **draw** auf.

### 3.3 Bibliothek verwenden (Beispiel: OpenGL)

Selbst geschriebene Funktion übergeben: *Callback*

```
void key_handler (unsigned char key, int x, int y)
```

```
{
```

```
    ...
```

```
}
```

```
...
```

```
glutKeyboardFunc (key_handler);
```

→ OpenGL ruft immer dann, wenn eine Taste gedrückt wurde, die Funktion `key_handler` auf.

### 3.3 Bibliothek verwenden (Beispiel: OpenGL)

Selbst geschriebene Funktion übergeben: *Callback*

```
void key_handler (unsigned char key, int x, int y)
```

```
{
```

```
    ...
```

```
}
```

```
...
```

gedrückte Taste

Mausposition

```
glutKeyboardFunc (key_handler);
```

→ OpenGL ruft immer dann, wenn eine Taste gedrückt wurde, die Funktion `key_handler` auf.

## 3.4 Projekt organisieren: make

- Regeln
- Makros



## 3.4 Projekt organisieren: make

- Regeln

```
philosophy: philosophy.o answer.o  
gcc philosophy.o answer.o -o philosophy
```

```
answer.o: answer.c  
gcc -c answer.c -o answer.o
```

```
philosophy.o: philosophy.c answer.h  
gcc -c philosophy.c -o philosophy.o
```

- Makros

## 3.4 Projekt organisieren: make

- Regeln
- Makros

PHILOSOPHY\_SOURCES = philosophy.c answer.h

PHILOSOPHY\_OBJECTS = philosophy.o answer.o

ANSWER\_SOURCES = answer.c

philosophy: \$(PHILOSOPHY\_OBJECTS)

gcc \$(PHILOSOPHY\_OBJECTS) -o philosophy

answer.o: \$(ANSWER\_SOURCES)

gcc -c answer.c -o answer.o

philosophy.o: \$(PHILOSOPHY\_SOURCES)

gcc -c philosophy.c -o philosophy.o

clean:

rm -f \$(PHILOSOPHY\_OBJECTS) philosophy

## 3.4 Projekt organisieren: make

- Regeln
- Makros

→ 3 Sprachen: C, Präprozessor, make