

Praktikum Programmiersprache C - Binärdateien

Daten können in unterschiedlichster Form in Dateien gespeichert werden. Eine sehr einfache Form ist, die Daten, wie sie in Strukturen im Speicher stehen, direkt in eine Datei zu schreiben. Dies ist unter Aspekten der Portabilität sehr unkomfortabel. Auch dürfen die Strukturen in diesem Fall keine Pointer enthalten, da diese Adressen für die Speicherung wertlos sind. In der Regel wird man zwischen einer internen und einer externen Datenrepräsentation unterscheiden. Dennoch soll sich dieses Praktikum solchen Binärdateien widmen, deren gleichlange Datensätze Strukturinhalte der Struktur tArt (Materialartikel) enthalten.

Im Downloadbereich stehen die Dateien `mat.dat` bzw. `mat.dat.intel` zum Download zur Verfügung. Es sind Binärdateien, die Strukturen, wie sie in `mat.h` beschrieben sind, enthalten. Auf intelbasierten PCs und kompatiblen ist `mat.dat.intel`, bzw. die 64-bit-Version zu wählen, auf IBM-workstation, wie iaix? ist `mat.dat` zu nutzen. Beide Dateien enthalten die selben Daten, aber in plattformabhängiger Form, deshalb die Unterscheidung.

Kontrollfrage: worin unterscheiden sich die einzelnen Dateien?

Schreiben Sie ein C-Programm, das die Datei **mat.dat** als Binärfile einliest, die Artikel sortiert und als Inventurliste mit folgendem Zeilenaufbau ausgibt. Der Speicherplatz, in dem die Artikel abgelegt werden, soll mit malloc auf dem heap ausgefasst werden. Überprüfen Sie den Erfolg der Operation.

Art.-Nr	Bezeichnung	Stueckzahl	Soll	Unterschrift
oooooooooooo	oooooooooooooooooooooooooooo	oooooooooooo

Der Aufbau der Datensätze ist aus dem Headerfile `mat.h`, das Sie mit `#include "mat.h"` in Ihr Programm einbinden können, ersichtlich. Die Dateien `mat.dat` und `mat.h` befinden sich unter

[mat.h](#), [mat.dat](#), [mat.dat.intel](#), [mat.dat.intel64](#)

Der Name der Materialdatei soll als Kommandozeilenparameter an das Programm übergeben werden. Überprüfen Sie an Hand von argc, ob die Parameter korrekt eingegeben wurden.

Die Ermittlung der Länge der Datei ist mit der Befehlsfolge

```
fseek(pFile,0,SEEK_END);
len=ftell(pFile);
fseek(pFile,0,SEEK_SET);
```

möglich.

Zum Einlesen einer gegebenen Anzahl von Bytes verwendet man die Funktion fread.

Zusatzaufgaben:

- Richten Sie einen Vektor von Pointern ein und sortieren Sie die Pointer.
- Geben Sie über einen Kommandozeilenparameter an, ob nach Bezeichnung oder nach Artikelnummer sortiert werden soll.