

Shell und Prozesse (WPF I4/I6 Bc)

Blatt 01: C-Programmierung unter Linux

Sommersemester 2024

Bearbeitung im Praktikum ab 04.04.2024

Aufgabe A11: Ausgabe auf Standardausgabe

Schreiben Sie ein Programm, welches die diesem Programm übergebenen Aufrufparameter auf der Standardausgabe ausgibt. Stellen Sie die Ausgabe zeilenweise in der Form

Nummer des Aufrufparameters: Wert des Aufrufparameters
dar.

Hinweis:

Die Lösung dieser Aufgabe wird bei einem späteren Themenblatt als Testprogramm benötigt.

Lernziel:

- Aufrufparameter eines Programms im Programm verarbeiten können.

Aufgabe A12: Eingabe von Standardeingabe

Schreiben Sie ein Programm, welches mehrere Namen von der Standardeingabe einliest und in umgekehrter Reihenfolge auf die Standardausgabe wieder ausgibt. Die Anzahl der einzulesenden Namen soll über einen Aufrufparameter festgelegt werden können. Verwenden Sie **nur ein Feld** als Zwischenspeicher für die Namen.

Achten Sie darauf, dass beim Einlesen der Namen nicht über den Eingabepuffer hinausgeschrieben wird.

Lernziele:

- Im Programm interaktiv Eingaben einlesen können.
- Dynamisch Speicher allokieren und wieder freigeben können.

Optionale Aufgabe A13: Speicherverwaltung

Gehen Sie von Ihrem Programm aus Aufgabe 2 aus und ändern Sie dieses so ab, dass es neben den Nach- auch die Vornamen einliest, die Informationen in einer **doppelt verketteten Liste** speichert und in umgekehrter Reihenfolge wieder auf der Standardausgabe ausgibt.

Lernziel:

- Mit Zeiger arbeiten können und den Umgang mit dynamischem Speicher vertiefen.

Optionale Aufgabe A14: Dateibehandlung

Lesen Sie die Vor- und Nachnamen der Personen in Ihrem Programm aus einer **ASCII-Datei** mit Namen "quelle" ein und schreiben Sie diese wie in Aufgabe 3 in umgekehrter Reihenfolge in eine andere ASCII-Datei mit Namen "umgekehrte-reihenfolge" heraus.

Lernziel:

- Dateien lesen und schreiben können.

Allgemeine Hinweise

Folgende C-Funktionen sind für die Lösung der Aufgaben hilfreich:

- `printf()` // Ausgabe auf die Standardausgabe
- `sscanf()` // Einlesen und Umwandeln von String in int
- `fgets(buffer, len, stdin)` // Einlesen von Standardeingabe mit \n
- `malloc(), free()` // Allokieren und Freigeben von Speicher
- `fopen(), fclose()` // Öffnen und Schließen einer Datei