Dynamische Datenstrukturen

1 Das Problem mit statischen Datenstrukturen

Wir haben gesehen, dass bei einem Array der Platz fix vorgegeben werden muss. Beim Array handelt es sich auch um eine primitive Datenstruktur. Einen fixen Speicherplatz zu belegen ist vor allem dann ein Problem, wenn wir es mit dynamisch wachsenden und schrumpfenden Datenmengen zu tun haben.

In der Informatik verwendet man deshalb Listen und Bäume, um eine flexible Datenstruktur zu haben.

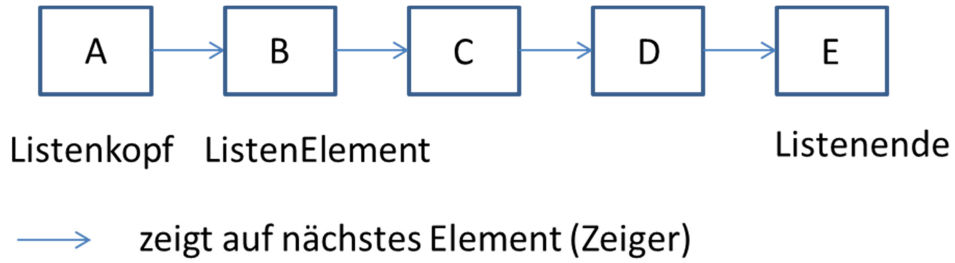
2 Listen für dynamische Datenstrukturen

Listen sind sehr wichtig und werden z.T. bereits als Grundelement der Sprache eingebaut. Programmiersprachen, die Listen oder dynamische Datenstrukturen als Grundelement anbieten, lassen dem Programmierer keinen direkten Zugriff auf die Speicherplatzverwaltung (also Speicherplatz reservieren und wieder freigeben).

Das mag dann etwas eleganter sein fürs Programmieren, ist aber weniger effizient. Deshalb hat das C nicht, weil Effizienz bei C sehr wichtig ist. Somit müssen solche Strukturen selber programmiert werden (es gibt aber genügend Beispiele !)

3 Wie sieht eine solche Liste aus ?

Das Prinzip einer Liste ist wie eine Kette (deshalb eine „verkettete Liste“):



Wir haben 3 Elemente:

Einen **Listenkopf**
Ein (oder mehrere) **Listenelement**(e)
Und einen **Zeiger**

Das letzte Element (Listenende) zeigt einfach auf NULL und ist deshalb wiederum ein Listenelement.

4 Wie suchen wir in der Liste?

Beim Array können wir direkt mittels Index auf ein bestimmtes Element zugreifen:

```
int myNumber = number[7];
```

Das geht leider bei der verketteten Liste nicht. Hier müssen wir die Liste durchgehen und jedes einzelne Element „anschauen“, bis wir das gesuchte Element gefunden haben. Dafür ist das Einfügen und Löschen in einer Liste je nach Fall einfacher, weil wir dann bloss ein Element hinzufügen (ans Ende) oder beim Entfernen eines Elements auf das nächste zeigen.

5 Aufbau einer verketteten Liste in C

Der zentrale Ansatz ist, dass die Listenelemente als **Strukturen** in C definiert werden (deshalb sind Strukturen so wichtig !):

Ebenfalls benützt die verkettete Liste Pointer (**Zeiger**), um die Elemente zu verknüpfen.

Ein Beispiel für verkettete Liste finden Sie im Modul 118 Script, in Kapitel 8. Ebenfalls ist dort nochmals eine Repetition zum Thema Zeiger.

Eine elegante Lösung hat auch Niklas Liechti programmiert.