

Systemnahe Programmierung SS 2024

Übersicht

Helmut Lindner

Allgemeines

Lernziele

*Verständnis der grundlegenden Programmierkonzepte auf Linux-Systemebene.
Gute Kenntnisse der Programmiersprache C und der verwendeten POSIX
Systemschnittstellen.*

Sie

- erwerben gute Kenntnisse der Programmiersprache C.
- wissen wie das Betriebssystem ihre Programme ausführt.
- verstehen die dynamische Speicherverwaltung in C Programmen.
- kennen das POSIX-API und sind in der Lage es für eigene Lösungen einzusetzen.
- verstehen, welche Mittel das Betriebssystem für Anwendungsprogramme zur Verfügung stellt.
- können Prozesse und Threads in C-Programmen verwenden
- wissen Bescheid um Prozesssynchronisation und Interprozesskommunikation

Voraussetzungen

- Grundlegende Linux und Shell Kenntnisse
- Programmierkenntnisse

Allgemeines

Struktur der Übungen

- Theorieeinheit(en)
- Nach jeder Theorieeinheit führen wir Praxisübungen durch, um das Gelernte zu vertiefen.
- Die Übungen sind zuerst im Moodle (VPL), später lokal auf ihrem Rechner durchzuführen.
- Eine längere Aufgabe wird als Projektarbeit durchgeführt
- Die Unterlagen werden im Moodle zur Verfügung gestellt
- Ihre Anwesenheit bitte selbst im Moodle erfassen (Punkt Allgemeines/Anwesenheit)

Kommunikation

- Feedback ist ausdrücklich erwünscht
- Es besteht immer die Möglichkeit auf Punkte nochmals einzugehen, die beim ersten Mal nicht klar geworden sind.
- Am besten gleich fragen wenn etwas nicht klar ist ...
- Sonst im Büro vorbeischaun oder per E-Mail
helmut.lindner@fh-joanneum.at

Allgemeines

Inhalte

- Einführung in die systemnahe Programmierung
- Einstieg in die C-Programmierung
- Variable und Datentypen
- Operatoren
- Kontrollstrukturen
- Funktionen
- Zeiger
- Dynamische Speicherverwaltung
- I/O Funktionen
- Prozesse
- Interprozesskommunikation
- Netzwerkprogrammierung

Allgemeines

Prüfungsmodalitäten

1. Abgabe der Übungsaufgaben (30%)

Beurteilung der Richtigkeit und Genauigkeit der Aufgaben.

2. Abschlussprojekt (40%)

Beurteilung der Richtigkeit und Qualität der Projektimplementierung

3. Überprüfung (30%)

Fragen über den Code und die Theorie aus den Einheiten und den Übungs-
Abgaben (kurzer Test).

Allgemeines

Punkteschlüssel

Werte in Prozent der erreichbaren Punkte

100-91 -> Sehr gut

90-81 -> Gut

80-71 -> Befriedigend

70-61 -> Genügend

≤ 60 -> Nicht genügend

Übungsaufgaben

Modus

- Die Übungsaufgaben sind alleine zu lösen.
- Übungsaufgaben die abgeschrieben wurden, werden als Ganzes als ungültig (0 Punkte) bewertet.
- Die Übungen der ersten Einheiten können direkt im Moodle VPL bearbeitet werden.
- Komplexere Aufgaben bitte im e-Learning hochladen (wird angekündigt wenn notwendig).

Übungsaufgaben

Format der Abgaben (nicht VPL)

- Alle Bestandteile die zur Lösung gehören (Sourcecode, Makefile, Texte,...) in einem eigenen Unterverzeichnis ablegen.
- Dieses Unterverzeichnis ist in einer .zip Datei zusammenfassen.
- Dateiname der .zip Datei:
<Vorname>_<Nachname>_Uebung_<Nummer der Übungsaufgabe>.zip
Beispiel: Helmut_Lindner_Uebung_1.zip
- Fragen die in Textform zu beantworten sind, als Text oder PDF Dokument ablegen.

Übungsaufgaben

Format Quellcode (nicht VPL)

In jeder Sourcecode-Datei einen Header mit folgenden Angaben:
Name, Personenkennzeichen, Nummer der Übung und Beispiel einfügen:

```
/*  
    Systemnahe Programmierung SS 2024  
  
    <Vorname> <Nachname>  
    Übung <Nr> Beispiel <Nr>  
  
*/
```

Auch in Textdokumenten

```
<Vorname> <Nachname>  
Übung <Nr> Beispiel <Nr>
```

am Anfang der Datei angeben.

Übungsaufgaben

Anforderungen

- Die Programmierbeispiele müssen ohne Fehlermeldungen und ohne Warnings compilierbar sein.
- Der Code muss lesbar sein: Einrückungen, Naming usw.
- Der Output muss verständlich sein, und sich auf die Aufgabe beziehen
-> gefragte Ergebnisse müssen den Ausgaben des Programms ersichtlich sein
Beispiel: Schreiben sie ein Programm, das von einer Zahl berechnet, ob sie gerade oder ungerade ist.
Output: Die Zahl 3 ist ungerade. oder 3 : ungerade
- Die beschriebenen Anforderungen müssen eingehalten werden (Punkteabzug).

Abschlussprojekt

Modus

- Das Abschlussprojekt wird als Einzel- oder Teamarbeit durchgeführt (noch offen).
- Dauer: ca. 2-3 Einheiten
- Die formalen Anforderungen sind analog denen der Übungsaufgaben.

Tools

Programmierungsumgebung und Tools

Ein virtuelle Maschine (Oracle VirtualBox) mit allen benötigten Tools steht zum Download unter

https://fhjoanneum-my.sharepoint.com/:u:/g/personal/helmut_lindner_fh-joanneum_at/EcuYYEVI5otHg8SnHrPucsUBniR6fkJMYi6B2Q50UurhGw?e=NeoLfh

bereit.

Der Link zum Download ist auch im Moodle verfügbar.

Ich empfehle diese VM zu verwenden, grundsätzlich können sie auch ihr eigenes Linux verwenden.

Allgemeines

Literatur

- C/C++: Das umfassende Lehrbuch. Ulrich Kaiser, Christoph Kecher. Galileo Press. ISBN: 978-3836227575
- C von A bis Z. Jürgen Wolf. Rheinwerk Verlag. ISBN 978-3-8362-1411-7
Auch als Openbook verfügbar:
http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/
- Linux-Unix Programmierung. Jürgen Wolf, Klaus-Jürgen Wolf. Rheinwerk Verlag. ISBN 978-3-8362-3772-7
Auch als Openbook verfügbar:
http://openbook.rheinwerk-verlag.de/linux_unix_programmierung
- Linux System Programming. Robert Love. O'Reilly Media. ISBN: 978-1-449-33953-1