

**„hello world.“ mit 13 Systemaufrufen?  
... wenn das Mal kein Unglück bring ...**

Die Funktion print in Beispiel 2.5 ist sehr ungeschickt umgesetzt, sie sollte dringend verbessert werden.

**Beispiel 2.9 nutzt die Funktion hello\_world, ganz ohne gegen libhello.so zu linkern.**

Berechne sin (3.0) ganz ohne gegen libm zu linken.

(Tipp in der manual-page findet sich ein (beinahe komplett) passendes Beispiel.)

**In Beispiel 6.1 läuft ganz offensichtlich etwas schief.**

Der Fehler sollte dringend korrigiert werden.

Auch ein gründlicher Test der verbesserten Version kann sicher nicht schaden.

**Sag mir wo der Speicher ist ...**

Beispiel 6.2 enthält ein gewaltiges Speicherleck dieses sollte dringendst gestopft werden.

Auch hier kann ein Test der verbesserten Version gewiss nicht schaden.

**Auch mit den reguläre Ausdrücken aus 7.0 lässt sich experimentieren.**

Mögliche Forschungsziele könnten sein:

- eine einfaches Programm, dass ähnlich wie grep nach vorgegebenen Mustern sucht.
- ein einfaches Programm, dass automatische Textersetzungen vornimmt.
- .... eine ganze Reihe weiterer Spielereien, auf die ich selbst noch nicht gekommen bin

(Tipp: um Kommandozeilen-Argumente nicht alle von Hand parsen zu müssen könnte sich ein Blick in die manual-pages von getopt, bzw. getopt\_long lohnen.)

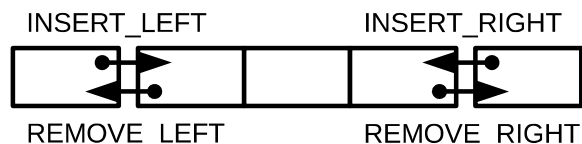
**Nützliche Datenstrukturen**

Nach dem Muster von 7.1, bzw. 7.2 lassen eine ganze Reihe nützlicher Datenstrukturen umsetzen.

Hier einige Beispiele zum ausprobieren:

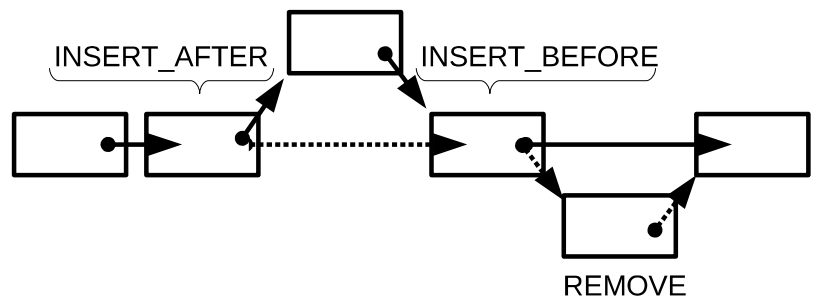
**Dqueue**

In diese Schlanqe kann an beiden Enden etwas hineingestopft und etwas entnommen werden.



**verkettete Liste**

Hier darf an beliebiger Stelle eingefügt und enternt werden.



**doppelt verkettete Liste**

Hier gleich in zwei Versionen gezeichnet.

Eventuell ist es gar nicht Mal soo dumm, wenn sich die Liste in den eigenen Schwanz beißt.

