Übungen Unix-Praktikum Blatt 8

Dr. Carsten Gnörlich (cg@techfak.uni-bielefeld.de) 11.01.2016

Abgabe: Samstag, 15.01.2016 23:59 Uhr

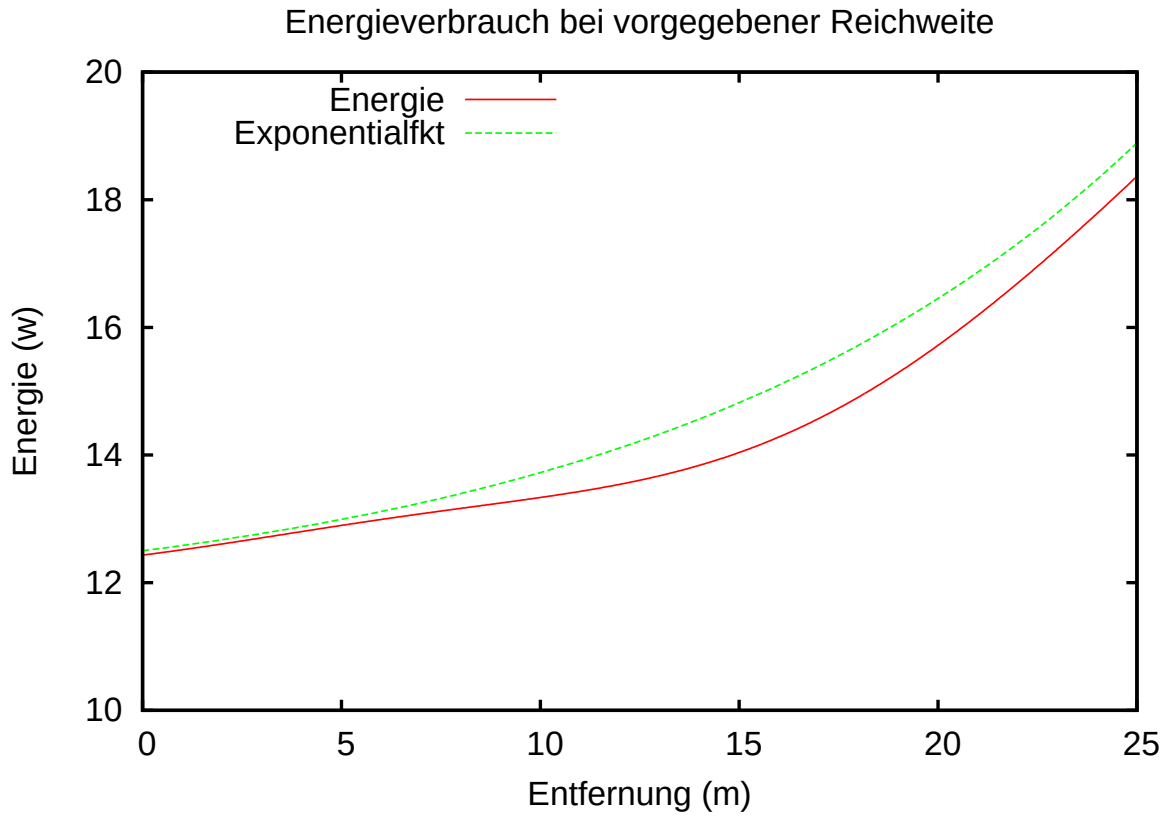
Aufgabe 1 (10 Punkte)

Im heutigen Beispiel- und Übungsarchiv sind einige Datenreihen zu finden:

http://www.techfak.uni-bielefeld.de/~cg/folien/dateien08.tar.gz

Visualisieren Sie die Datenreihe aus der Datei `aufg1.data` wie nachfolgend gezeigt. Beachten Sie dabei die Benennung der Koordinatenachsen, den Titel der Grafik sowie die Verschiebung der Legende in die linke obere Ecke. Überlagern Sie die Datenkurve zum Vergleich mit der Exponentialfunktion.

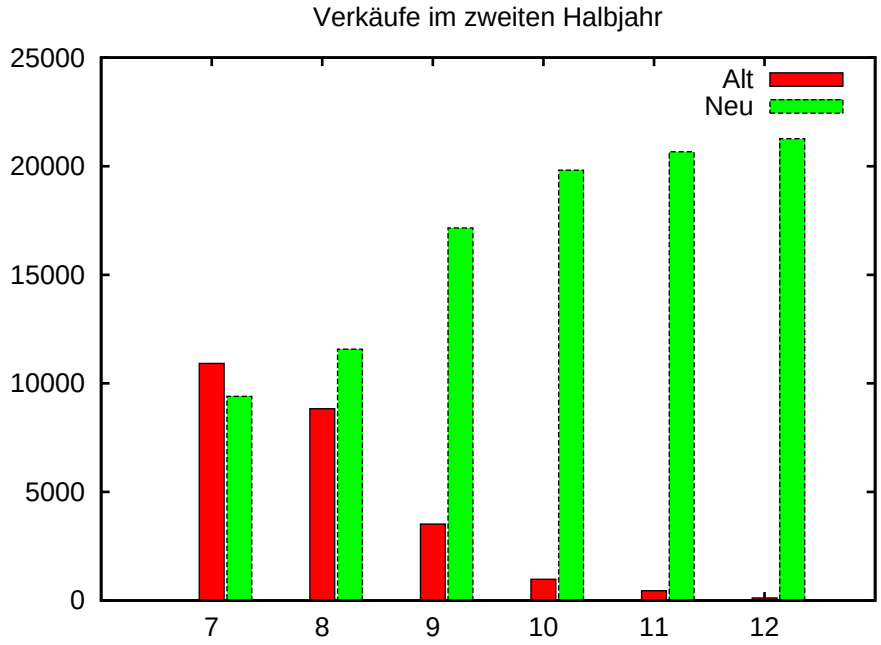
Die Exponentialfunktion hat die folgende Form (in gnuplot-Schreibweise): f(x)=a+exp(x/b). Um die Funktion anschaulich zu überlagern, müssen Sie geeignete Werte für a und b wählen.



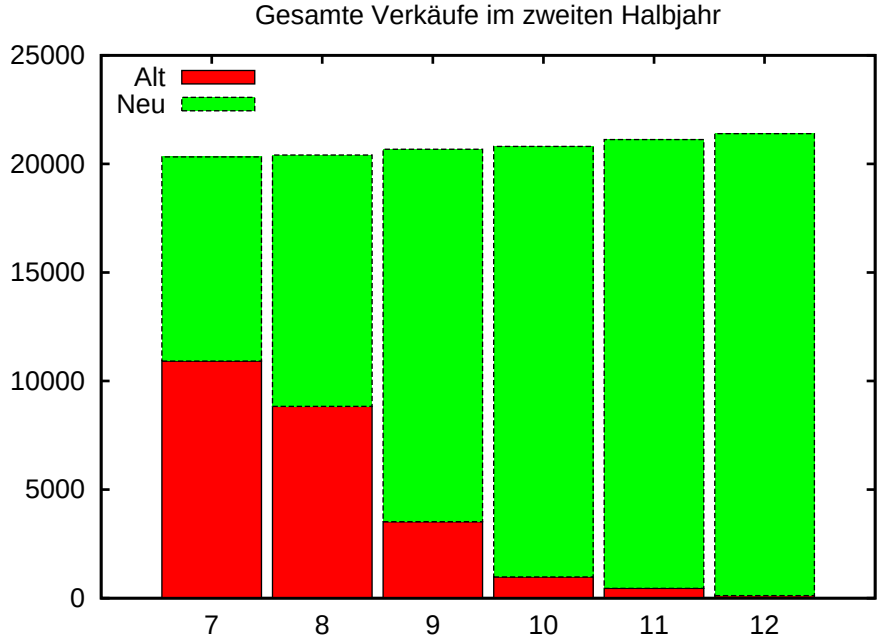
Aufgabe 2 (20 Punkte)

In der Datenreihe `aufg2.data` hat eine Firma die Verkäufe zweier Produkte für das zweite Geschäftshalbjahr gesammelt.

Aufgabe 2a) (10 Punkte)
Geben Sie die Verkäufe mit gnuplot wie nachstehend gezeigt als Balkendiagramm aus:



Aufgabe 2b) (10 Punkte)
Stellen Sie die Verkäufe außerdem kumuliert wie nachfolgend dar, um einen Überblick über die gesamte Geschäftsentwicklung zu bekommen:



Aufgabe 3 (15 Punkte)

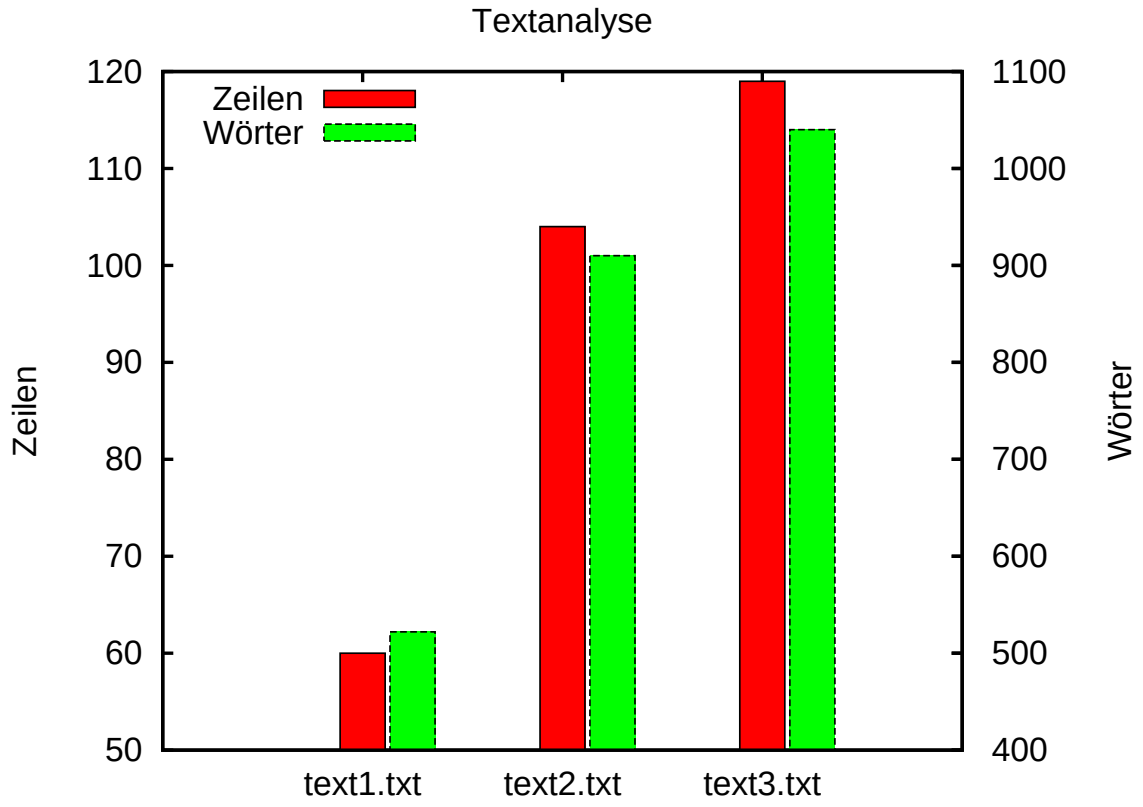
Analysieren Sie den Aufbau der Dateien `text1.txt`, `text2.txt` und `text3.txt`.

Aufgabe 3a) (5 Punkte)

Geben Sie eine Kombination von Kommandozeilenbefehlen an, um die Anzahl der Worte und Zeilen in den drei Textdateien zu ermitteln und in einer Datei `text.data` zu speichern (Hinweis: Erinnern Sie sich an das Programm `wc`!).

Aufgabe 3b) (10 Punkte)

Verwenden Sie gnuplot, um die Anzahl der Worte und Zeilen in den drei Dateien wie nachfolgend zu visualisieren. Beachten Sie die Verwendung von zwei y-Achsen sowie die manuelle Festlegung von deren Wertbereichen. Plazieren Sie die Legende wie in dem Beispiel angegeben und beschriften Sie die x-Achse mit den Namen der Dateien.



Aufgabe 4 (5 Punkte)

Schreiben Sie eine Shell-Funktion `log_entry`, die einen gegebenen Text zusammen mit einem Zeitstempel und den Namen des Rechners ausgibt, auf dem sie aufgerufen wurde.

Die Nutzung und Ausgabe der Shell-Funktion soll wie folgt aussehen:

```
#!/bin/bash

function log_entry()
{
  # Diesen Teil müssen Sie schreiben
}

log_entry "Datenbank neu initialisiert"
sleep 3  # wartet 3 Sekunden
log_entry "Anfrage bearbeitet"

> ./skript.bash
Nov 27 12:03:05 bonnie: Datenbank neu initialisiert
Nov 27 12:03:08 bonnie: Anfrage bearbeitet
```

Der Zeitstempel läßt sich mit Hilfe des `date`-Befehls erzeugen. Das genaue Ausgabeformat des Datums ist über einen Formatstring konfigurierbar; so gibt beispielsweise der folgende Formatstring den Namen und die Nummer des aktuellen Tages aus:

```
> date "+%A der %d."
Donnerstag der 27.
```

Mehr Informationen zu dem Formatstring finden sie in der Manualpage zu `date`.
Der Rechnername steht in der Datei `/etc/hostname`.

Zusatzaufgabe (ohne Bewertung)

Schreiben Sie ein Shellskript, das einen Wurf mit n Würfeln simuliert und das Ergebnis in Form des typischen Augenmusters eines sechseitigen Würfels ausgibt:

```
> ./wuerfel.bash 3
* *  * *    *
* *   *
* *  * *  *
> ./wuerfel.bash 5
* *         *  * *  * *
      *    *   * *
* *       *    * *  * *
```

Lösungshinweise:

Es bietet sich an, die Simulation und Ausgabe eines Würfels in Form einer Shell-Funktion zu realisieren und diese dann wiederholt in einer Schleife aufzurufen.

Das Hauptproblem bei dieser Aufgabe besteht darin, eine mehrzeilige Ausgabe sequenziell aufzubauen. Hat man eine Zeile ausgegeben und einen Zeilenvorschub erzeugt, ist es nicht mehr ohne Weiteres möglich, den Zeilenvorschub rückgängig zu machen und weitere Ausgaben in der betreffenden Zeile zu tätigen.

Als Hilfskonstruktion kann man den Weg wählen, den Inhalt für die auszugebenden Zeilen zunächst in Variablen zu speichern. Dabei nutzt man aus, daß sich an die bestehenden Inhalte von Variablen problemlos weitere Texte anfügen lassen. Erst wenn man alle Ausgaben in den Variablen gesammelt hat, gibt man den Inhalt der Variablen aus.

Da die Würfel in der gezeigten Darstellung drei Textzeilen umfassen, kann man die Ausgabe in drei Variablen a, b und c sammeln. Deren Aufbau und Ausgabe erfolgt nach dem folgenden Schema (das man durch direkte Eingabe ausprobieren kann):

```
> a="* *"
> b=" * "
> c="* *"
```

Ein “Verlängern” der Ausgabe ist nun einfach:

```
> a="${a}  *"
> b="${b} * "
> c="${c}*  "
```

Sowohl in den Zuweisungen als auch bei der Ausgabe kann auf die Anführungszeichen nicht verzichtet werden, da die Kommandozeile sonst Leerzeichen aus der Ausgabe entfernt:

```
> echo -e "$a\n$b\n$c\n"
* *  *
 *  *
* **
```

Downloads (Folien, Übungsblätter)

http://www.techfak.uni-bielefeld.de/∼cg/lehre-unix.html

Hinweis zur Abgabe

Für Studierende im Studienmodell 2011 - damit auch für Euch als Erstsemester im Wintersemester 2014/2015 - ist *keine Abgabe und Korrektur* der Übungen vorgesehen. Bitte bearbeitet die Aufgaben zur Selbstkontrolle. Natürlich könnt Ihr gerne Euren Tutoren Fragen zur Aufgabe stellen und um Lösungshinweise bitten. Es werden allerdings keine kompletten Musterlösungen zur Verfügung gestellt.

Bitte beachtet auch, daß keine Leistungspunkte für die Bearbeitung der Lösungen erworben werden können, auch nicht als individuelle Ergänzung.

Für Studierende im Studienmodell 2002 müssen in Einzelfällen noch Lösungen abgegeben werden, um Leistungspunkte zu bekommen. Bitte nehmt in diesem Fall Kontakt mit Euren Tutoren auf und gebt die Lösungen bitte per E-Mail an Euren Tutor ab. Shellskripte müssen als *.bash*-Datei im Textformat als Anhang abgegeben werden; dies erleichtert den Tutoren die Korrektur erheblich.

E-Mail-Adressen

Eure Tutoren haben die folgenden E-Mail-Adressen
(zuzüglich des `@techfak.uni-bielefeld.de` natürlich):

Markus Flachmann	mflachmann
Patric Steckstor	psteckstor
Alexander Stiebing	astiebing