

Bitte bearbeiten Sie die Aufgaben bis zum 8.11.2024, 8:00 Uhr. Bitte schicken Sie Ihre Lösung, Output und Quellcode als tar archive an marks@physi.uni-heidelberg.de mit Angabe Ihres Names.

Aufgabe 1: File IO und Funktionen

Schreiben Sie ein Programm, das das File "[ZeitWegMessung.txt](#)" liest. Das File enthält die Daten von Fahrzeugen, die eine Teststrecke mit der gleichen Geschwindigkeit durchfahren. Es werden Zeiten und zurückgelegte Wegstrecken gemessen. In der ersten Spalte steht der gemessene Zeitnullpunkt. In der zweiten die gemessene Zeit und in der dritten die zurückgelegte Wegstrecke.

- Bestimmen Sie im Programm die Anzahl der Messungen.
- Speichern Sie die Daten aus dem Text File spaltenweise in 3 Arrays unter Verwendung von `<vector>`
- Berechnen Sie die mittlere Geschwindigkeit und geben Sie einen Fehler der Messung an. Schreiben Sie dazu 2 Funktionen, die die Daten in Form von vector arrays übernehmen. Beide Werte und die Anzahl der gelesenen Messungen sollten in einem File "results.txt" gespeichert werden.
- Erzeugen Sie eine Bibliothek libmyutil.a, die beide Funktionen enthält und das entsprechende header File myutil.h. Wie wird die Bibliothek verwendet?

Aufgabe 2: Prozessieren von Daten auf der Shell

Betrachten Sie noch einmal das File "ZeitWegMessung.txt" .Nutzen Sie für die folgende Bearbeitung nur Linux commands. Die Lösungen sind in maximal 2 Zeilen teilweise mit einem Zwischenschritt auszuführen. Eine Mikrobeschreibung des Linux Tools awk finden Sie [hier](#).

- Geben Sie die Messungen in einem Terminal aus.
- Bestimmen Sie die Anzahl der Messungen.
- Verwenden Sie awk um ein neues File zu erzeugen, in dem die zweite und dritte Spalte vertauscht sind.
- Subtrahieren Sie den mittleren gemessenen Zeitnullpunkt von den gemessenen Zeitwerten und erzeugen Sie ein neues File mit den subtrahierten gemessenen Zeitwerten als vierte Spalte.
- Bestimmen Sie die Geschwindigkeit und erzeugen Sie ein neues File mit der Geschwindigkeit als 5. Spalte. Berechnen Sie die mittlere Geschwindigkeit.

Aufgabe 3: Function pointer

Schreiben Sie eine Funktion `integrate(...)`, die für eine beliebige gegebene Funktion eine numerische Integration durchführt. Die Funktion soll mit Hilfe von Funktionszeigern übergeben werden.

Wie wird ein tar archive erzeugt?

- i) Erzeugen Sie ein directory (`mkdir Aufgabe_1`), das den Quellcode, ein Textfile mit dem Output und eventuell weitere Daten enthält.
- ii) Mit `tar -cvf meinDirectory.tar meinDirectory` erzeugen Sie das tar file `meinDirectory.tar`