Dr. Andreas Borchert Institut für Angewandte Informationsverarbeitung 30. November 2009
Wolfgang Kaifler / Michael Mattes Blatt 8

Systemnahe Software I
im Wintersemester 2009/2010

Abgabetermin: bis 08. Dezember 2009

20 Fragen zum dynamischen Speicher (6 Punkte)

Schauen Sie sich die Manpage der Funktionen `malloc`, `calloc`, `realloc` und `free` an und versuchen Sie, folgende Fragen zu beantworten:

Gibt es einen Unterschied zwischen `malloc(100)` und `calloc(1,100)`? Wenn ja, welchen?
Führt der Aufruf `free(0);` zu einem Problem?
Gibt es einen Unterschied zwischen `calloc(2,50)` und `calloc(1,100)`? Wenn ja, welchen?
Ist es erlaubt, `free(p)` zweimal hintereinander für den selben Zeiger `p` aufzurufen?
Gibt es einen Unterschied zwischen `realloc(0, 100)` und `malloc(100)`? Wenn ja, welchen?
Welche der vier Funktionen kann die anderen drei ersetzen und wie muss die Funktion jeweils aufgerufen werden?

21 Sortieren mit dynamischem Speicher (18 Punkte)

Der große Vorteil dynamisch allokierten Speichers ist die Möglichkeit, stark unterschiedliche Ein- und Ausgabemengen effektiv zu handhaben. Wenn bisher gefragt war, bis zu 100 Zahlen zu sortieren, konnte man `int zahlen[100];` verwenden. Um dann größere Anzahlen zu handhaben, musste diese Grenze im Code erhöht und das Programm neu übersetzt werden.

Mit dynamischem Speicher kann die Speichermenge ganz einfach zur Laufzeit des Programms erhöht werden. Erweitern Sie daher Ihr Sortierprogramm vom Blatt 4 bzw. 3 so, dass es beliebig viele Zahlen sortieren kann. Auf der Webseite finden Sie Beispieleingaben mit unterschiedlich vielen Zahlen.

Machen Sie sich auch Gedanken darüber, wie viel neuen Speicher Sie jeweils am Stück allokieren – viele Ein-Byte-Anforderungen hintereinander sind nämlich sehr viel aufwändiger als eine große Anforderung.

Hinweis: Wenn Sie zum Sortieren großer Listen einen Algorithmus mit quadratischer Laufzeit verwenden (z.B. Bubblesort), dann wird das Sortieren sehr lange dauern. Da dies jedoch nicht Haupt-Augenmerk der Vorlesung ist, müssen sich in dieser Aufgabe nicht um einen anderen Algorithmus kümmern. Eine korrekte Speicherverwaltung inklusive Speicherfreigabe ist für diese Aufgabe ausreichend.

Viel Erfolg!