Dr. Johannes Mayer
Axel Blumenstock
Christian Ehrhardt
Abteilung Angewandte Informationsverarbeitung
23. Januar 2006
Blatt 12

Allgemeine Informatik III / Systemnahe Software I (WS 2005/2006)

Abgabetermin: 31. Januar 2006

Benford's Law (10 Punkte)

Ausschlaggebend für die Entdeckung von Benford's Law waren Logarithmentafeln, die auf den vorderen Seiten stärker abgegriffen waren, als auf den hinteren Seiten – so, als hätten sich die Benutzer bevorzugt für die niedrigen Zahlen interessiert.
Werfen wir einen Blick in eine Tageszeitung, treffen wir auf Zahlen aller Art: Benzinpreise, Einkommen von Familien, Unfallstatistiken, Hochwasserstände, Einwohnerzahlen von Städten usw. Soll nun die Wahrscheinlichkeit, dass eine beliebig herausgegriffene Zahl mit einer Eins beginnt, geschätzt werden, so erhält man meist 1/9 als Antwort. Die meisten Leute gehen also von einer Gleichverteilung der Anfangsziffern aus (wir wollen von führenden Nullen absehen).
Doch diese Annahme ist falsch. Analysiert man beispielsweise die Anfangsziffern von Bevölkerungsdichten, Atomgewichten, Flusslängen oder ähnlichen Größen[1], stellt man fest: Die Eins als Anfangsziffer liegt immer vorn! In etwa 1/3 aller Fälle fängt die Zahl mit einer Eins an, nur in etwa 1/22 der Fälle mit einer Neun!
Wir wollen nun Benford's Law anhand der Größenverteilung von Dateien überprüfen.

Die Aufgabe

(a) Schreiben Sie hierfür eine Funktion, die ein Verzeichnis und alle seine Unterverzeichnisse rekursiv abläuft. Das Startverzeichnis soll als Kommandozeilenargument dem Programm übergeben werden.

Folgen Sie symbolischen Links nicht, um Endlosschleifen zu vermeiden (konkret: *lstat()* statt *stat()*). Reagieren Sie auch robust auf Fehlschläge beim Aufruf der Bibliotheksfunktionen. Zur Kontrolle können Sie für jede gefundene Datei eine Zeile mit zugehörigen Informationen ausgeben.

[1]Diese Größen müssen freilich unabhängige Realisierungen von (nicht notwendigerweise identisch verteilten) Zufallsvariablen und über mehr als eine Zehnerpotenz verteilt sein.

(b) Erweitern Sie Ihr Programm so, dass die Größe (in Bytes) aller darin enthaltenen regulären Dateien ausgelesen werden und eine Häufigkeitsverteilung über die Anfangsziffern erstellt wird. Ignorieren Sie dabei Dateien mit Größe null. Geben Sie schließlich die Häufigkeitsverteilung in tabellarischer Form aus.

Die Ausgabe könnte wie folgt aussehen:

```
$ ./benford ../..
Ziffer 1:    5102 mal ( 27.81 %)
Ziffer 2:    3322 mal ( 18.11 %)
Ziffer 3:    1992 mal ( 10.86 %)
Ziffer 4:    1775 mal (  9.68 %)
Ziffer 5:    1475 mal (  8.04 %)
Ziffer 6:    1258 mal (  6.86 %)
Ziffer 7:    1318 mal (  7.18 %)
Ziffer 8:    1163 mal (  6.34 %)
Ziffer 9:     941 mal (  5.13 %)
```

Nützliche Bibliotheksfunktionen

- `DIR *opendir(char *pathname)` öffnet einen „directory stream“ für das mit *pathname* angegebene Verzeichnis. Die darin enthaltenen Dateien können danach mit
- `struct dirent *readdir(DIR *dir)` sukzessive gelesen werden.
- `int closedir(DIR *dir)` gibt die durch *dir* belegten Ressourcen wieder frei.
- `int lstat(char *pathname, struct stat *buf)` ermittelt Informationen zur angegebenen Datei und legt sie in buf ab. Danach kann auf die über die Felder in buf zugegriffen werden. Im Gegensatz zu `stat()` folgt `lstat()` symbolischen Links *nicht*.
- `void perror(char *s)` gibt nach dem Fehlschlagen einer Bibliotheksfunktion oder eines Systemaufrufs zunächst *s*, dann einen Doppelpunkt und danach eine zum Fehler passende Meldung auf der Standardfehlerausgabe aus. Beispielsweise:

```
./irgend/ein/verzeichnis: Permission denied
```

Weitere Hintergründe zu Benford's Law, oder wie das Gesetz zur Überprüfung von Steuererklärungen herangezogen werden kann, beschreibt der Artikel „Die Welt vom Planeten Zob“ von Jörg Albrecht:
`http://www.mathematik.uni-karlsruhe.de/seite/presse/de#Zob`

Viel Erfolg!