Prof. Dr. Uwe Schmidt

ptlwedel
PRIVATE BERUFSFACHSCHULE

16. August 2013

Aufgaben zur Klausur **Unix** im SS 2013 (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.

Nutzen Sie die Rückseiten der Klausur zur Entwicklung der Lösungen und übertragen die fertigen Lösungen in das Aufgabenblatt.

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:

Gegeben sei folgendes bash-Skript:

```
#!/bin/bash

echo '#!/bin/bash'

for i in $*
do
    echo "echo $i 1>&2"
    echo "cat >$i <<'Ende von $i'"
    cat $i
    echo "Ende von $i"
done
```

Verfeinern Sie dieses Skript so, dass es auch mit Dateinamen korrekt arbeitet, die Leerraum und andere Sonderzeichen enthalten.

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Aufgabe 2:

Reguläre Ausdrücke sind ein gutes Werkzeug zur Beschreibung von Textmustern. Für den Aufbau von regulären Ausdrücken gibt es folgende Regeln:

1. jedes Zeichen aus dem Alphabeth (z.B. dem ASCII Zeichensatz) ist ein regulärer Ausdruck.
2. Sonderzeichen, die zum Aufbau der Ausdrücke verwendet werden,
   z.B. `\ [ ] . * + ( ) | ?`, müssen mit einem `\` maskiert werden.
   `\.` steht also für das Zeichen `.`, für nicht druckbare Zeichen gibt es Ersatzsequenzen, z.B. für Zeilenvorschub `\n`, für Tabulator `\t`.
3. `.` steht für ein beliebiges Zeichen
4. `[`$z_1 z_2 z_3$`]` steht für eine Menge von Zeichen z_1, z_2, z_3 , hier dürfen auch Intervalle angegeben werden: `[`z_1`-`z_n`]`.
5. `[^`$z_1 z_2 z_3$`]` steht für die Menge aller Zeichen außer z_1, z_2 und z_3.
6. Wenn r_1 und r_2 reguläre Ausdrücke sind, dann auch $r_1 r_2$ (Folge, Sequenz, r_1 gefolgt von r_2)
7. Wenn r_1 und r_2 reguläre Ausdrücke sind, dann auch r_1 | r_2 (Alternativen, Auswahl, r_1 oder r_2)
8. r_1`*` steht für die 0,1,2,...-fache Wiederholung von r_1
9. r_1`+` steht für die 1,2,...-fache Wiederholung von r_1
10. r_1`?` steht für r_1 oder die leere Zeichenfolge.
11. `(`r_1`)` steht für r_1, Klammern stehen also zum Zusammenfassen von regulären Ausdrücken.

Beispiele:

`[a-zA-Z0-9]`
: steht für die Menge der Buchstaben und Ziffern.

`[^&]`
: steht für die Menge aller Zeichen außer `&`.

`[^0-9]`
: alles außer Ziffern.

`ab*`
: Eine Zeichenfolge, die mit `a` als 1. Zeichen gefolgt von beliebig vielen `b`'s.

`ab+`
: Eine Zeichenfolge, die mit `a` als 1. Zeichen gefolgt von beliebig vielen `b`'s aber mindestens einem `b`.

`(ab)*`
: Eine Zeichenfolge, die abwechselnd aus `a`'s und `b`'s besteht.

`abc|def`
: entweder `abcef` oder `abdef`.

`(abc)|(def)`
: entweder `abc` oder `def`.

`a?`
: ein `a` oder nichts.

Für die folgenden Aufgaben verwenden Sie bitte nur die auf der vorigen Seite beschriebenen Syntaxregeln.

Aufgabe:

Aus Texten sollen Datumsangaben gefiltert werden. Es soll nach Datumsangaben gesucht werden, die aus einer 2-stelligen Tagesangabe bestehen, diese soll durch ein − von der Monatsangabe getrennt sein. Die Monatsangabe ist entweder eine 2-stellige Zahl oder ein 3-buchstabiges Kürzel für den Monat, z.B. `jan, feb, mar, ...`. Nach der Monatsangabe folgt wieder durch ein − getrennt eine 2- oder 4-stellige Jahreszahl.

Geben Sie einen regulären Ausdruck für dieses Datumsformat an:

. . . . . .

. . . . . .

. . . . . .

In Java–Programmen sind Kommentare erlaubt, die durch 2 Schrägstriche eingeleitet werden und bis zum Zeilenende gehen.

Geben Sie einen regulären Ausdruck für diese Kommentarart an:

. . . . . .

. . . . . .

In HTML–Texten wird mit Tags gearbeitet um Textteile zu markieren. Ein HTML-Tag besitzt eine Elementnamen. Dieser besteht aus einer Folge von Buchstaben und Ziffern, außerdem sind − und **:** innerhalb von Namen erlaubt. Das erste Zeichen muß ein Buchstabe sein.

Beispiele

korrekt: `H1 HTTP-EQUIV BIG`
falsch: `ABC- 1ABC X--Y`

Geben Sie einen regulären Ausdruck für diese Elementnamen an:

. . . . . .

. . . . . .

. . . . . .

Aufgabe 3:

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: