

Hardwarenahe Programmierung
Gruppe 05 (Florian)

Es ist soweit: In dieser Übung nutzen Sie dynamische Speicherverwaltung in C! Außerdem verwenden Sie `valgrind`, um Ihre Programme auf Speicherlecks zu untersuchen.

- *Denken Sie wie immer daran, Ihre Lösungen im ILIAS hochzuladen und den Test im ILIAS zu absolvieren!*

Aufgabe 1 *Auto-Daten (`liste.c` abgeben)*

Ein Autosammler möchte Listen von Autos verwalten. Eine Liste sieht beispielsweise so aus:

```
Lamborghini Diablo  
Ferrari F50  
Lamborghini Diablo  
BMW M6
```

- (a) Diese Autos sollen nacheinander aus einer Datei in eine verkettete Liste eingelesen werden.
- (b) Wenn ein Auto mehrfach vorkommt, dann soll es an das Ende der Liste angehängt werden.
- (c) Wenn ein Auto das erste Mal vorkommt, dann soll es an den Anfang der Liste angehängt werden.
- (d) Danach sollen die Daten in eine weitere Datei ausgegeben werden.
- (e) Implementieren Sie dazu die nötigen Funktionen in `liste.c`. Die `main`-Funktion ist Ihnen in der Datei `liste.c` schon vorgegeben und darf nicht verändert werden.
- (f) Der Speicher muss dabei dynamisch mit `malloc` und `free` verwaltet werden. Abschließend muss aller allozierter Speicher freigegeben werden. Nutzen Sie `valgrind`, um dies zu überprüfen.
- (g) Testen Sie Ihr Programm mit dem Test-Skript `test.sh`.

Aufgabe 2 *Dynamisches Array (dyn_array.c und dyn_array.h abgeben)*

In dieser Aufgabe sollen Sie eine Datenstruktur erstellen, die ein String-Array mit dynamischer Kapazität implementiert. Der Speicher für die Strings muss dazu dynamisch alloziert werden. Achten Sie auf eine sorgfältige Speicherverwaltung und auf die Erfüllung aller Unittests!

- (a) Definieren Sie in der Datei *dyn_array.h* die Datenstruktur **dyn_array** mit den Feldern **length** (Größe des Arrays), **fill_level** (Füllstand des Arrays) und **strings** (das String-Array selbst). Wählen Sie für die Felder geeignete Datentypen.
- (b) Implementieren Sie die in *dyn_array.c* vorgegebenen Funktionsprototypen und prüfen Sie Ihre Implementierungen mit den vorgegebenen Unittests.
- (c) Überprüfen Sie Ihr Programm mit **valgrind**.