

Aufgaben zur Klausur **Unix** im WS 2004/05 (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 7 Seiten

Aufgabe 1:

Welche der folgenden Aussagen treffen zu?

1. *rcs* ist ein UNIX-Kommando zum Kopieren, Löschen und Sichern von Dateien (remove,copy,save). ja nein weiß nicht
2. Eine effiziente Versionsverwaltung legt für jede Modifikation einer Datei eine vollständige Kopie in einem Archiv an. ja nein weiß nicht
3. In einem *RCS*-Archiv werden mehrere Quelldateien aufbewahrt. ja nein weiß nicht
4. *CVS* ist eine Erweiterung von *RCS* ja nein weiß nicht
5. Versionsverwaltungen, die mit Rückwärts-Deltas arbeiten, sind effizienter als Versionsverwaltungen mit Vorwärts-Deltas. ja nein weiß nicht
6. Versionsverwaltungssysteme werden bei Softwarefirmen nur sporadisch eingesetzt, da ihre Anschaffung kostspielig ist. ja nein weiß nicht
7. Versionsverwaltungssysteme haben den Nachteil, daß dadurch die *Makefiles* unhandlich werden. ja nein weiß nicht
8. In einem gut gepflegten *CVS*-Archiv werden nicht nur die Quellen eines Projektes abgelegt, sondern auch die daraus gefertigten Binärdateien. ja nein weiß nicht
9. Versionsverwaltungssysteme werden bei Softwarefirmen nur sporadisch eingesetzt, da sie zu viel Speicher für das Archivieren verbrauchen. ja nein weiß nicht
10. Versionsverwaltungssysteme werden bei Softwarefirmen nur sporadisch eingesetzt, da sie die Arbeit im Team behindern. ja nein weiß nicht
11. Versionsverwaltungssysteme werden bei Softwarefirmen nur sporadisch eingesetzt, da nur Programmquellen verwaltet werden können, aber keine Entwurfsdokumente und Testfälle ja nein weiß nicht
12. In einem *CVS*-Archiv werden ganze Verzeichnisbäume abgespeichert. ja nein weiß nicht
13. Es ist möglich, die Versionen der Quellen aus einem ausgelieferten Programm in Binärformat zu erkennen. ja nein weiß nicht

Aufgabe 2:

Die folgenden fünf Tcl Variablen sind gegeben. Sie sind durch die Anweisungen

```
set 11 {a1 a2 a3}
set 12 {b1 b2}
set 13 c
set 14 {}
set 1 [list $11 $12 $13 $14]
```

erzeugt und initialisiert worden. Welche Ausgaben erzeugen die folgenden Anweisungen

```
puts $1
```

.....

```
puts $14
```

.....

```
puts [list [list $14]]
```

.....

```
puts [list {list $14}]
```

.....

```
puts [lindex $1 1]
```

.....

```
puts [lindex $1 [llength $1]]
```

.....

```
puts [lindex [lindex $1 0] 2]
```

.....

```
puts [linsert $l2 2 x y]
```

.....

```
puts [linsert $l3 end end]
```

.....

```
puts [lreplace $l1 1 1]
```

.....

```
puts [lreplace $l1 1 end b2 c3]
```

.....

```
puts [concat $l1 $l2]
```

.....

```
puts [concat $l4 $l3 $l4]
```

.....

```
puts [lsearch $l b1]
```

.....

```
puts [lsearch $l2 b1]
```

.....



Aufgabe 3:

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 Alphabet (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. `[z1z2z3]` steht für eine Menge von Zeichen z_1, z_2, z_3 , hier dürfen auch Intervalle angegeben werden: `[z1-zn]`.
5. `[^z1z2z3]` 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_1r_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.

Aufgabe:

Mit einer über eine Pipe verbundenen Kommandosequenz der folgenden Gestalt

```
ls -l <datei...> | egrep <muster>
```

werden Texte des Formates erzeugt und gefiltert

```
-rw-r--r--  1 root root      15785 2002-10-04 22:08 a2ps.cfg
drwxr-xr-x  4 root root         104 2004-12-09 18:10 acpi
-rw-r--r--  1 root root      2579 2004-10-02 02:54 aliases
drwxr-xr-x  2 root root         48 2004-10-04 17:24 aliases.d
-rw-r--r--  1 root root     12288 2005-01-03 16:23 aliases.db
crw-----  1 theo audio    14,   4 2004-10-02 10:38 audio
brw-r-----  1 root disk    29,   0 2004-10-02 10:38 aztcd
brw-r-----  1 root disk    29,   0 2004-10-02 10:38 aztcd0
prw-----  1 root root         0 2005-01-30 10:53 initctl
brw-r-----  1 root disk    30,   0 2004-10-02 10:38 lmscd
srw-rw-rw-  1 root root         0 2005-01-31 10:00 log
crw-----  1 root root    10, 140 2004-10-02 10:38 relay8
br-----  1 root root    31,  24 2003-10-02 10:38 rflash0
...
```

Geben Sie für die Suchanfragen auf der nächsten Seite Textmuster für das `egrep`-Kommando an, die aus einer wie oben strukturierten Liste bestimmte Einträge aus dem Dateisystem selektiert.

Alle Verzeichnisse:

.....

Alle Dateisystemeinträge, die `theo` als Besitzer haben:

.....

Alle HTML-Quelldateien (die üblicherweise die Dateierdung `.html` besitzen):

.....

Alle Einträge, die in 2004 oder später das letzte Mal beschrieben worden sind:

.....

Alle Dateien, die zwischen 10:00 und 10:59 beschrieben worden sind:

.....

Alle vollständig schreibgeschützten Einträge:

.....

Alle einfachen Dateien, die leer sind:

.....

Alle einfachen Dateien, auf die es mehr als einen Hardlink gibt:

.....

Alle Einträge, die ein `?` im Namen haben

.....