# Gdl 2 — Systemnahe Programmierung in C

## Inhalt


A Überblick über die Lehrveranstaltung . . . . . A.1
A.1 Thema: Systemnahe Programmierung in C. . . . . A.1
A.2 Aufbau der Lehrveranstaltung . . . . . A.2
1 Vorlesung . . . . . A.2
2 Übungen . . . . . A.2

B Organisatorisches . . . . . B.3
B.1 Vorlesungsbetrieb . . . . . B.3
B.2 Vorlesungsskript . . . . . B.4
B.3 Literatur . . . . . B.4
B.4 Übungen . . . . . B.5
B.5 Programmieraufgaben . . . . . B.7
B.6 Prüfung . . . . . B.7




Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.



C Systemarchitekturen . . . . . C.1
C.1 Softwareschichten . . . . . C.2
C.2 Was sind Betriebssysteme?. . . . . C.5
1 Verwaltung von Betriebsmitteln. . . . . C.6
2 Klassifikation von Betriebssystemen . . . . . C.8
3 Betriebssystemarchitekturen. . . . . C.10
4 Betriebssystemkomponenten . . . . . C.11

D Einführung in die Programmiersprache C . . . . . D.1
D.1 C vs. Java . . . . . D.1
1 C hat nicht . . . . . D.2
2 C hat . . . . . D.2
D.2 Sprachüberblick . . . . . D.3
1 Erstes Beispiel. . . . . D.3
2 Aufbau eines C-Programms . . . . . D.5
3 Allgemeine Form eines C-Programms: . . . . . D.6
4 wie ein C-Programm nicht aussehen sollte: . . . . . D.7
D.3 Datentypen. . . . . D.8
1 Was ist ein Datentyp? . . . . . D.9
2 Standardtypen in C . . . . . D.10
3 Variablen . . . . . D.12
4 Ganze Zahlen . . . . . D.15
5 Fließkommazahlen . . . . . D.16
6 Zeichen . . . . . D.17
7 Zeichenketten (Strings) . . . . . D.19




Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

D.4 Ausdrücke . . . . . D.20
D.5 Operatoren . . . . . D.21
1 Zuweisungsoperator = . . . . . D.21
2 Arithmetische Operatoren . . . . . D.22
3 spezielle Zuweisungsoperatoren . . . . . D.23
4 Vergleichsoperatoren . . . . . D.24
5 Logische Operatoren . . . . . D.25
6 Bitweise logische Operatoren . . . . . D.27
7 Logische Shiftoperatoren . . . . . D.28
8 Bedingte Bewertung . . . . . D.30
9 Komma-Operator . . . . . D.31
10 Typumwandlung in Ausdrücken . . . . . D.32
11 Vorrangregeln bei Operatoren . . . . . D.33
D.6 Einfacher Programmaufbau . . . . . D.34
1 Struktur eines C-Hauptprogramms . . . . . D.35
2 Anweisungen . . . . . D.35
3 Blöcke . . . . . D.36
4 Einfache Ein-/Ausgabe . . . . . D.37
5 Einzelzeichen E/A . . . . . D.39
6 Formatierte Ausgabe . . . . . D.40
7 C-Präprozessor — Kurzüberblick . . . . . D.42
8 C-Präprozessor — Makrodefinitionen . . . . . D.43
9 C-Präprozessor — Einfügen von Dateien . . . . . D.44


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

D.7 Kontrollstrukturen . . . . . D.45
1 Bedingte Anweisung . . . . . D.46
2 Fallunterscheidung . . . . . D.51
3 Schleifen . . . . . D.53
4 abweisende Schleife . . . . . D.53
5 nicht-abweisende Schleife . . . . . D.55
6 Laufanweisung . . . . . D.56
7 Schleifensteuerung . . . . . D.58
D.8 Funktionen . . . . . D.59
1 Überblick . . . . . D.59
2 Beispiel Sinusberechnung . . . . . D.61
3 Funktionsdefinition . . . . . D.62
4 Funktionsaufruf . . . . . D.65
5 Beispiel . . . . . D.67
6 Regeln . . . . . D.68
7 Funktionsdeklarationen — Beispiel . . . . . D.71
8 Parameterübergabe an Funktionen . . . . . D.72
D.9 Programmstruktur & Module . . . . . D.74
1 Softwaredesign . . . . . D.74
2 Top-down Entwurf . . . . . D.75
3 Module in C . . . . . D.78
4 Gültigkeit von Namen . . . . . D.80
5 Globale Variablen . . . . . D.81
6 Einschränkung der Gültigkeit auf ein Modul . . . . . D.84
7 Lokale Variablen . . . . . D.85
8 Gültigkeitsbereiche — Übersicht . . . . . D.86


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

9 Lebensdauer von Variablen . . . . . D.87
10 Getrennte Übersetzung von Programmteilen — Beispiel . . . . . D.90

E Vom C-Programm zum laufenden Prozess . . . . . E.1
E.1 Übersetzen - Objektmodule . . . . . E.1
E.2 Binden und Bibliotheken . . . . . E.3
E.3 Programme und Prozesse . . . . . E.5
E.4 Speicherorganisation eines Prozesses . . . . . E.6

F Zeiger, Felder und Strukturen in C . . . . . F.1
F.1 Zeiger(-Variablen) . . . . . F.1
1 Einordnung . . . . . F.1
2 Überblick . . . . . F.2
3 Definition von Zeigervariablen . . . . . F.3
4 Beispiele . . . . . F.3
5 Adressoperatoren . . . . . F.4
6 Zeiger als Funktionsargumente . . . . . F.5
F.2 Felder . . . . . F.7
1 Eindimensionale Felder . . . . . F.7
2 Initialisierung eines Feldes . . . . . F.8
3 Zugriffe auf Feldelemente . . . . . F.10
4 Mehrdimensionale Felder . . . . . F.11
5 Zugriffe auf Feldelemente bei mehrdim. Feldern . . . . . F.12
6 Initialisierung eines mehrdimensionalen Feldes . . . . . F.13


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

F.3 Zeiger und Felder . . . . . F.14
1 Arithmetik mit Adressen . . . . . F.15
2 Vorrangregeln bei Operatoren . . . . . F.16
3 Beispiele . . . . . F.16
4 Zeigerarithmetik und Felder . . . . . F.18
5 Vergleichsoperatoren und Adressen . . . . . F.21
F.4 Eindimensionale Felder als Funktionsparameter . . . . . F.22
F.5 Zeiger, Felder und Zeichenketten . . . . . F.24
F.6 Dynamische Speicherverwaltung . . . . . F.30
F.7 Explizite Typumwandlung — Cast-Operator . . . . . F.32
F.8 sizeof-Operator . . . . . F.33
F.9 Felder von Zeigern . . . . . F.34
F.10 Argumente aus der Kommandozeile . . . . . F.36
1 Datenaufbau . . . . . F.37
2 Zugriff - Beispiel: Ausgeben aller Argumente (1) . . . . . F.38
F.11 Zeiger auf Zeiger . . . . . F.40
F.12 Zeiger auf Funktionen . . . . . F.41
F.13 Strukturen . . . . . F.42
1 Motivation . . . . . F.42
2 Deklaration eines Strukturtyps . . . . . F.43
3 Definition einer Struktur . . . . . F.45
4 Zugriff auf Strukturkomponenten . . . . . F.46
5 Initialisieren von Strukturen . . . . . F.47
6 Strukturen als Funktionsparameter . . . . . F.48
7 Zeiger auf Strukturen . . . . . F.49
8 Felder von Strukturen . . . . . F.51


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

9 Strukturen in Strukturen . . . F.52
10 Rekursive Strukturen . . . F.53
F.14 Verbundstrukturen — Unions . . . F.57
F.15 Bitfelder . . . F.58
F.16 Typedef . . . F.61
F.17 Ergänzungen zum Präprozessor . . . F.62
1 Parametrierte Makros . . . F.62
2 Deaktivieren von Makros . . . F.64

G Ein-/Ausgabe . . . G.1
G.1 Überblick . . . G.1
G.2 Standard Ein-/Ausgabe . . . G.2
G.3 Öffnen und Schließen von Dateien . . . G.3
G.4 Zeichenweise Lesen und Schreiben . . . G.5
G.5 Formatierte Ausgabe . . . G.8
1 Bibliotheksfunktionen — Prototypen (Schnittstelle) . . . G.8
2 Formatangaben . . . G.9
G.6 Formatierte Eingabe . . . G.10
1 Bibliotheksfunktionen — Prototypen (Schnittstelle) . . . G.10
2 Bearbeitung der Eingabe-Daten . . . G.11
3 Format-Anweisungen . . . G.12


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

H Dateisysteme . . . H.1
H.1 Allgemeine Konzepte . . . H.1
H.2 Allgemeine Konzepte (2) . . . H.2
H.3 Beispiel: UNIX (Sun-UFS) . . . H.4
1 Pfadnamen . . . H.5
2 Programmierschnittstelle für Kataloge . . . H.7
3 Programmierschnittstelle für Dateien . . . H.10
4 Inodes . . . H.11

I Speicherorganisation eines Prozesses . . . I.1
I.1 Stackaufbau eines Prozesses . . . I.2
1 Prinzip . . . I.2
2 Beispiel . . . I.3

J Mikrocontroller-Programmierung . . . J.1
J.1 Überblick . . . J.1
J.2 Mikrocontroller vs. Betriebssystem-Plattform . . . J.2
1 Betriebssystemunterstützung . . . J.3
2 Mikrocontroller-Umgebung . . . J.4
J.3 Beispiel: AVR-Mikrocontroller (ATmega-Serie) . . . J.5
1 Architektur . . . J.5
2 In Speicher eingebettete Register (memory mapped registers) . . . J.7
3 I/O-Ports . . . J.8
4 Programme laden . . . J.9
5 Programm starten . . . J.10


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

6 Fehler zur Laufzeit . . . . . J.10
J.4 Interrupts . . . . . J.11
1 Motivation . . . . . J.11
2 Implementierung . . . . . J.13
3 Ablauf . . . . . J.14
4 Pegel- und Flanken-gesteuerte Interrupts . . . . . J.15
J.5 Nebenläufigkeit . . . . . J.16
1 Überblick . . . . . J.16
2 Nebenläufigkeit durch Interrupts . . . . . J.17
3 Nebenläufigkeitsprobleme allgemein . . . . . J.22
4 Umgang mit Nebenläufigkeitsproblemen . . . . . J.23
J.6 Programmierung eingebetteter Systeme . . . . . J.25
1 Misra C . . . . . J.26
2 Java für eingebettete Systeme? . . . . . J.27
3 Objektorientierte Programmierung . . . . . J.27
4 Typsichere Sprachen . . . . . J.28
5 Java für eingebettete Systeme . . . . . J.30
6 Modellbasiertes Design . . . . . J.31
7 Familienbasierter Ansatz . . . . . J.32

K letzte Infos / Organisatorisches . . . . . J.33


Reproduktion jeder Art oder Verwendung dieser Unterlage, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.