Argumentverarbeitung und File-I/O

13. Aufgabe *10 Punkte*

Schreiben Sie ein C-Programm analog zum Unix-Befehl *cut*.

Aufrufformen:

mycut -c*charlist* [*filename*]
mycut -d*sep* **-f***colnumbers* [*filename*]

Beispiele:

mycut -c1-3,6,9- sample.txt

> Dateiinhalt: `abcdefghijkl`, Ausgabe: `abc f ijkl`
>
> Mit der Option **-c** kann angegeben werden, die "wievielten" Zeichen einer Zeichenkette heraus gefiltert werden sollen. In diesem Beispiel werden die Zeichen 1 bis 3, das 6.te Zeichen und die Zeichen 9 bis zum Ende der Zeichenkette heraus gefiltert.

mycut -d: **-f**1-3,6,9-

> Eingabetext:`0:1:2:3:4:5:6:7:8:9:10`, Ausgabe: `012 5 8910`
>
> Mit der Option **-d** kann ein Feldtrenner angegeben werden, die darauffolgende Option **-f** spezifiziert die Spaltennummern, die ausgegeben werden sollen. In diesem Beipiel wird der **:** als Trenner verwendet und die Spalten 1 bis 3 sowie die 6.te Spalte und die Spalten 9 bis Ende ausgegeben.

Die Optionen **-c** beziehungsweise **-d** und **-f** wird dem Programm als Argument übergeben. Wird ein *filename* zusätzlich als Argument übergeben, dann soll aus dieser Datei gelesen werden, ansonsten von der Standardeingabe.

Hinweise:

- Implementieren Sie ein *usage*-Funktion für den Fall einer fehlerhaften Argumentübergabe.
- Sämtliche Fehlermeldungen sollen auf *stderr* ausgegeben werden.
- Die Ausgabe erfolgt nach *stdout*.
- Für die notwendigen String-Operationen sei auf *string.h* verwiesen.
- Im Anhang des Skriptes Seite 117 *ff* sind String- und File-I/O-Funktionen beschrieben.
- Vergessen Sie nicht Ihr Programm in Module zu gliedern.