Prof. Dr. Uwe Schmidt

ptlwedel
PRIVATE BERUFSFACHSCHULE

25. Januar 2011

Aufgaben zur Klausur **Unix** im WS 2010/11 (IA 351)

Zeit: 75 Minuten erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg!

Diese Klausur besteht einschließlich dieses Deckblattes aus 5 Seiten.

Aufgabe 1:

Was ist ein *link* im UNIX Filesystem?

. . . . . .

. . . . . .

Wie erzeugt man einen *link*?

. . . . . .

. . . . . .

Wie löscht man einen *link*?

. . . . . .

. . . . . .

Wozu braucht man *links*?

. . . . . .

. . . . . .

Was ist ein symbolischer *link*?

. . . . . .

. . . . . .

Wozu braucht man symbolische *links*?

. . . . . .

. . . . . .

Aufgabe 2:

Gegeben sei eine Datei mit Wörtern für ein Wörterbuch. Die Datei sei so organinsiert, dass auf jeder Zeile genau ein Wort steht. Alle Wörter bestehen dabei nur aus den 26 Kleinbuchstaben des ASCII Zeichensatzes. Man findet solche Dateien zum Beispiel auf manchen Unix–Systemen unter `/usr/share/dict/words`.

Geben sie in den folgenden Aufgaben einen regulären Ausdruck an, mit dem man die jeweils gesuchten Wörter mit dem `egrep` Kommando selektieren kann. `egrep` verarbeitet den vollen Umfang der Syntax für reguläre Ausdrücke, `grep` akzeptiert auf manchen Systemen nur eine eingeschränkte Syntax. Beachten Sie bitte, dass `egrep` im Standardverhalten nur nach Teilzeichenreihen sucht und alle Zeilen mit einer gefundenen Teilzeichenreihe ausgibt.

Nutzen Sie nur die in der Vorlesung verwendete Syntax für reguläre Ausdrücke, und keine Erweiterungen, die dort nicht behandelt worden sind.

1. Gesucht sind alle Wörter, die mit `a` beginnen und mit `z` enden.

   .....

2. Gesucht sind alle Wörter, die mindestens zwei mal das Zeichen `z` enthalten.

   .....

3. Gesucht sind alle Wörter mit drei bis fünf Zeichen

   .....

4. Gesucht sind alle Wörter, die alle Vokale enthalten, und zwar in alphabetischer Reihenfolge. Nach einem `a` muss also irgendwann ein `e` folgen, nach einem `e` irgendwann ein `i` usw.

   .....

5. Gesucht sind alle Wörter, die genau zwei mal das Zeichen `y` enthalten.

   .....

6. Gesucht sind alle Wörter, die ein Doppel-`o` oder ein Doppel-`a` enthalten

   .....

7. Gesucht sind alle Wörter, die als Vokale genau ein `a` und ein `o` enthalten, wobei das `a` als erstes im Wort vorkommen soll.

   .....

8. Gesucht sind alle Wörter, die mindestens ein `a` und ein `o` enthalten.

   .....

Aufgabe 3:

Beschreiben Sie, was das folgende Kommando berechnet:

find ./public_html −name '∗.html' −print | xargs grep 'UNIX'

. . . . . .

. . . . . .

. . . . . .

Erweitern Sie das Kommando so, dass Fehlermeldungen nicht mehr auf der Konsole sichtbar sondern ignoriert werden, die Ausgabe in eine Datei `find.out` geschrieben wird und während der Programmausführung weiter gearbeitet werden kann.

. . . . . .

. . . . . .

Modifizieren Sie das Kommando so, dass gezählt wird, wieviele HTML-Dateien gefunden werden.

. . . . . .

. . . . . .

Modifizieren Sie das Kommando so, dass gezählt wird, wie häufig das Wort *Microsoft* in allen HTML-Dateien im Home-Verzeichnisbaum vorkommt.

. . . . . .

. . . . . .

Entwickeln Sie ein Kommando mit der gleichen Funktionalität wie das letzte Kommando, das aber das xargs-Programm nicht verwendet.

. . . . . .

. . . . . .

Aufgabe 4:

Beschreiben Sie, was und in welcher Reihenfolge intern in einer Shell und dem UNIX–Betriebsystemkern an Prozess– und Dateioperationen abläuft (Prozesserzeugung, Prozessbeendigung, Ein– und Ausgabe–Umlenkung) wenn folgendes Kommando ausgeführt wird (NICHT: Welche Funktionalität ist in diesem Kommando enthalten).

cat $(find . −name "*.html" −print) | grep −i "mailto:" | wc −l

Beschreibung und/oder Skizze: