Prof. Dr. Franz Schweiggert Abteilung Angewandte Informationsverarbeitung 22. Mai 2003
Johannes Mayer Blatt 3

Software Engineering Praxis (SS 2003)

Abgabetermin: 5. Juni 2003

1 Gibt's denn das? (10 Punkte)

In der Vorlesung haben Sie ja schon etwas über referentielle Integrität gehört. Das bedeutet, dass jeder Fremdschlüssel auch in der referenzierten Tabelle vorkommt und dort Primärschlüssel ist. Zum Beispiel bei den beiden Tabellen *Course* und *Studies* ist die „courseid“ bei Studies ein Fremdschlüssel und referenziert den zugehörigen Datensatz in Course, den es deshalb geben muss. Schreiben Sie ein Perl-Skript, das für diese beiden kleinen Unix-Datenbanken überprüft, ob die referentielle Integrität nicht verletzt ist, d. h. zu jeder courseid in *Studies* ein entsprechender Datensatz in *Course* existiert. Dieses Skript soll melden, für welche courseid's diese Integrität verletzt ist.
Beispiel:

```
thales$ refint.pl
course ID ``informatik'' not found!
course ID ``matheba'' not found!
course ID ``wima1'' not found!
thales$
```

Beachten Sie, dass die beiden Tabellen in der ersten Zeile jeweils die Spaltennamen enthalten!

2 Lustige Übersetzungen (15 Punkte)

Schreiben Sie ein kleines Übersetzerprogramm, das eine Web-Seite von der (als einziges Kommandozeilen-Argument) angegebenen URL einliest und diese Wort für Wort vom Englischen ins Deutsche übersetzt. Auf dem Netz gibt's ein *Englisch-Deutsch-Wörterbuch*. Dieses sollen Sie unverändert in Ihrem Programm verwenden. *Nehmen Sie nur solche Wörterbuch-Einträge, die aus einem Wort bestehen!* Ihr Programm soll im Stande sein, HTML-Seiten zu übersetzen. Dabei darf natürlich alles in Kommentaren (also zwischen `<!--` und `-->`) und in Tags (zwischen `<` und `>`, wobei kein Kommentar) nicht übersetzt werden.
Tipp: `LWP::Simple::get($url)` „zieht“ die Datei von der angegebenen URL und liefert ihren Inhalt als String zurück. Der Modifikator `?` am Ende eines Quantifikators (z.B. `.*?`) bewirkt bei regulären Ausdrücken ein Matching minimaler Länge.

Beispiel:

```
thales$ translate.pl
Usage: translate.pl <URL>
thales$ translate.pl http://www.june29.com/IDP/IDPfiles.html
<html><head><title>The Internet Dictionary Project: Download the
Dictionary Files</title></head>
<body bgcolor="#FFFFFF">
<img src="graphics/IDP1.gif" alt="The Internet Dictionary Project"><br>
<img src="graphics/IDPfiles.gif" alt="Download the Dictionary Files"><p>

The files welches sind produced mit die IDP und die vielerlei volunteers
wer haben visited dieser
pages sind erreichbar hier zu anyone zu unterladen.  Please read die
[...]
thales$
```

3 Attacke! (15 Punkte)

Ständig werden die Rechner des Mathe-Netzes von überall auf der Welt angegriffen. Manchmal sogar extrem heftig von einzelnen IP-Adressen. Damit wir nicht mühsam uns durch die Log-Dateien quälen müssen, gibt es ein Perl-Skript, dass die Log-Dateien parst und einen Bericht über die Angriffe erzeugt. Eine Zeile aus der *Log-Datei* sieht vereinfacht wie folgt aus:

```
123.35.54.7:1253 -> 134.60.166.1:21
```

(Von Angreifer:Port Verbindung zu Opfer:Port.)
Schreiben Sie nun ein Perl-Skript, das alle Angreifer sortiert nach der Anzahl der Angriffe auflistet und jeweils noch die angegriffenen IP-Adressen und die Anzahl der Angriffe mit angibt.
Beispiel:

```
thales$ attack.pl
126.34.51.43
        attacks: 6
        targets: 134.60.66.77
123.35.54.7
        attacks: 3
        targets: 134.60.22.1, 134.60.166.1
134.60.66.5
        attacks: 1
        targets: 134.60.166.1
thales$
```

Tipp: Bei sort können Sie eine eigene Vergleichs-Prozedur angeben.

Viel Erfolg!