

Aufgaben zur Klausur **Unix** im WS 2006/07 (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 6 Seiten

Aufgabe 1:

Gegeben seien folgende sechs C-Quellcode-Dateien `Test.c`, `Mod1.c`, `Mod2.c`, `Mod1.h`, `Mod2.h` und `Types.h`. `Test.c` enthält das Hauptprogramm, dieses benutzt Routinen, die in `Mod1.h` deklariert sind und in `Mod1.c` implementiert sind. `Mod1.c` benutzt Routinen, die in `Mod2.h` deklariert sind und in `Mod2.c` implementiert sind. Alle `.c`-Dateien verwenden globale Datendefinitionen aus `Types.h`

Schreiben Sie einen `Makefile` zum Erzeugen eines Programms `Test` aus den oben beschriebenen Dateien. Beachten Sie dabei alle Abhängigkeiten zwischen den Dateien, entwickeln Sie den `Makefile` aber so, daß keine überflüssigen Aktionen gemacht werden.

Nutzen Sie keine im `make`-System vordefinierten oder eingebauten Regeln.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Aufgabe 2:

Geben Sie ein Unix-Kommando an, mit dem eine Tabelle ausgegeben wird, in der für alle '.html'-Dateien im momentanen Arbeitsverzeichnis und in allen Unterverzeichnissen die Anzahl der Zeilen, Wörter und Zeichen angegeben werden.

.....

.....



Aufgabe 3:

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

```
set v1 {}  
set v2 x  
set v3 {y1 y2}  
set v4 {z1 z2 z3}  
set l [list $v4 $v3 $v2 $v1]
```

erzeugt und initialisiert worden. Welche Ausgaben erzeugen die folgenden Anweisungen

```
puts $l
```

.....

```
puts $v1
```

.....

```
puts [list {list $v1}]
```

.....

```
puts [list [list $v1]]
```

.....

```
puts [lindex $l [llength $l]]
```

.....

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

.....

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

.....

```
puts [linsert $v2 end end]
```

.....

```
puts [linsert $v3 2 x y]
```

.....

```
puts [concat $v1 $v2 $v1]
```

.....

```
puts [concat $v4 $v3]
```

.....

```
puts [lsearch $v3 b1]
```

.....

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

.....

```
puts [lreplace $v4 1 1]
```

.....

```
puts [lreplace $v4 1 end b2 c3]
```

.....

Aufgabe 4:

Geben Sie Vorteile von virtueller Adressierung gegenüber realer Adressierung in präzisen Stichworten (keine Allgemeinplätze) an.

1)

2)

3)

4)

5)

