Hochschule Bochum
Bochum University
of Applied Sciences

Angewandte Informatik

Prof. Dr. Peter Gerwinski

6. Dezember 2012

3 Bibliotheken

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);
```

$\longrightarrow$ 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);
```

$\longrightarrow$ OpenGL ruft immer dann, wenn eine Taste gedrückt wurde, die Funktion key_handler auf.

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

```
PHILOSOPHY_OBJECTS = philosophy.o answer.o

philosophy: $(PHILOSOPHY_OBJECTS)
        gcc $(PHILOSOPHY_OBJECTS) -o philosophy
```

$\longrightarrow$ 3 Sprachen: C, Präprozessor, make

4 Hardwarenahe Programmierung

4.1 Bit-Operationen

C-Operator	Verknüpfung
&	Und
|	Oder
^	Exklusiv-Oder
~	Nicht

4 Hardwarenahe Programmierung

4.1 Bit-Operationen

C-Operator	Verknüpfung
&	Und
|	Oder
^	Exklusiv-Oder
~	Nicht
<<	Verschiebung nach links
>>	Verschiebung nach rechts

4 Hardwarenahe Programmierung

4.1 Bit-Operationen

C-Operator	Verknüpfung	Anwendung
&	Und	Bits gezielt löschen
|	Oder	Bits gezielt setzen
^	Exklusiv-Oder	Bits gezielt invertieren
~	Nicht	Alle Bits invertieren
<<	Verschiebung nach links	Maske generieren
>>	Verschiebung nach rechts	Bits isolieren

4 Hardwarenahe Programmierung

4.1 Bit-Operationen

C-Operator	Verknüpfung	Anwendung
&	Und	Bits gezielt löschen
|	Oder	Bits gezielt setzen
^	Exklusiv-Oder	Bits gezielt invertieren
~	Nicht	Alle Bits invertieren
<<	Verschiebung nach links	Maske generieren
>>	Verschiebung nach rechts	Bits isolieren

Aufgabe: Schreiben Sie C-Funktionen, die ein „Array von Bits“ realisieren, z. B.

void set_bit (**int** i);	Bei Index *i* auf 1 setzen
void clear_bit (**int** i);	Bei Index *i* auf 0 setzen
int get_bit (**int** i);	Bei Index *i* lesen

4 Hardwarenahe Programmierung

4.1 Bit-Operationen

C-Operator	Verknüpfung	Anwendung
&	Und	Bits gezielt löschen
|	Oder	Bits gezielt setzen
^	Exklusiv-Oder	Bits gezielt invertieren
~	Nicht	Alle Bits invertieren
<<	Verschiebung nach links	Maske generieren
>>	Verschiebung nach rechts	Bits isolieren

Anwendungen:

- Speicherplatz effizient nutzen
- Kommunikation mit externen Geräten
- speziell: Grafiktreiber