

Probeklausur

Computergrundlagen WS 2010/2011

JP Dr. Axel Arnold Dr. Olaf Lenz Florian Rühle
Thomas Zauner Shervin Rafatnia Kai Kratzer
Rudolf Weeber

10. Februar 2011

Name	
Vorname	
Matrikelnummer	

Hinweise

- In der Regel gibt der verfügbare freie Platz einen Hinweis darauf, welchen Umfang die Lösung haben sollte.
- Falls der Platz nicht ausreichen sollte, verwende zusätzliche Blätter. Beschrifte diese mit Deinem Namen und Matrikelnummer!
- Die Maximalpunktzahl ist 60.

1 Unixgrundlagen (10 Punkte)

Aufgabe 1:

(1 Punkt)

Was ist der Unterschied zwischen einem Prozess und einem Programm?

Antwort:

- Ein Prozess ist eine Instanz eines Programms.
- Ein Programm gibt es immer maximal einmal, es kann jedoch mehrere Prozesse des selben Programmes geben.

Aufgabe 2:

(1 Punkt)

Zähle 3 Dienste im Internet auf.

Antwort:

WWW, Email, Usenet, FTP, Telnet, SSH, ...

Aufgabe 3:

(1 Punkt)

Worin unterscheiden sich die folgenden beiden Befehle?

```
cat gpl.txt > new.txt
cat gpl.txt >> new.txt
```

Antwort:

Befehl 1 kopiert den Inhalt die Datei `gpl.txt` nach `new.txt` und überschreibt dabei eine eventuell vorhandene Datei `new.txt`. Befehl 2 hängt den Inhalt von `gpl.txt` hinten an die Datei `new.txt` an. Wenn sie noch nicht existiert, wird sie dabei erstellt.

Aufgabe 4:

(1 Punkt)

Mit welchem Shell-Befehl kann man sämtliche Dateien aus dem aktuellen Verzeichnis in das Unterverzeichnis `newdir` kopieren, die die Endung `.txt` haben?

Antwort:

```
cp *.txt newdir/
```

Aufgabe 5:

(1 Punkt)

Was tut der Befehl `man cp`?

Antwort:

Der Befehl ruft die Hilfeseite (Manpage) des Unix-Befehls `cp` auf.

Aufgabe 6:

(1 Punkt)

Was tut der Befehl `grep -i computergrundlagen notes.txt`?

Antwort:

Sucht in der Datei `notes.txt` nach der Zeichenkette `computergrundlagen` und gibt die Zeilen auf der Standardausgabe aus, die sie enthalten. Groß- und Kleinschreibung wird beim Suchen nicht beachtet (`-i`).

Aufgabe 7:

(1 Punkt)

Was tut der Befehl `emacs notes.txt &`?

Antwort:

Startet den Editor `emacs` und öffnet damit die Datei `notes.txt` im aktuellen Verzeichnis. Der Prozess wird dabei im Hintergrund ausgeführt und das Terminal wieder frei gegeben.

Aufgabe 8:

(1 Punkt)

Was tut der Befehl `./myscript`?

Antwort:

Führt (wenn vorhanden und ausführbar) im aktuellen Verzeichnis die Datei `myscript` aus.

Aufgabe 9:

(1 Punkt)

Was tut der Befehl `chown olenz:icp notes.txt`?

Antwort:

Der Befehl ändert den Besitzer der Datei `notes.txt` zu `olenz` und die Gruppe zu `icp`.

Aufgabe 10:

(1 Punkt)

Was tut der Befehl `ssh horst@icp.uni-stuttgart.de`?

Antwort:

Versucht, sich über eine sichere Verbindung als Benutzer `horst` auf dem Rechner `icp.uni-stuttgart.de` einzuloggen. Als nächstes würde wahrscheinlich nach einem Passwort gefragt.

2 Permissions (4 Punkte)

Auf einem Unix-Rechner gibt Benutzer `olenz` die Befehle `groups olenz floh cgl1355` und `ls -la` in einer Shell ein und erhält folgende Ausgabe:

```
$ groups olenz floh cgl1355
olenz  : icp cgl1011 dozent
floh   : icp cgl1011
cgl1355 : cgl1011
bob    : user
$ ls -la
total 4
drwxrwxr-x  2 olenz icp      60 2010-10-27 13:23 .
drwxr-xr-x 22 olenz icp    4096 2010-10-27 13:22 ..
-rw-r--r--  1 olenz cgl1011  0 2010-10-27 13:23 bla.txt
-rwxrw----  1 floh  dozent   0 2010-10-27 13:23 file.sh
-rw-r-----  1 olenz icp      0 2010-10-27 13:22 musterloesungen.txt
```

Aufgabe 11: (1 Punkt)

Welcher der Benutzer `olenz`, `floh`, `cgl1355` und `bob` kann die Datei `musterloesungen.txt` lesen?

Antwort:

`olenz` und `floh`

Aufgabe 12: (1 Punkt)

Welcher der Benutzer `olenz`, `floh`, `cgl1355` und `bob` kann eine neue Datei in diesem Verzeichnis anlegen?

Antwort:

`olenz` und `floh`

Aufgabe 13: (1 Punkt)

Welcher der Benutzer `olenz`, `floh`, `cgl1355` und `bob` kann die Datei `file.sh` ausführen?

Antwort:

Nur `floh`

Aufgabe 14: (1 Punkt)

Welchen Befehl muss `floh` ausführen, um dafür zu sorgen, daß `olenz` die Datei `file.sh` ausführen kann (jedoch nicht `bob` oder `cgl1355`)?

Antwort:

`chmod g+x file.sh`

3 Turingmaschine (4 Punkte)

Eine Turingmaschine ($\Gamma = \{_, 1\}$, $Z = \{A, B, C, D\}$) benutzt die folgende Übergangstabelle:

state	read	write	move	next state
A	-	-	→	A
A	1	1	→	B
B	-	-	←	C
B	1	1	→	B
C	-	-	←	D
C	1	-	←	D
D	-	-	←	STOP
D	1	1	←	D

Das Eingabeband enthält dabei die Zeichenkette „..._111_...“ und der Lese-/Schreibkopf ist auf dem ersten „_“ ganz links positioniert. Der Anfangszustand ist A.

Aufgabe 15:

(3 Punkte)

Simuliere die Turingmaschine! Schreibe dazu die folgende Tabelle fort. Rahme die Position des Lese-/Schreibkopfes ein.

Zustand	Band
A	_ 1 1 1 _
A	_ 1 1 1 _
B	_ 1 1 1 _
B	_ 1 1 1 _
B	_ 1 1 1 _
C	_ 1 1 1 _
D	_ 1 1 _ _
D	_ 1 1 _ _
D	_ 1 1 _ _
STOP	_ 1 1 _ _

Aufgabe 16:

(1 Punkt)

Was ist das Ergebnis der Berechnung? Was tut das Programm? (Hinweis: Zahlen werden bei dieser Turingmaschine im unären Zahlensystem notiert, d.h. eine „3“ wird als „111“ notiert, eine „5“ als „11111“)

Antwort:

Das Ergebnis ist 2 (unär 11). Das Programm subtrahiert eins (1) von der unären Zahl auf dem Band.

4 Python (11 Punkte)

Aufgabe 17:

(2 Punkte)

Was gibt der folgende Python-Befehl aus, und warum?

```
print 3/4
```

Antwort:

Der Befehl gibt 0 (Null) aus. Zähler und Nenner sind ohne Dezimalpunkt, also als Integers definiert. Dementsprechend ist das Ergebnis für Python auch ein Integer. Es wird bei der Rechnung nicht gerundet, sondern abgeschnitten, also wird aus 0.75 eine Null.

Betrachte das folgende Pythonprogramm:

```
number==0
while Number < 100
    if number < 50
        print "n_ist_kleiner_als_50:", number
    else
        print "n_ist_groesser_gleich_50:", number
number+=1
```

Aufgabe 18:

(2 Punkte)

Das Programm enthält ein paar Fehler. Schreibe hier das korrigierte Programm hin.

Antwort:

```
number=0
while number < 100:
    if number < 50:
        print "n_ist_kleiner_als_50:", number
    else:
        print "n_ist_groesser_gleich_50:", number
    number+=1
```

Aufgabe 19:

(1 Punkt)

Was tut das korrigierte Programm der vorigen Aufgabe?

Antwort:

Es gibt die Zahlen 0 bis 99 aus und schreibt jeweils davor, ob die Zahl kleiner oder größer gleich 50 ist.

Aufgabe 20:

(3 Punkte)

Die Fakultät ist wie folgt rekursiv definiert:

$$n! = \begin{cases} 1 & \text{falls } n \leq 1 \\ n(n-1)! & \text{sonst} \end{cases}$$

Schreibe eine Pythonfunktion `fac`, das die Fakultät wie in der Formel rekursiv berechnet.

Hinweis Eine Pythonfunktion wird wie folgt definiert:

```
def fac(n):  
    print(n)
```

Antwort:

```
def fac(n):  
    if (n<=1):  
        return 1  
    else:  
        return n * fac(n-1)
```

Aufgabe 21:

(3 Punkte)

Schreibe eine Pythonfunktion, die die Fakultät iterativ (d. h. mit Hilfe einer Schleife) berechnet.

Antwort:

```
def fac(n):  
    if n<=0:  
        return 1  
    ergebnis = 1  
    for i in range(1, n+1):  
        ergebnis *= i  
    return ergebnis
```

5 Asymptotisches Verhalten (6 Punkte)

Aufgabe 22:

(2 Punkte)

Angenommen, eine Gruppe von n Leuten teilen sich einen Kuchen. Von welcher Ordnung ($\mathcal{O}_{n \rightarrow \infty}(\cdot)$) ist die Größe (Gewicht) des Kuchenstückes pro Person?

Antwort:

Von der Ordnung $\mathcal{O}\left(\frac{1}{n}\right)$.

Aufgabe 23:

(2 Punkte)

Ordne den folgenden Funktionen eine der Ordnungen

$\mathcal{O}_{x \rightarrow \infty}(x)$, $\mathcal{O}_{x \rightarrow \infty}(x^2)$, $\mathcal{O}_{x \rightarrow \infty}(2^x)$, $\mathcal{O}_{x \rightarrow \infty}\left(\frac{1}{x}\right)$ oder $\mathcal{O}_{x \rightarrow \infty}(x \log x)$ zu.

- $f(x) = \frac{1}{1000} + \frac{1}{200} \times 2^x$
- $g(x) = -0.9 + \frac{1}{\log(x+8)} + \frac{1}{2}(x+1)$
- $h(x) = \frac{1}{200}x^2 + \frac{1}{50}x$

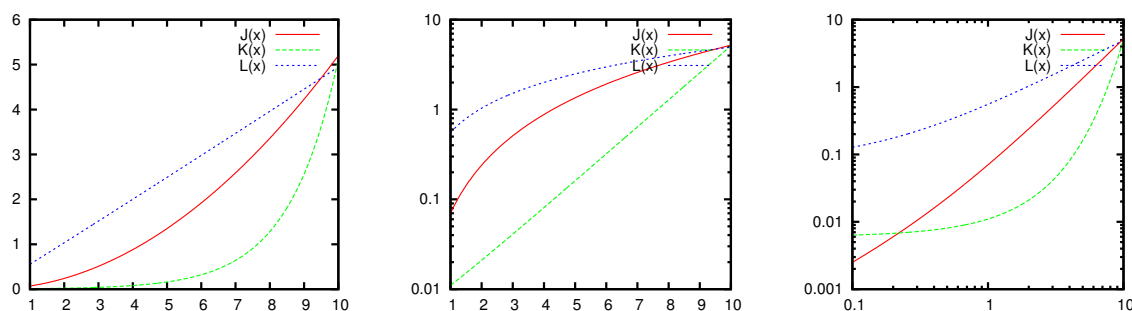
Antwort:

- $f(x) = K(x)$ hat Ordnung $\mathcal{O}(2^x)$
- $g(x) = L(x)$ hat Ordnung $\mathcal{O}(x)$
- $h(x) = J(x)$ hat Ordnung $\mathcal{O}(x^2)$

Aufgabe 24:

(2 Punkte)

In den folgenden Graphen sind die Funktionen aus der vorigen Aufgabe geplottet. Welche der Funktionen entspricht welchem Plot?



Antwort:

Siehe Aufgabe davor.

6 Boole'sche Algebra (5 Punkte)

Gegeben sei der Boole'schen Ausdruck $F = \neg(\neg(a \vee b) \vee (a \wedge \neg b))$.

Aufgabe 25: (1 Punkt)

Stelle für F eine Wertetafel mit jeweils allen Belegungen der Variablen a und b auf.

Antwort:

Aufgabe 26: (1 Punkt)

Welche Aufgabe erfüllt in diesem Zusammenhang das folgende Pythonskript?

```
for a in [True, False]:
    for b in [True, False]:
        F = not (not (a or b) or (a and not b))
        print("a=%s b=%s F=%s" % (a, b, F))
```

Antwort:

Es gibt genau diese Wertetabelle aus.

Aufgabe 27: (3 Punkte)

Vereinfache den Ausdruck F so lange, bis keines der Gesetze der boole'schen Logik mehr anwendbar ist.

Notiere bei jedem Rechenschritt, welches Gesetz verwendet wurde!

Antwort:

$$\begin{aligned}
 & \neg(\neg(a \vee b) \vee (a \wedge \neg b)) \\
 = & \neg\neg(a \vee b) \wedge \neg(a \wedge \neg b) & (20) \\
 = & (a \vee b) \wedge \neg(a \wedge \neg b) & (17) \\
 = & (a \vee b) \wedge (\neg a \vee \neg\neg b) & (21) \\
 = & (a \vee b) \wedge (\neg a \vee b) & (17) \\
 = & (a \wedge \neg a) \vee b & (4) \\
 = & 0 \vee b & (9) \\
 = & b & (11)
 \end{aligned}$$

$$\begin{aligned}
 a \wedge (b \wedge c) &= (a \wedge b) \wedge c & (1) \\
 a \vee (b \vee c) &= (a \vee b) \vee c & (2) \\
 a \wedge (b \vee c) &= (a \wedge b) \vee (a \wedge c) & (3) \\
 a \vee (b \wedge c) &= (a \vee b) \wedge (a \vee c) & (4) \\
 a \wedge b &= b \wedge a & (5) \\
 a \vee b &= b \vee a & (6) \\
 a \wedge (a \vee b) &= a & (7) \\
 a \vee (a \wedge b) &= a & (8) \\
 a \wedge \neg a &= 0 & (9) \\
 a \vee \neg a &= 1 & (10) \\
 a \vee 0 &= a & (11) \\
 a \wedge 1 &= a & (12) \\
 a \vee a &= a & (13) \\
 a \wedge a &= a & (14) \\
 a \vee 1 &= 1 & (15) \\
 a \wedge 0 &= 0 & (16) \\
 \neg\neg a &= a & (17) \\
 \neg 0 &= 1 & (18) \\
 \neg 1 &= 0 & (19) \\
 \neg(a \vee b) &= \neg a \wedge \neg b & (20) \\
 \neg(a \wedge b) &= \neg a \vee \neg b & (21)
 \end{aligned}$$

7 Zahlensysteme (4 Punkte)

Aufgabe 28:

(3 Punkte)

Rechne die Hexadezimalzahl ABBA von Hand (u.U. unter Zuhilfenahme der nebenstehenden Tabelle) in das Binär-, Dezimal-, und Oktalsystem um. Notiere den Lösungsweg!

	2	8	10	16
0	0	0	0	0
1	1	1	1	1
10	2	2	2	2
11	3	3	3	3
100	4	4	4	4
101	5	5	5	5
110	6	6	6	6
111	7	7	7	7
1000	10	8	8	8
1001	11	9	9	9
1010	12	10	A	A
1011	13	11	B	B
1100	14	12	C	C
1101	15	13	D	D
1110	16	14	E	E
1111	17	15	F	F
10000	20	16	10	10

Antwort:

$$\begin{aligned}
 (ABBA)_{16} &= 10 \cdot 16^3 + 11 \cdot 16^2 + 11 \cdot 16^1 + 10 \cdot 16^0 \\
 &= (1010)_2 \cdot 2^{4^3} + (1011)_2 \cdot 2^{4^2} \\
 &\quad + (1011)_2 \cdot 2^{4^1} + (1010)_2 \\
 &= (1010\ 1011\ 1011\ 1010)_2 \\
 (ABBA)_{16} &= 10 \cdot 16^3 + 11 \cdot 16^2 + 11 \cdot 16^1 + 10 \cdot 16^0 \\
 &= 40960 + 2816 + 176 + 10 = (43962)_{10} \\
 (ABBA)_{16} &= (125672)_8 \quad (\text{aus binär mit Tabelle})
 \end{aligned}$$

Aufgabe 29:

(1 Punkt)

In welchem Zahlensystem rechnet ein Computer? Warum?

Antwort:

Im binären System. Computer arbeiten mit binären Schaltungen (eins oder null, an oder aus).

Aufgabe 30:

(1 Punkt)

Welchen Vorteil bietet das Hexadezimalsystem gegenüber dem Dezimalsystem im Computerumfeld?

Antwort:

Innerhalb der Systeme von Zweierpotenzen (2, 8, 16, etc.) lässt sich sehr viel schneller umrechnen, und intern wird binär verwendet.

Aufgabe 31:

(2 Punkte)

Wenn man in Python die Rechnung `600000000000.0*(1.0/600000000000.0)` ausführt, dann erhält man als Ergebnis `0.9999999999999999`. Wieso ist das Ergebnis nicht 1?

Antwort:

Der Computer produziert aufgrund der begrenzten Genauigkeit der gespeicherten Variablen einen Rundungsfehler bei jeder Berechnung. Addieren sich diese Rundungsfehler auf, können Ergebnisse ungenau werden.

8 L^AT_EX(5 Punkte)

Aufgabe 32:

(1 Punkt)

Was ist der Unterschied zwischen visuellem und logischem Markup?

Antwort:

Beim visuellen Markup wird vom Autor für einen bestimmten Textabschnitt das visuelle Aussehen (Fettdruck, kursiv, ...) festgelegt. Beim logischen Markup legt der Autor die Rolle des Abschnittes im Text fest. Wie diese Rolle im Textsatz dargestellt wird, entscheidet der Setzer, der das L^AT_EX-Format festlegt.

Aufgabe 33:

(2 Punkte)

Welcher der folgenden L^AT_EX-Befehle ist visuelles, welcher logisches Markup?

- `\textit{...}` Antwort: visuell
- `\textbf{...}` Antwort: visuell
- `\emph{...}` Antwort: logisch
- `\begin{center} ... \end{center}` Antwort: visuell
- `\begin{itemize} ... \end{itemize}` Antwort: logisch
- `\section{...}` Antwort: logisch

Aufgabe 34:

(2 Punkte)

Die folgende Tabelle wurde mit L^AT_EX gesetzt.

Name	Telefon	Raum
Olaf Lenz	63607	209
Axel Arnold	67609	201

Ergänze das folgende Stück von L^AT_EX-Code so, daß es die Tabelle erzeugen würde.

```
\begin{tabular}{|l|r|r|}
\hline
\textbf{Name} & \textbf{Telefon} & \textbf{Raum} \\
\hline
Olaf Lenz & 63607 & 209 \\
Axel Arnold & 67609 & 201 \\
\hline
\end{tabular}
```

9 Bildbearbeitung (2 Punkte)

Aufgabe 35:

(2 Punkte)

Welches Grafikformat (PNG, JPG oder SVG) eignet sich am besten für die folgenden Grafiktypen, und warum?

- Fotos im WWW
- Mathematische Plots zur Verwendung in einem (wissenschaftlichen) Schriftstück
- einfache Diagramme zur Veröffentlichung auf einer Webseite

Antwort:

Für Fotos im WWW empfiehlt sich JPG, da es gut komprimiert ist für kurze Ladezeiten und für Fotos optimiert. Für mathematische Plots ist SVG als Vektorgrafik am besten geeignet, da beliebig skalierbar. Für einfache Diagramme auf einer Webseite nimmt man am besten PNG, da scharfe Kanten in JPGs schlecht kodiert werden können und Vektorgrafiken noch nicht alle Browser anzeigen können.

10 Gnuplot (2 Punkte)

Aufgabe 36:

(2 Punkte)

Die Datei `histo.dat` enthalte die Daten einer mittels Zufallszahlen erzeugten verrauschten Normalverteilung. Skizziere den vom folgenden Gnuplot-Skript erzeugten Plot.

```
g(x) = A*exp(-B*(x-m)**2)
fit g(x) "histo.dat" via A,B,m
set xrange [-5,5]
plot "histo.dat" with yerrorbars, g(x)
```

Antwort:

11 C (7 Punkte)

Aufgabe 37:

(1 Punkt)

Was tut der Interpreter einer Programmiersprache?

Antwort:

Der Interpreter liest den geschriebenen Code Zeile für Zeile ein und führt die Befehle aus.

Aufgabe 38:

(3 Punkte)

Schreibe die folgende `while`-Schleife in eine `for`-Schleife um, die dasselbe tut

```
int i=0;
while (i < 35)
{
    printf("%d\n", fib(i));
    i++;
}
```

Antwort:

```
for (int i=0; i < 35; i++)
{
    printf("%d\n", fib(i));
}
```

Aufgabe 39:

(3 Punkte)

Welche Ausgabe erzeugt das folgende C-Programm?

```
#include <stdio.h>
int main()
{
    int a[3];
    int *b = &a[2];

    for (int i = 0; i < 3; i++)
        a[i] = i*i;
    printf("%d_%d_%d\n", a[0], a[1], a[2]);
    *b /= 2;
    printf("%d_%d_%d\n", a[0], a[1], a[2]);
}
```

Antwort:

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