Universität Ulm - Abteilung Angewandte Informationsverarbeitung
Dr. Johannes Mayer und Hans Braxmeier, Wintersemester 2003

12. Übungsblatt zur Vorlesung Allgemeine Informatik III

Abgabetermin: Dienstag, 03.01.2004

Cut *(10 Punkte)*

Unter Unix zählen viele Kommandos zu den Filtern. Ein Filter liest Daten von der Standardeingabe (`stdin`) ein, bearbeitet bzw. „filtert" diese Daten und gibt das Ergebnis auf der Standardausgabe (`stdout`) wieder aus. Fehler werden auf die Fehlerausgabe (`stderr`) geschrieben.

Ein bekanntes Filterkommando zum Bearbeiten von sog. „kleinen Unix-Datenbanken" ist `cut`. Diese Textdateien enthalten pro Zeile einen Datensatz, wobei die Attributwerte durch ein Trennzeichen getrennt sind. Beispiel (Inhalt der Datei `logins`):

```
lob:Bettina Milchglas:3.2.1981
thomas:Thomas Muster:17.11.1977
rubby:Ulrike Huber:13.2.1978
aber:Markus Lüge:1.1.1980
ray:Raimund Zurzel:3.5.1979
```

Mit `cut` können nun einzelne Spalten aus einer solchen Tabelle in Textform gefiltert werden.

Implementieren Sie eine vereinfachte Version von `cut`, mit folgenden Optionen:

- Mit der Option `-d` soll ein Trennzeichen definiert werden können (Fehlt diese Option, wird der Tabulator als Trennzeichen verwendet).
- Die Option `-f` zeigt an, welche Spalten gefiltert werden sollen, wobei mehrere Spalten durch Kommas getrennt werden (ohne Whitespace dazwischen!). Diese Option muss immer angegeben werden. Fehlt sie, so liegt ein Fehler vor. Die Spalten befinden sich in der Ausgabe immer in derselben Reihenfolge wie in der Eingabe – unabhängig von deren Reihenfolge im Parameter der Option `-f`.
- Mit der Option `-o` soll eine Ausgabedatei spezifiziert werden können. Andernfalls erfolgt die Ausgabe auf die Standardausgabe, die – sofern sie nicht (mittels `>`) umgelenkt ist – auf das Terminal geht.
- Jedes weitere Argument wird als Eingabedatei interpretiert, aus der/denen gelesen werden soll. Gibt es kein weiteres Argument, so wird von der Standardeingabe gelesen.

Im Hauptprogramm soll stets von `stdin` gelesen und nach `stdout` geschrieben werden. Alles andere soll über Umlenkungen (von Standardeingabe und Standardausgabe) realisiert werden. Von Standardeingabe können sie nach einer Umlenkung immer noch wie gehabt mit `fgets()`, etc. lesen und auf die Standardausgabe können Sie nach der Umlenkung ebenfalls wie gehabt mit `printf()`, etc. schreiben. Die Argumentverarbeitung soll mit `getopt()` realisiert werden. Siehe hierzu die Beispiele in den Übungen.
Sie dürfen vereinfachend annehmen, dass eine Zeile nie länger als 1000 Zeichen ist.

Beispiele:

`cut -d: -f3 logins`: Extrahiert aus der Datei `logins` die 3. Spalte:

```
3.2.1981
17.11.1977
13.2.1978
1.1.1980
3.5.1979
```

`cut -d: -f3,1,2 logins`: Extrahiert aus der Datei `logins` die Spalten 3, 1 und 2 (aber in der Reihenfolge 1, 2, 3!):

```
lob:Bettina Milchglas:3.2.1981
thomas:Thomas Muster:17.11.1977
rubby:Ulrike Huber:13.2.1978
aber:Markus Lüge:1.1.1980
ray:Raimund Zurzel:3.5.1979
```

Tipp: Einfach mal in die Manpage von `cut` reinschauen und das Kommando testen. Wir implementieren jedoch eine vereinfachte Version!

Viel Erfolg!