

Hardwarenahe Programmierung Gruppe 14 (Lars)

In den bisherigen Übungen haben Sie Nutzereingaben zur Laufzeit des Programms über Tastatureingaben implementiert. In dieser Übung schreiben Sie C-Programme, die ihren Input als Kommandozeilen-Parameter und -Optionen erhalten, und die Ausgabedaten ebenfalls in Dateien schreiben können.

Wichtig: Falsch formatierte Eingabedaten müssen Sie nicht abfangen. Denken Sie wie immer daran, Ihre Lösungen im ILIAS hochzuladen und den Test im ILIAS zu absolvieren!

Aufgabe 1 Kleiner Taschenrechner (**Pflichtaufgabe**)

In dieser Aufgabe sollen Sie einen Taschenrechner implementieren, der Anweisungen über Kommandozeilenparameter entgegennimmt.

(a) Erstellen Sie hierzu eine Datei *taschenrechner.c*. Das Programm beginnt mit der Zahl 1 und soll drei Optionen behandeln können:

- **-m zahl:** Multipliziert mit **zahl**.
- **-d zahl:** Dividiert durch **zahl**. Sie dürfen annehmen, dass **zahl** größer als 0 ist.
- **-h:** Gibt eine Hilfestellung zur Benutzung des Taschenrechners aus. In der Ausgabe muss das Wort **Hilfe** enthalten sein.

Alle restlichen Parameter, die nicht verwendet wurden, sollen zum Schluss auf das Endergebnis aufaddiert werden.

Bei folgendem Beispielaufruf wird mit der Zahl 10 multipliziert, durch 5 dividiert, mit 20 multipliziert und die restlichen Parameter 60 und 100 dazu addiert:

```
> ./taschenrechner -m 10 60 -d 5 -m 20 100
200
```

Sie dürfen annehmen, dass alle Eingaben richtig formatiert sind. Ihr Programm muss also keine sinnvolle Ausgabe liefern, wenn ein Benutzer versucht, zum Beispiel Enten anstatt Zahlen zu addieren:

```
> ./taschenrechner Ente Ente Ente
Gans
```

Verwenden Sie das Skript `test.sh`, um ihr Programm zu überprüfen.

Aufgabe 2 *Lebensmittel (Pflichtaufgabe)*

In dieser Aufgabe verwalten Sie einen Supermarkt. Sie erhalten eine Liste von Lebensmitteln über die Standardeingabe, die Sie nach Bezeichnung, Alter, Kategorie und Mindesthaltbarkeit filtern und ausgeben sollen.

- Die Liste *lebensmittel.txt* enthält vier Spalten. In der ersten Spalte steht die Bezeichnung, in der zweiten das Alter (in Tagen), in der dritten die Kategorie und in der vierten die Mindesthaltbarkeit:

Kirschkuchen	3	Backware	5
Kirschsaft	10	Getränk	30
Sauerkirsche	12	Obst	8

- Ihr Programm soll folgende Kommandozeilen-Optionen anbieten:
 - a** Betrachtet nur Lebensmittel, die abgelaufen sind, also älter sind als die Mindesthaltbarkeitsdauer.
 - m zahl** Betrachtet nur Lebensmittel mit einer Mindesthaltbarkeit von mindestens **zahl**.
 - b string** Betrachtet nur Lebensmittel mit der Bezeichnung **string**.
 - dateiname** Wenn ein anderer Parameter als die vorherigen drei übergeben wird, dann soll angenommen werden, dass dies ein Dateiname ist. In diesem Fall soll die Ausgabe in diese Datei anstatt auf **stdout** ausgegeben werden.
- Verwenden Sie *test.sh* zum testen Ihres Programms. Dieses Test-Skript ruft Ihr Programm mit verschiedenen Parametern auf und vergleicht die Ausgabe mit den erwarteten Ausgaben, die wir Ihnen ebenfalls zur Verfügung stellen.

Beispielaufrufe:

```
./lebensmittel < lebensmittel.txt
```

Gibt alle Lebensmittel aus.

```
./lebensmittel < lebensmittel.txt > ausgabe.txt
```

Schreibt alle Lebensmittel in die Datei **ausgabe.txt**.

```
./lebensmittel ausgabe.txt < lebensmittel.txt
```

Schreibt ebenfalls alle Lebensmittel in die Datei **ausgabe.txt**.

```
./lebensmittel -a < lebensmittel.txt
```

Gibt alle Lebensmittel aus, die abgelaufen sind.

```
./lebensmittel -b Ei < lebensmittel.txt
```

Gibt alle Eier aus.

```
./lebensmittel -a -m 4 < lebensmittel.txt
```

Gibt abgelaufene Lebensmittel aus, die mindestens 4 Tage haltbar waren.