

Technische Informatik 3 – Embedded Systems

Kapitel FF: Prüfungsvorbereitung

Prof. Dr. Benjamin Kormann

Fakultät für Elektro- und Informationstechnik

26.06.2023



Prüfungsvorbereitung

Informationen zur Prüfung

Rahmenbedingungen

- Termin: Montag, 17.07.2023, 16:00 Uhr im T-Gebäude
- Modulprüfung: schriftlich, 90 Minuten, 90 Punkte
 - Programmieren: 60 Punkte (Mindestpunktzahl: 10)
 - Embedded Systems: 30 Punkte (Mindestpunktzahl: 5)
- Die Bonuspunkte werden auf die erreichte Gesamtpunktzahl addiert
- Hilfsmittel: beliebige schriftliche

Freie Zeiteinteilung



Rechnerprüfung EXaHM

- Programmieren: Bearbeitung am Rechner in Visual Studio Code
 - Angabe in PDF, Antworten in Visual Studio Code (ausführbarer Code sowie Antworten auf Verständnisfragen)
 - Vorlesungsunterlagen werden als PDF im Prüfungssystem eingeblendet
- Embedded Systems auf Blatt Papier
- Prüfungsunterlagen werden vollständig zu Beginn ausgeteilt

Prüfungsvorbereitung

Wichtige Prüfungshinweise

Eigene freie Zeiteinteilung

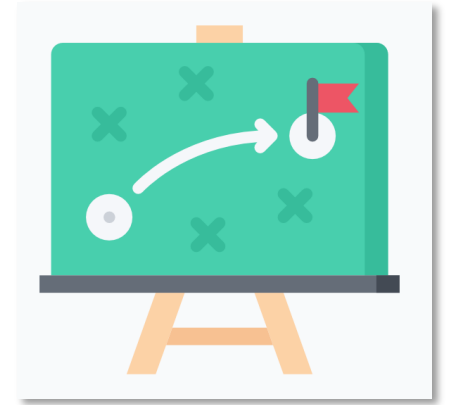
- Bringen Sie eine eigene Uhr mit (kein Smart Device)

Legen Sie sich eine Bearbeitungsstrategie zu

- Zeiteinteilung für die Bearbeitung der beiden Teile (Programmieren, Embedded) sowie Aufgaben
- Kennen Sie Ihre Stärken & Schwächen

Verlieren Sie sich nicht im Detail (besonders am Rechner)

- KEIN Over-Engineering betreiben
- Ausführen und Debugging kosten wertvolle Zeit
- Syntaktische Sicherheit in Python und C



Prüfungsvorbereitung

Relevante Themen für die Prüfung

TI3-Embedded Systems

- Grundlagen: Programmaufbau, Datentypen, Funktionen, Kontrollstrukturen, Kompilieren
- Ein-/Ausgabe: Konsole und Datei
- Zeiger: Syntax, Call-by-X, Funktionszeiger, Array, (Dynamische) Speicherverwaltung
- Struktur: Definition, Anwendung
- Linux: Prozessinformationen, Prozesserzeugung, Signale
- Bildverarbeitung: Bildkodierung

Ausgeschlossen

- Unit Tests, CI/CD, Git

TI3-Programmieren

- Kap 1: Datentypen, Programmaufbau, Hauptmodul, dynamische Typisierung, EAFP, built-in: enumerate, zip
- Kap 2: Algorithmus, if, for, while, def
- Kap 3: Slicing, {List,Dict} Comprehension
- Kap 4: Call-by-X, Parameter, Rekursion, Lambda Funktionen
- Kap 5: Module, Dateibearbeitung, Exceptions
- Kap 7: UML, Klassen, Objekte, Sichtbarkeit, get/set, Properties, statische Variablen, Vererbung, überschreiben, abstrakte Klassen
- Kap 8: Type Hints, Logging

Ausgeschlossen

- Kap 6: XML, JSON, struct
- Profiling, Unit Tests, CI/CD, Git

Prüfungsvorbereitung

Übungsaufgabe 1: Rekursion

Aufgabe 1: Ausgabe

- Implementieren Sie eine rekursive Funktion `void print_rec(int num)`, die für `num = 5` folgende Ausgabe erzeugt:
5 4 3 2 1 0 1 2 3 4 5

Aufgabe 2: Schriftliches Addieren

- In der Grundschule wird das schriftliche Addieren wie folgt gelehrt:

The diagrams illustrate the standard columnar addition of 5738 and 3884, showing the progression of carry-over (blue numbers) from right to left:

$\begin{array}{r} 5738 \\ + 3884 \\ \hline \end{array}$	$\begin{array}{r} 5738 \\ + 3884 \\ \hline 2 \\ \hline \end{array}$	$\begin{array}{r} 5738 \\ + 3884 \\ \hline 22 \\ \hline \end{array}$	$\begin{array}{r} 5738 \\ + 3884 \\ \hline 622 \\ \hline \end{array}$	$\begin{array}{r} 5738 \\ + 3884 \\ \hline 9622 \\ \hline \end{array}$	$\begin{array}{r} 5738 \\ + 3884 \\ \hline 9622 \\ \hline \end{array}$
---	---	--	---	--	--

- Implementieren Sie eine rekursive Funktion `int add(int a, int b)`, die zwei Zahlen gemäß dem oben dargestellten Verfahren addiert.

Prüfungsvorbereitung

Übungsaufgabe 2: Bildverarbeitung

Aufgabe 1: Bild spiegeln

- Gegeben ist ein eindimensionaler Array vom Typ `uint8_t`, der Bildinformationen eines 8 Bit Graustufenbildes beinhaltet. Die Dimensionen in Pixel sind `width` und `height`.
- Entwickeln Sie einen Algorithmus, der das Bild in der Mitte horizontal spiegelt (Pixel 0 wird mit Pixel `width-1` getauscht)
- Hinweis: Gehen Sie von einem Bild mit geraden Bilddimensionen aus

Aufgabe 2: Bild punktspiegeln

- Gegeben ist ein eindimensionaler Array vom Typ `uint16_t`, der Bildinformationen eines 16 Bit Graustufenbildes beinhaltet. Die Dimensionen in Pixel sind `width` und `height`.
- Entwickeln Sie einen Algorithmus, der eine Punktspiegelung des Bildes um den Mittelpunkt realisiert (Pixel 0 wird mit Pixel `width*height-1` getauscht)
- Hinweis: Gehen Sie von einem Bild mit gerade Bilddimensionen aus

Prüfungsvorbereitung

Übungsaufgabe 3: Doppelt-verkettete Liste

Aufgabe 1: Definieren Sie eine Struktur zur Repräsentation einer doppelt-verketteten Liste

- Die Struktur soll neben den Verwaltungselementen (Zeiger auf Vorgänger und Nachfolger) noch eine `id` beinhalten

```
struct node
{
    // TODO
};
```

Aufgabe 2: Implementieren Sie eine Funktion `struct node createNode(int id)`

- Die Funktion soll einen neuen Knoten mit einer angegebenen `id` erstellen
- Der Rückgabewert ist kein Zeiger auf diesen neu erstellten Knoten
- Hinweis: Achten Sie auf das Speichermanagement

Aufgabe 3: Implementieren Sie eine Funktion `void addNode(struct node* root, struct node* newNode)`

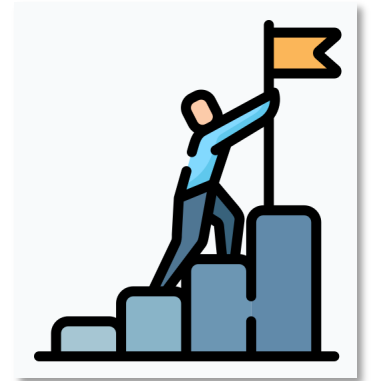
- Einer bereits vorhandenen doppelt-verketteten Liste `root` soll ein weiterer Knoten `newNode` am Ende angehängt werden
- Achten Sie dabei auf korrekte Zeigerarithmetik
- Hinweis: Der Vorgänger des ersten Knotens und der Nachfolger des letzten Knotens müssen auf `NULL` zeigen

Prüfungsvorbereitung

Zusammenfassung

TI3 bedeutet Softwareentwicklung

- Die Entwicklung von Software gelingt nur durch eigenständiges und häufiges Üben
- Sie erlangen nur dann ein tiefgreifendes Verständnis, wenn Sie regelmäßig viele eigene Programme schreiben



Viel Erfolg!