Übungen zur Stringverarbeitung: Blatt 2

Programmierung in AWK

1. LaTeX-Dokument abschicken
   Sie möchten Ihre brandheißen, unerhört neuen Forschungen in Form eines Artikels bei einem wissenschaftlichen Verlag einreichen. Der Verlag schreibt Ihnen nicht nur das zu verwendende LaTeX-Template vor, sondern auch noch, dass alle Bilder als „fig_1.eps“ etc. bezeichnet werden sollen. Wenn in einem Bild zwei Diagramme nebeneinander stehen, dann sollen sie analog als „subfig_1a.eps“,„subfig_1b.eps“ etc. benannt werden. Ihr Artikel („heisse_forschung.tex“) liegt zusammen mit den Bildern und viel Unrat, der sich beim Schreiben angesammelt hat, in einem Verzeichnis, und Sie müssen jetzt sowohl die Bilder umbenennen als auch das LaTeX-File ändern und alles zusammenzippen. Glücklicherweise hat Ihnen ein ehemaliger Doktorand ein awk-Skript hinterlassen, das genau diese eklige Aufgabe erfüllen soll („lapack_numbered.awk“). Sie probieren es aus:

   ```
   make distrib
   ./lapack_number.awk heisse_forschung.tex >distrib/heisse_forschung.tex
   cp heisse_forschung.bbl distrib
   ls distrib
   zip -r artikel.zip distrib
   ```

   a) Erklären Sie, wie das Programm funktioniert. 3 Punkte
   b) Finden Sie ein Beispiel, bei dem das Programm nicht funktioniert 2 Punkte
   c) Ändern Sie das Programm so ab, dass die Bilder nach dem Schema des Verlags umbenannt werden, und dass man nicht die BiBTeX-Liste von Hand kopieren muss 5 Punkte

   Hinweise

   a) Zu 1b: Überlegen Sie sich, ob ein includegraphics-Befehl wirklich immer ein Grafik einbindet. Falls Sie dazu kein Beispiel finden, sehen Sie sich den gensub-Ausdruck an. Werden wirklich immer alle Bilder gefunden?

   b) Zu 1c: Der Name des aktuell bearbeiteten Files steht in `FILENAME`

2. Textformat-Konvertierung
   Sie verwenden die Rheologie-Software Rheoplus von Anton Paar, um Ihre Proben automatisch durchzumessen. Sie würden die Daten anschließend gerne mit Gnuplot weiterverarbeiten. Die Export-Funktionen von Rheoplus sind äußerst dürftig; es gibt einige interne undokumentierte Binärformate, und dann kann man noch die Daten aus dem Tabellenfenster in eine Textdatei kopieren. („impulsresponseff4fg4.txt“). Leider ist das Format recht eigenwillig; nicht nur, dass deutsche Tausendertrenner ».« und Dezimalkommata »,« eingefügt sind, die die meisten Programme aus dem Takt bringen, es sind auch einige Daten als Prosatext eingefügt. Gnuplot möchte folgendes Datenformat haben:

   - Dezimaltrenner ist ».«, Tausendertrenner gibt es nicht. Exponentialschreibweise ist OK.
   - Kommentarzeilen, die überlesen werden sollen, beginnen mit einem »#«

- Linien, die nicht verbunden werden sollen, werden durch eine Leerzeile getrennt
- Datensätze, die indivduell angesprochen werden können, („plot "blabla.dat" index <*n*>“), werden durch zwei Leerzeilen getrennt. Kommentarzeilen zählen nicht als Leerzeilen

Die Rheologie-Software schreibt aus „Abschnitten“ bestehende „Datenreihen“.

Schreiben Sie ein awk-Skript, welches die Beispieldatei in ein Gnuplot-freundliches Format konvertiert. Dabei soll die Prosa als Kommentar erhalten bleiben. „Datenreihen“ sollen durch zwei Leerzeilen getrennt werden, „Abschnitte“ durch eine Leerzeile. Beispiel-Ausgabe der Musterlösung ist in „gnuplot.dat“ enthalten. 10 Punkte

Hinweis: Die Musterlösung sucht nach Stichwörtern, um zwischen Zahlen und Kommentarzeilen zu unterscheiden. Eine echte Zahlenreihe beginnt z. B. immer nach der Zeile mit „Messpkt“, die Einheitenzeile muss dann noch übersprungen werden. Andere Lösungen sind auch möglich.

3. Assoziative Arrays
Die Datei „„usenet.txt“ enthält 623 Usenet-postings. Die Datei hat folgendes Format: Eine Nachricht beginnt mit dem sog. Header, der aus Zeilen der Form „Feld: Wert“ besteht. Der Header endet mit der ersten Leerzeile. Danach folgt der Text des Postings, abgeschlossen mit einem einzelnen Punkt auf der Zeile. Dies ist der einzige Text, der nicht im Posting erscheinen darf.

   Messen Sie die Kreativität der Poster. Gehen Sie dabei wie folgt vor: In jedem Posting zählen Sie die Anzahl der verschiedenen Wörter. Diese Zahl schlagen Sie dann dem jeweiligen Poster zu (aus dem „From:“-Header). Am Schluss geben Sie alle Poster und die kumulative Anzahl verschiedener Wörter aus. 10 Punkte

   Hinweise

   a) Benutzen Sie ein assoziatives Array, um auf effiziente Weise die verschiedenen Wörter zu zählen. Benutzen Sie ein weiteres assoziatives Array, das Sie mit den Namen der Poster indizieren

   b) Die einfachste Methode, um die Wörter zu erhalten, ist FS auf einen passenden regulären Ausdruck zu setzen, so dass $1...$NF die einzelnen Wörter enthält. (⇒ Satzzeichen beachten). Dabei können allerdings Nullstrings entstehen. Es ist zulässig, FS während des Programms zu verändern.

   c) Die gawk-Sonderfunktion »asort()« gibt die Länge eines Arrays zurück; das ist in der Tat die empfohlene Methode, um die Anzahl der Elemente in einem Array zu bestimmen.

Abgabe bis 10. Mai 2006 per E-Mail an `Christian.Gollwitzer@uni-bayreuth.de`