Institut für Informatik, Arbeitsbereich PVS WS 22/23

Prof. Sergei Gorlatch

Übungen zu Betriebssysteme

Blatt 1 (C-Programmierung)

Aufgabe 1 (Dynamische Speicherverwaltung in C)

a) **Verwendung von `malloc` und `free`.** *15 Punkte*

Vervollständigen Sie das vorgegeben Programm so, dass eine dynamische Speicherverwaltung mit zusätzlicher Überwachung der angelegten und wieder freigegebene Speicherbereiche erfolgt. Ihre Implementierung soll nicht freigegebenen Speicher erkennen und den Entwickler über diese Fehler informieren.

Implementieren Sie dazu die Funktionen in der Datei `monitoring_alloc.c` gemäß der Spezifikationen. Das mitgelieferte Programm `leaking_progam` verwendet die von `monitoring_alloc.h` bereitgestellten Funktionen zur dynamischen Speicherverwaltung. Allerdings beinhaltet dieses Programm einige Fehler bezüglich der Freigabe von Speicher, die es nun mit einer korrekten Implementierung von `monitoring_alloc` zu finden gilt.

Gehen Sie wie folgt vor:

- Verwalten Sie die mittels `monitoring_alloc` angelegten und wieder freigegebene Speicherblöcke in dem Array `allocated_blocks`. Zur Vereinfachung können sie zunächst davon ausgehen, dass höchstens `MAX_ALLOCATIONS` viele Speicherreservierungen, die zusammen nicht größer als `MAX_TOTAL_ALLOCATION_SIZE` sind, vorgenommen werden können.
- Reservieren Sie in `monitoring_alloc_malloc` mittels des Systemaufrufs `malloc` den angeforderten Speicherblock und speichern Sie die Informationen in `allocated_blocks`.
- In `monitoring_alloc_free` müssen sie den Speicher wieder freigeben und den entsprechenden Eintrag im Array wieder freigeben.
- In der Funktion `shutdown_monitoring_alloc` soll das Array auf nicht freigegebene Blöcke geprüft werden und die Anzahl zurückgeben. Für solche Blöcke soll ein Meldung ausgegeben werden.
- Achten Sie auch darauf in der init Funktion die Datenstrukturen richtig zu initialisieren.

b) **Fehlersuche.** *3 Punkte*

Finden Sie die Speicherlöcher im Beispielprogramm! Angabe reicht.

Aufgabe 2 (Binärer Baum in C)

12 Punkte

Implementieren Sie einen binären Suchbaum in C. Ein binärer Suchbaum ist ein binärer Baum, wobei für jeden Knoten des Baums jederzeit gilt, dass die Werte der Knoten des linken Teilbaums kleiner sowie die Werte der Knoten des rechten Teilbaums größer sind als der Wert des Knotens selbst. Implementieren Sie dazu folgende Funktionen:

- `node_t* createTree(int rootValue);`
  Diese Funktion erzeugt einen neuen Wurzelknoten des Baum und weist ihm den Wert `rootValue` zu. Es wird ein Zeiger auf den angelegten Wurzelknoten zurückgegeben.
- `void insert(node_t *tree, int value);`
  Fügt dem Baum mit dem Wurzelknoten `tree` einen Knoten mit dem Wert `value` hinzu. Der Knoten soll so eingefügt werden, dass die Eigenschaften des binären Suchbaums erhalten bleiben.
- `int binary_search(node_t *tree, int value);`
  Sucht nach dem Wert `value` im Baum mit dem Wurzelknoten `tree`. Wird der Wert gefunden, so wird `1` zurückgegeben, wird der Wert nicht gefunden `0`.
- `void cleanUpTree(node_t *tree);`
  Gibt allen dynamisch allozierten Speicher frei, der im Baum mit dem Wurzelknoten `tree` alloziert worden ist.

Testen Sie Ihre Implementierung mit der Datei `main.c`.

Hinweise zur Abgabe:

Das Übungsblatt muss bis zum **31.10.2022, 12:00 Uhr** abgegeben werden.
Halten Sie sich strikt an die Vorgaben im LearnWeb: *siehe hier.* Nichteinhalten der Vorgaben führt automatisch zu Punktabzug!

Die Bearbeitung muss in Gruppen von 3 oder 4 Teilnehmern erfolgen.
Fragen können in der Übung oder im LearnWeb geklärt werden. Abgabe per E-Mail an den jeweiligen Tutor der entsprechenden Übungsgruppe mit Subject „`Abgabe Uebung 1`“. Textaufgaben müssen als PDF-Datei abgegeben werden.

Bei der E-Mail Abgabe bitte nur **eine einzige** .zip oder .tgz oder .tbz Datei abgeben!

Die Abgaben sollen in den entspechenden Vorgabedateien implementiert werden. Die Abgaben müssen sich auf einem Linux-System der IVV5 mit dem bereitgestellten Makefile übersetzen lassen.

Wichtig: Bei der Abgabe in der E-Mail *alle* Namen und Matrikelnummern angeben.
Pro fehlender Angabe (Name oder Matrikelnummer) kann ein Punkt abgezogen werden!