Implementierung kleiner Datenbanken unter UNIX I (SS 1999)

Abgabetermin 12. Mai 1999

1 Info über Tabellen (7 Punkte)

Auf Ihrem System haben Sie unterschiedliche ASCII-Tabellen:

- *personen*: eine Tabelle mit den Adressen und Geburtstagen Ihrer Freunde. (Spaltentrenner ist #)
- *passwd*: Die berühmt berüchtigte login-Tabelle. (Spaltentrenner ist :) ...
- Und viele andere Tabellen, die die wildesten Spaltentrennern benutzen.

Um die ganze Sache übersichtlicher zu gestalten beschließen Sie folgendes:
Sei der $Tabellenname = personen$,

- dann enthält die Datei *personen.header* den verwendeten Spaltentrenner und eine Liste der Spaltennamen, und
- die Datei *personen.data* enthält die eigentlichen Daten.

```
personen.header:

Zeile 1:#
Zeile 2:Name#Vorname#Geburtstag



personen.data:

Lehn#Michael#0731/58371
.
.
.
```

Ihre Aufgabe:
CREATE_INDEX ist ein kleines (von Ihnen geschriebenes) Shell-Skript, das Informationen zu Headerdateien in einem bestimmten Format erzeugt.
Hier erst einmal eine Testausgabe:

```
thales$ CREATE_INDEX personen.header rechner.header
personen        #
        Name    1
        Vorname 2
        Telefon 3
rechner  :
        Name    1
        IP      2
        Raum    3
```

Die Syntax für *CREATE_INDEX* lautet:

```
CREATE_INDEX tab1.header [tab2.header ...]
```

Und das Ausgabeformat sei wie folgt festgelegt:

```
<Tabellenname><TAB><Trenner>
<TAB><Spaltenname><TAB><Spaltennummer>
<TAB><Spaltenname><TAB><Spaltennummer>
.
.
.
<Tabellenname><TAB><Trenner>
<TAB><Spaltenname><TAB><Spaltennummer>
.
.
.
```

2 Ein "cut-wrapper" (13 Punkte)

In der Vorlesung haben Sie den *cut*-Befehl kennengelernt. Will man eine Liste aller Namen aus der *passwd*, kann man Ihn wie folgt verwenden:

```
ypcat passwd | cut -d ":" -f 5
```

Was einen daran stört ist, daß man sowohl den Spaltentrenner, als auch die Spaltennummer kennen muß. Aber mit dem tollen *CREATE_INDEX*- Skript aus Aufgabe 1, kann man ein zweites Skript *SELECT* schreiben, mit der die ganze Sache so abläuft:

```
Die Dateien:

passwd.header:
:
login: ... : ... : ... :Name: . . .

passwd.data:
kraus::13503:230:Stefan Kraus:/home/turing/kraus:/bin/sh
kalic::13935:230:Miralem Kaljic:/home/turing/kalic:/bin/sh
exone::13637:230:Jens Nitschke:/home/turing/exone:/bin/sh
.
.
.

Und deren Bearbeitung via "SELECT":
```

```
thales$ SELECT passwd Name
Stefan Kraus
Miralem Kaljic
Jens Nitschke
.
.
.
```

Die SYNTAX:
Als erstes Argument erwartet *SELECT* den Tabellennamen, danach eine Liste der gewünschten Spalten.
Hinweis:
Nennen Sie ihr Skript nicht *select* (kleingeschrieben), denn das ist ein Shell-Buildin und könnte ein Fehler sein, der tage kostet.