Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Programmieren in C

Modularisierung
(aus Tag 21 in «C in 21 Tagen»)

Dr. Adrian Koller, Büro E307, adrian.koller@hslu.ch

Mit mehreren Quelldatei arbeiten

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Code organisieren, was zusammen gehört zusammen halten
  → was nicht, separat!
  - Quellcode aufteilen in
    - Header File bitOperation.h
    - Source File bitOperation.c
  - In main.c

```
#include <stdio.h>
#include ‘’bitOperation.h’’
```

    ‘’ ‘’ → compiler sucht lokal im Ordner
    < > → compiler sucht im Systempfad

  - Compilieren:
    gcc –Wall –g main.c bitOperation.c –o myProgram

Knowhow schützen

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Nur eine Header Datei und eine bereits kompilierte Bibliothek publizieren

→ Funktionen können verwendet werden,
Details der Implementierung bleiben versteckt

Beispiel:	HALCON	(Bildverarbeitung)
	MATLAB	(Engineering Software)
	…	

Code Organisation

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

```
// Hauptmodul

#include ‘’bitOperation.h’’

int main(…)
{
..
}
```

```
// Source File:
// Modul Bit Operation

#include ‘’bitOperation.h’’

void SetBitN(..)
{
..
}
```

```
// Header File:
// Modul Bit Operation

#include <importantLib.h>

// Function Prototypes
void SetBitN(..);
```

Code Erzeugung: Präprozessor

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

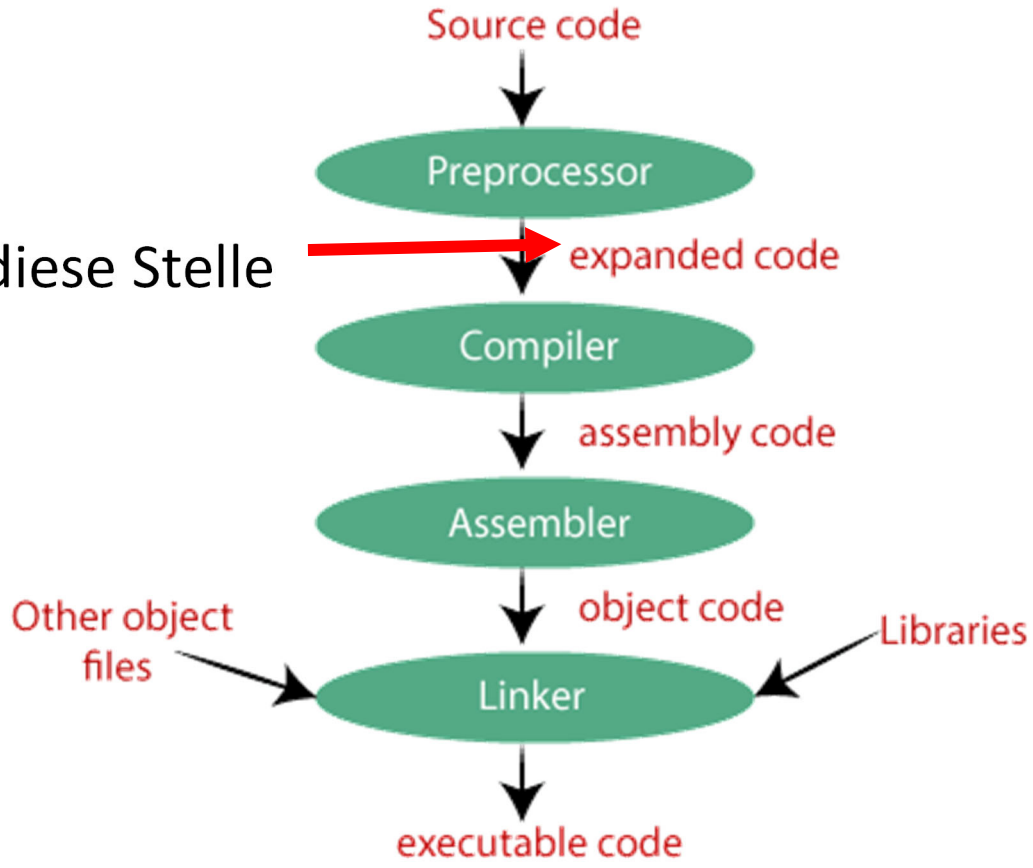
- Präprozessor verarbeitet alle # - Direktiven

  #include <stdio.h>

  → kopiert dieses File an diese Stelle

Bedingte Kompilierung

```
#define DEBUG
#ifdef DEBUG
        //blah
#endif
```



Mehrere Files zusammenkompilieren

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

gcc mainModule.c sideModule.c –o main

→Zuerst alle Source Files aufliste

→ -o zeigt an «Output»

→ main = Ziel für den Output = Executable

Für grössere Projekt «make» verwenden

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

Übung

- Schreibe ein Programm **helloModule**

  Modul hello.c / hello.h:

  enthält Funktion void helloWorld(void)

  → Schreibt «Hello World» an die Konsole

  Hauptmodul main.c:

  enthält Funktion «main», ruft «helloWorld» auf.

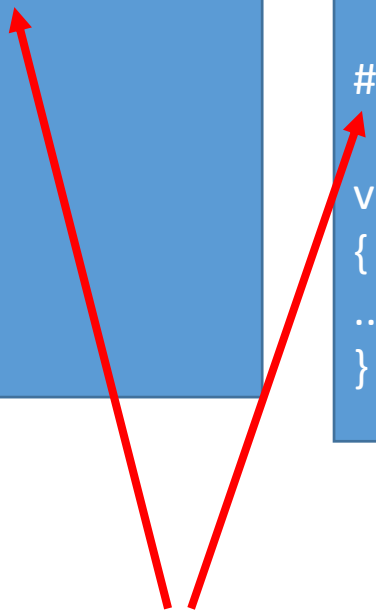
Header einbinden

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

```
// Hauptmodul

#include ‘’bitOperation.h’’

int main(…)
{
..
}
```

```
// Source File:
// Modul Bit Operation

#include ‘’bitOperation.h’’

void SetBitN(..)
{
..
}
```

```
// Header File:
// Modul Bit Operation

#include <importantLib.h>

// Function Prototypes
void SetBitN(..);
```



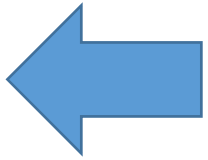
Hoppla…

«Header Guard»

Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur
FH Zentralschweiz

- Verhindert, dass Code mehrfach eingebunden wird
  → Präprozessor regelt die Einbindung

```
#ifndef _MY_HEADER_FILE_H_
#define _MY_HEADER_FILE_H_

// function protypes
// module variables

#endif // _MY_HEADER_FILE_H_
```



wird nur beim 1. Mal eingebunden