

Klausur

Computergrundlagen WS 2013/2014

JP Dr. Axel Arnold Dr. Olaf Lenz Florian Weik
Georg Rempfer Rudolf Weeber Tobias Richter
Shervin Rafatnia Johannes Zeman Dominic Röhm

4. März 2014

Name	
Vorname	
Matrikelnummer	

Hinweise

- Die Maximalpunktzahl ist 100.
- Der verfügbare freie Platz gibt einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Die Klausur ist sehr umfangreich, um alle Themengebiete abdecken zu können. In der Regel wird es nicht möglich sein, alle Aufgaben vollständig zu bearbeiten. Bearbeite deswegen zuerst die Themengebiete, die Dir besonders liegen!
- Falls der Platz nicht ausreichen sollte, verwende zusätzliche Blätter. Beschrifte diese unbedingt mit Deinem Namen und Deiner Matrikelnummer!
- Einige Fragen ähneln den Fragen aus der Übungsklausur oder vorigen Klausuren, sind aber i.d.R. *nicht* identisch! Lies die Fragen deshalb bitte *genau* durch!

Viel Erfolg!

1 Unixgrundlagen (20 Punkte)

Aufgabe 1: (1 Punkt)

Was ist der Unterschied zwischen dem World Wide Web (WWW) und dem Internet?

Antwort:

- Internet: physikalisches Netzwerk
- WWW: ein Dienst im Internet

Aufgabe 2: (3 Punkte)

Was tut der folgende Befehl?

```
cd testdir && find . -size -50M | wc -l
```

Antwort:

Zählt die Dateien in testdir, die kleiner als 50MB sind.

Aufgabe 3: (2 Punkte)

Das Verzeichnis testdir hat folgenden Inhalt:

```
data.dat gpl-1.0.txt gpl-3.0.txt lgpl.txt  
gnu.url gpl-2.0.txt gpl-3.3.txt
```

Nenne den Shell-Befehl, der aus allen txt-Dateien, deren Name mit „gpl“ anfängt, die Zeilen herausucht, in denen das Wort „license“ vorkommt, und deren Gesamtanzahl ausgibt.

Antwort:

```
grep -c "license" gpl*.txt  
grep "license" gpl*.txt | wc -l
```

Aufgabe 4: (2 Punkte)

Mit welchem Shellbefehl kann man die Datei /bin/Espresso, welche auf dem Rechner cip1.icp.uni-stuttgart.de liegt, auf den lokalen Rechner in das aktuelle Verzeichnis kopieren?

Antwort:

```
scp user@cip1.icp.uni-stuttgart.de:/bin/Espresso .
```

Aufgabe 5:

(6 Punkte)

Markiere und benenne die vier *Syntaxfehler* im folgenden bash-Skript. Was tut das Skript?

```
#!/bin/bash

# set variables
defaultdir=.
maxsize =640x640 # BUG: Leerzeichen vor "="

# command line handling
dir=$1
test -z $dir && dir=$defaultdir

# find images
images=`ls $dir/*.jpg`

if test -z "$images"; do # BUG: "do" statt "then"
    echo "No_images_in_$dir!"
    exit 2
# BUG: "fi" fehlt
# resize images
for image in $images; do: # BUG: ":" falsch
    dirname=`dirname $image`
    basename=`basename $image .jpg`
    newimage=$dirname/${basename}_small.jpg
    echo "$image_-_>_$newimage"
    convert $image -resize $max_size $newimage
done
```

Antwort:

Ändert die Größe aller JPG-Dateien im aktuelle Verzeichnis auf maximal 640x640 und hängt _small an.

Aufgabe 6:

(4 Punkte)

Im CIP-Pool hat Benutzer olenz den folgenden Dialog in der Shell:

```
> groups olenz shervin cgl13-007
olenz : icp klausur sysguru sm assist video cpp12
shervin : icp cgl kidsweek video klausur
cgl13-007 : cgl
> ls cglstuff Oeffentlich
> ls -la
total 0
drwxr-xr-x  4 olenz icp 104 Feb 24 14:35 .
drwxr-xr-x 12 olenz icp 148 Feb 24 14:26 ..
-rwxr--r-x  1 olenz icp   0 Feb 24 14:28 bar.dat
drwxr-xr-x  2 olenz cgl   6 Feb 24 14:28 cglstuff
---xr--rw-  1 olenz icp   0 Feb 24 14:28 foo.txt
-rw-----  1 olenz cgl   0 Feb 24 14:27 musterloesung
drwx----- 2 olenz icp   6 Feb 24 14:29 Oeffentlich
```

Welcher der Benutzer olenz, shervin, cgl13-007 kann welchen der folgenden Befehle erfolgreich ausführen?

```
cat foo.txt
mv musterloesung Oeffentlich
```

Antwort:

cat foo.txt

- olenz: Kein Leserecht (Benutzer hat kein Leserecht)
- shervin: Leserecht über die Gruppe
- cgl13-007: Leserecht über Others

mv musterloesung Oeffentlich

- olenz: Schreibrechte in . und Oeffentlich
- shervin: Keine Schreibrechte in . und Oeffentlich
- cgl13-007: Keine Schreibrechte in . und Oeffentlich

Aufgabe 7:

(2 Punkte)

Das Verzeichnis `Oeffentlich` soll Dateien enthalten, die alle Benutzer im System lesen können sollen. Sind die Zugriffsrechte dafür sinnvoll gesetzt? Wenn nein, mit welchem Befehl können sie richtig gesetzt werden?

Antwort:

```
chmod 755 Oeffentlich
chmod a+rx Oeffentlich
```

2 Python (15 Punkte)

Aufgabe 8:

(6 Punkte)

Das folgende Pythonprogramm hat verschiedene Fehler. Versuche zunächst zu verstehen, was das Programm wohl tun soll. Dann korrigiere die Fehler. Dabei handelt es sich um zwei syntaktische Fehler und zwei logische Fehler.

```
import sys

for arg in sys.argv[1:]:
    x = int(arg)
    a, b = 0, 1
    i == 1 # BUG: "==" statt "="

    while b < x # BUG: ":" fehlt
        a, b = b, a+b
    i += 1 # BUG: Falsche Einrueckung

    if b == x: # BUG: in naechster Zeile: "->" statt "."
        print "{}_ist_Element_{}_der_Folge"->format(x, i)
        continue

print x, "ist_nicht_in_der_Folge"
```

Antwort:

Wenn man auf der Kommandozeile Zahlen angibt, wird überprüft, ob die Zahl in der Fibonacci-Folge auftaucht.

Aufgabe 9:

(1 Punkt)

Die Funktion `sqrt` befindet sich im Package `math`. Schreibe ein Pythonprogramm, das das Paket einbindet und die Wurzel von 9 ausgibt.

Antwort:

```
import math
print math.sqrt(9)
```

Aufgabe 10:

(6 Punkte)

Schreibe eine Python-Funktion `pot(x, n)`, die für ganzzahlige positive `n` den Wert von x^n berechnet, *ohne dabei den Python-Operator `**` zu verwenden*.

Antwort:

```
def pot(x, n):
    result = 1.0
    for i in range(n):
        result *= x
    return result
```

Aufgabe 11:

(2 Punkte)

Was ist in Python der Unterschied zwischen den Zuweisungen `x = 5` und `y = "5"`, und wie wandelt man das Eine in das Andere um?

Antwort:

Die Variable `x` ist als Ganzzahl (**int**) definiert, die Variable `y` als Zeichenkette (**str**). Die Datentypen werden wie folgt ineinander umgewandelt:

```
y = str(x)
x = int(y)
```

3 C (15 Punkte)

Aufgabe 12:

(2 Punkte)

Wofür verwendet man üblicherweise das Tool `gcc`?

Schreibe eine minimale Kommandozeile zur Verwendung von `gcc` mit der Datei `Beispiel.c`. Dabei soll `gcc` eine Datei `Beispiel` erzeugen.

Antwort:

Zum Übersetzen (Kompilieren) von C-Quelltext in maschinenabhängigen Binärcode.

```
gcc -o Beispiel Beispiel.c
```

Aufgabe 13:

(3 Punkte)

Korrigiere die drei (logischen und syntaktischen) Fehler in folgendem C-Programm:

```
#include <stdio.h>
#include <math.h>

double compute_pi(int N) {
    double step = 1.0/N;
    int sum = 0.0; // BUG: sum sollte "double" sein
    for (double x = 0.0; x < 1.0; x += step)
        sum += sqrt(1.0 - x*x);
    return 4.0 * sum / N;
}

int main() {
    int N = 100000000;
    double pi == compute_pi(N); // BUG: "==" statt "="
    printf("%lf\n", pi); //BUG %d statt %lf
}
```

Aufgabe 14:

(1 Punkt)

In der vorherigen Aufgabe lautete die erste Zeile `#include <stdio.h>`. Was passiert in dieser Zeile und weshalb ist sie für das Programm unerlässlich?

Antwort:

Bindet die Standardbibliothek für die Ein- und Ausgabe ein und ermöglicht die Verwendung vom `printf`.

Aufgabe 15:

(2 Punkte)

Vervollständige im folgenden C-Quelltext die Funktion `multipliziere()` so, dass das Array `A` elementweise mit dem Faktor 3 multipliziert wird.

```
#include <stdio.h>

void multipliziere(int *A, int n) {
    for(int i = 0; i<n; ++i)
        A[i] *= 3;
}

int main() {
    int A[10];
    for(int i = 0; i<10; ++i)
        A[i] = i;
    multipliziere(A, 10);
    for(int i = 0; i<10; ++i)
        printf("%d\n", A[i]);
}
```

Aufgabe 16:

(3 Punkte)

Schreibe einen Abschnitt in der Sprache C, der einen neuen Datentyp `Ball` als Verbund definiert, der eine Zeichenkette `Name`, eine ganze Zahl `Nummer` und ein Array aus Fließkommazahlen mit drei Einträgen für die `Position` enthält. Erzeuge eine Variable vom Typ `Ball` und setze den zweiten Eintrag der `Position` auf den Wert `42.0`.

Antwort:

```
typedef struct{
    char* Name;
    int Nummer;
    double Position[3];
} Ball;

Ball b1;
Ball_1.Position[1] = 42.0;
```

Aufgabe 17:

(3 Punkte)

Welche Ausgabe erzeugt das folgende C-Programm?

```
#include <stdio.h>

int main() {
    int A[10];
    /* A mit 0,1,...9 füllen */
    for (int i = 0; i<10; ++i)
        A[i] = i;
    int *B = &A[2];
    B -= 2;
    printf("%i,_%i,_%i\n", A[0], A[1], A[2]);
    A[2] *= 2;
    printf("%i,_%i,_%i\n", B[0], B[1], B[2]);
}
```

Antwort:

```
0, 1, 2
0, 1, 4
```

Aufgabe 18:

(1 Punkt)

Schreibe eine Zeile C-Code, in der eine Variable A als Zeiger auf **double** definiert wird und alloziere dynamisch ein Array der Größe 100.

Antwort:

```
double *A = (double*)malloc(100*sizeof(double));
```

4 Algorithmen und Datenstrukturen (25 Punkte)

Hinweis Im folgenden ist mit *abstrakter Datenstruktur* die Benutzersicht auf eine Datenstruktur gemeint, also etwa eine Liste oder eine assoziative Liste (Wörterbuch). Mit *Datenstruktur* ist die Implementation einer abstrakten Datenstruktur gemeint, also etwa eine Hashmap, eine verkettete Liste oder ein Array.

Aufgabe 19: (5 Punkte)

Beschreibe einen Algorithmus, der möglichst schnell überprüft, ob ein sortiertes Array einen bestimmten Wert enthält.

Antwort:

Bisektion:

1. Wenn das zu betrachtende Array leer ist, dann enthält das Array das Element nicht und der Algorithmus ist fertig
2. Betrachte das Element in der Mitte des Arrays
3. Wenn das Mittelelement das gesuchte Element ist, dann enthält das Array das Element und der Algorithmus ist fertig
4. Ist das gesuchte Element kleiner als das Mittelelement, dann wende den Algorithmus rekursiv auf das Array an, das die Elemente unterhalb enthält
5. Sonst wende den Algorithmus rekursiv auf das Array an, das die Elemente oberhalb enthält

Aufgabe 20: (5 Punkte)

Beschreibe den Mergesort-Algorithmus!

Antwort:

Aufteilen der Liste in zwei Unterlisten, die jeweils (per Mergesort) sortiert werden, falls sie nicht gerade Länge ≤ 1 haben und damit schon sortiert sind. Diese beiden Listen werden daraufhin wie ein Reißverschluss zusammengeführt. Dazu wird an die Ergebnisliste jeweils das kleinere der beiden links stehenden Element der Unterlisten angefügt, bis alle Element angefügt sind.

Aufgabe 21:

(1 Punkt)

Welche Datenstruktur sollte man verwenden, um ein „Wörterbuch“ zu implementieren, so dass unabhängig von der Größe des Wörterbuchs in konstanter Zeit auf die Übersetzung eines beliebigen Wortes zugegriffen werden kann?

Antwort:

| Hashmap

Aufgabe 22:

(6 Punkte)

Schreibe eine C-Funktion, die eine C-Zeichenkette umdreht. Aus „Hallo Welt“ soll also „tleW ollaH“ werden.

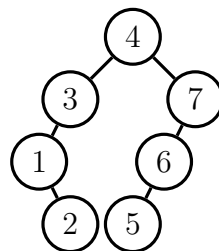
Antwort:

```
void revert(char *word) {
    int a = 0;
    int b = strlen(word)-1;
    while (a < b) {
        char tmp = word[a];
        word[a] = word[b];
        word[b] = tmp;
        a++; b--;
    }
}
```

Aufgabe 23:

(2 Punkte)

Was ist die Ausgabe eines Programms, das den folgenden Binärbaum in symmetrischer Traversierung (linker Teilbaum, Knoten, rechter Teilbaum) ausgibt?



Antwort:

| 1234567

Aufgabe 24:

(6 Punkte)

Schreibe eine C-Funktion, die die Anzahl der Elemente in einem Binärbaum zählt. Ein Knoten des Binärbaumes sei definiert wie folgt:

```
struct node {
    int value;
    struct node *left, *right;
};
```

Antwort:

```
int count_tree(struct node* tree) {
    if (tree != NULL) return 0;
    return count_tree(tree->left) +
           count_tree(tree->right) + 1;
}
```

5 L^AT_EX und HTML (20 Punkte)

Aufgabe 25:

(4 Punkte)

Erkläre die L^AT_EX-Befehle im folgenden Dokument:

```
\documentclass[a4paper]{article}
%Eckstein, Eckstein, alles muss versteckt sein...
\begin{document}
  {\huge\textbf{1.\ "Uberschrift}}\
  {\large{1.1. Unter\ "uberschrift}}\
  ziemlich \textit{einfaches} Dokument
  mit Aufz\ "ahlung:\
  1. eins\
  2. zwei
\end{document}
```

Antwort:

- **\documentclass**: Auswählen der L^AT_EX-Klasse, Festlegung des Seitenformats (0.5P)
- Ein Kommentar, der nur im Code erscheint (0.5P)
- **\begin{document} ... \end{document}** sagt L^AT_EX, dass hier der eigentliche Inhalt des Dokuments liegt. (0.5P)
- **\huge**: Schriftgröße sehr groß (0.5P)
- **\large**: Schriftgröße etwas größer (0.5P)
- **\textbf**: fetter Schriftgrad (0.5P)
- **\textit**: kursiver Schriftgrad (0.5P)
- ****: Zeilenumbruch (0.5P)
- ****: Nächster Buchstabe wird zu Umlaut (0.5P)

Aufgabe 26:

(5 Punkte)

Schreibe den Inhalt (alles zwischen `\begin{document}` und `\end{document}`) des \LaTeX -Dokuments aus der vorigen Aufgabe unter Verwendung sinnvollen *logischen* Markups.

Antwort:

```
\section{Eine \ "Uberschrift} (0.5P)
\subsection{Eine Unter\"uberschrift} (0.5P)
Ich bin ein ziemlich \emph{einfaches} LaTeX-Dokument
mit einer Aufz\"ahlung: (1P)
\begin{enumerate} (1P)
  \item Nummer eins
  \item Nummer zwei
\end{enumerate}
```

Aufgabe 27:

(5 Punkte)

Schreibe ein HTML-Dokument, dessen Inhalt und logische Struktur dem des \LaTeX -Dokuments aus der vorigen und vorvorigen Aufgabe entspricht. Auch hier soll *logisches* Markup verwendet werden! Der Befehl `\documentclass[a4paper]{article}` muss dabei **nicht** übersetzt werden.

Antwort:

```
<html> (1P fuer das Grundgeruest)
<head>
<!--Eckstein , Eckstein ,
alles muss versteckt sein...--> (1P)
</head>
<body>
<h1>1. Eine &Uuml;berschrift</h1> (0.5P)
<h2>1.1. Eine Unter&uuml;berschrift</h2> (0.5P)
Ich bin ein ziemlich <em>einfaches</em> LaTeX-Dokument
mit einer Aufz&auml;hlung: (1P)
<ol> (1P)
<li>Nummer eins</li>
<li>Nummer zwei</li>
</ol>
</body>
</html>
```

Aufgabe 28:

(3 Punkte)

Abbildungen werden in \LaTeX üblicherweise in einer `figure`-Umgebung eingebunden. Schreibe den Teil eines \LaTeX -Dokuments, der die Datei `bild.jpg` in einer solchen Umgebung einbindet. Die Grafik soll horizontal zentriert sein und die Bildunterschrift „Ein Bild“ haben.

Antwort:

```
\begin{figure}
  \centering
  \includegraphics{bild}
  \caption{Ein Bild}
\end{figure}
```

Aufgabe 29:

(1 Punkt)

In \LaTeX gibt es den Befehl `\documentclass`. Welche Möglichkeit gibt es in HTML-Dokumenten, um Formatvorlagen zu beschreiben und verwenden?

Antwort:

Das Einbinden einer CSS-Datei, manuelle Style-Optionen oder Ähnliches.
(Eine Antwort reicht)

Aufgabe 30:

(2 Punkte)

Was sieht man, wenn man folgende HTML-Seite in einem Browser öffnet? Was würde man sehen, wenn man den Code von `<style...>` bis einschließlich `</style>` entfernen würde?

```
<html>
  <head>
    <title>
      Ein schlechter Witz! Hoffentlich liest den keiner.
    </title>
    <style type="text/css">
      b {color: white;}
      em {color: white;}
    </style>
  </head>
  <body bgcolor="white", text="black">
    <p>
      <b>Was ist der Unterschied zwischen einem Storch?</b>
      <br>
      <em>Keiner. Beide Beine sind gleich lang!</em>
    </p>
  </body>
</html>
```

Antwort:

Mit CSS sieht man eine weiße Seite.

Ohne CSS kann man einen großartigen Witz lesen.

6 Visualisierung (5 Punkte)

Aufgabe 31:

(2 Punkte)

Beschreibe, was die folgenden Gnuplot-Kommandos machen:

```
1 set yrange [0:100]
2 set terminal postscript eps enhanced color
3 set output "daten.eps"
4 plot "data.dat" w lp
```

Antwort:

- y-Achse geht von 0 bis 100
- Ausgabe als farbiges EPS-Bild
- Festlegen des Dateinamens
- Plottet die Daten der Datei `data.dat` mit durch Linien verbundenen Punkten.

Aufgabe 32:

(3 Punkte)

Skizziere die Ausgabe des folgenden Pythonprogramms. Die Achsenbeschriftungen kannst Du weglassen!

```
from matplotlib.pyplot import *
from numpy import *

x = linspace(1.0, 10.0, 10)

y1 = x**2
y2 = 5*x

subplot(121)
plot(x, y1, '-')
plot(x, y2, '--')

subplot(122)
loglog(x, y1, '-')
loglog(x, y2, '--')

show()
```

Antwort:

