Prüfungsprotokoll
Diplom Hauptprüfung Physik
Nichtphysikalisches Wahlpflichtfach Informatik
April 2006
Prof. German

Vorlesungen:
SS 05: Informatik II für Nebenfachstudierende, Dr. Hofmann
WS 05/06: Informatik I für Nebenfachstudierende, Dr. Klehmet

Vorbereitung:
Versuchen den Inhalt des nicht besonders guten (z.T. auch fehlerhaftem) Vorlesungsskript in anderer Literatur zu finden
INFI: Internet, Selfhtml, PHP5 (Dirk Taggesell), Programmieren lernen mit PHP 5 (Jörg Krause)
INFII: Betriebssysteme (Rüdiger Brause) + Internet (manche Skripte sind besser als das Buch!), Multimedia-Technologie (Steinmetz) + Internet (irgendwie war das Skript nicht besonders Nahe am Buch), MySQL/PHP-Datenbankanwendungen (Greenspan)
Ca. 3 Wochen: Exzerpieren des Stoffs (150 Karteikarten), ca. 1 Woche: auswendig lernen

Prüfung:
Prüfer: Prof. German, Beisitzer: Dr. Heindl
Ziemlich genau 45 Minuten
Der Prüfer hatte sich das Repetitorium aus dem Vorlesungsskript der beiden Semester ausgedruckt, und ist das chronologisch durchgegangen. Ich weiß nicht ob er das immer so macht, aber als Tipp sollte man wohl alles das gut können, was stichpunktartig im Repetitorium vermerkt ist.
Prüfungsatmosphäre freundlich.

Note:
1,0 (andere Noten von Kommilitonen in meinem Prüfungszeitraum: 1,3 / 1,0)

Datendarstellung und Rechnerarchitektur

F: Zahlendarstellung, zuerst die ganzen Zahlen: Ganze Dezimalzahl in Dualzahl umrechen?
A: sukzessive Division – Beispiel vorgerechnet:
23:2=11 R 1
11:2 =5 R 1
5:2 = 2 R 1
2:2 = 1 R 0
hier hab ich fälschlicherweise schon 23 = 0111 hingeschrieben, dann gestutzt und gesagt, dass mich die Null am Anfang verwirrt, und festgestellt dass ich ja die 1 noch dividieren muss:
1:2 = 0 R 1
23 = 10111, die letzte 1 (also die erste Zeile in der Rechnung) gibt an, dass die Zahl nicht gerade ist, daher diese Reihenfolge der Ziffern
F: Zur Darstellung der negativen Zahlen gibt es ja die Möglichkeit mit der Komplementdarstellung
A: Bei Dualzahlen gibt es zwei Komplemente:
B-Komplement (hier also 2er Komplement) $Z+\dot{Z}=B^n$
B ist die Basis, n die jeweilige Länge der Zahlen
(B-1)- Komplement (hier also 1er Komplement) $Z+\dot{Z}=B^n-1$
Man nutzt für die Darstellung der negativen Zahlen das 2er-Komplement, da die Zahl und ihr Komplement dann B^n ergeben, was ja eine 1 mit n Nullen ist, und wenn man nur n Stellen zulässt ist das Null. Also rechnen wie mit Zahl und ihrem Negativem.
Komplement einer Zahl, z.B. 1101 -1er-> 0010 -2er-> 0011
F: Gleitkommaarithmetik
A: Zahl in Gleitkommaarithmetik = Mantisse * Basis $^{\text{Exponent}}$

Damit man Vorzeichen bei Exponent einsparen kann, verwendet man Charakteristik, Addition mit positiver Konstante, oft 50, dann bei zwei Stellen Zahlenbereich von -50 bis +49

Mantisse muss in Normalform vorliegen, also 0,1-0,9 so dass man den Nuller und das Komma einsparen kann, man braucht dann nur noch Vorzeichenbit.

F: Wie geht das wenn man zwei Zahlen addiert

A: Beispiel: $0{,}24*10^{-20} + 0{,}99*10^{-19}$

Zuerst positive Charakteristik und Einsparung von „0,", Annahme: drei Stellen für Mantisse:

=> 240|30 + 990|31

Dann Angleichung der Charakteristik:

=> 024|31 + 990|31

Addition:

=> 1014|31

Normierung der Mantisse:

$101|32 = 0{,}101*10^{-18}$

(Die Rechnung hat bei mir nicht so toll geklappt – hab das Prinzip immer richtig erklärt, aber die Zahlen … - hab mit damit nicht aufgehalten, und nur gesagt, dass die Zahlen irgendwie nicht so ganz stimmen, hat niemanden gestört)

F: Welche Fehler können dabei auftreten?

A: Genauigkeitsverlust durch Angleichen der Charakteristik, kann sogar dazu führen, dass wenn man immer nur eins dazu zählt, dass man gar nicht gesamten Zahlenraum erreichen kann.

Weiterer Fehler: Overflow, wenn über Bereich hinaus.

F: Grundaufbau eines Computers?

A: Computer = Zentraleinheit (CPU + Hauptspeicher) und Ein/Ausgabesteuerung.

I/O: Kommunikation mit Ein-, Ausgabegeräten und Massenspeicher und mit Netzsystem zu anderen Computern

F: Was tut der Prozessor?

A: Herz eines Computers, arbeitet Programm ab, hat dazu spezielle Hilfsspeicherzellen, sog. Register in denen er mit Hilfe der Arithemtical Logical Unit elementare Rechen- und logische Operatoren ausführen kann, z.B. Vergleichen, Addieren oder in Arbeitsspeicher schreiben oder von dort lesen.

F: Welche Speicherarten, Register ja schon angesprochen

A: Register in CPU, Haupt/Arbeitsspeicher und externe Speicher (Diskette, Optische Platten, Festplatten)

Hauptspeicher sog. RAM-Speicher mit wahlfreiem Zugriff auf jedes Bit. Zwei Arten:

Dynamic RAM im eigentlichen Arbeitsspeicher, billiger, Kondensatoren brauchen Spannungspulse, zwei Varianten:

Synchroneous DRAM: an Taktfrequenz des Prozessors angepasst

DDR DRAM: Doppelte Datenrate, da auch an abfallender Taktflanke Zugriff

Im Cache-Speicher: Static RAM: teurer aber 10mal schneller, Flip-Flop-Speicherzellen, brauchen ständige Stromversorgung

(Bemerkung des Prüfers, dass Stoff sehr gut verinnerlicht)

HTML

F: Grundgerüst einer HTML-Datei

A: Grundsätzlich Kopf und Körper

Als erstes Dokumenttypdeklaration:

```
<!DOCTYPE HTML PUBLIC
```

dann kryptische Zeichen und Vermerk auf die Dokumenttypdeklaration des W3C

danach eigentlicher Head:

```
<head>
<title> … </title>
<meta name="…" content="…">
</head>
```

Meta-Tag dient zur Angabe von Inhalt, Stichwörtern, Autor…

Title-Tag für den Titel der Seite im Browserfenster

Dann folgt eigentlicher Körper, der das HTML-Dokument enthält:

```
<body>
```

…
</body>
</html>

F: In der Vorlesung Skripting angeschaut, Server-Side und Client-Side, Erklärung?

A: Clientside: Im eigentlichen HTML-Teil, von Benutzer einsehbarer Code (daher ungeeignet für sicherheitsrelevante Daten), Browser muss unterstützen, aber Server nicht
Serverside: Server gibt nur Ergebnis aus, User sieht Code nicht, aber ständige Client-Server-Verbindung nötig. Server muss Sprache unterstützen

PHP

F: PHP als Serverside-Skripting und Beispiel für Programmiersprache. Wie sieht Funktion zur Berechnung der Fakultät aus?

A: Einbettung in HTML

```
<?php
function fak ($a[=voreingestellter Wert]){
$b=0;
for ($i=0; $i<=$a; $i++){
$b=$b*$i;}
return $b;}
```

Klammern hatte ich zuerst teilweise vergessen, und wurde vom Prüfer darauf angesprochen.

F: Und was passiert bei Null

A: Da müsste die Funktion 1 zurückgeben, tut sie nicht, also noch if:

```
If($a==0)
        Return 1;
Else{
        Block von oben}
```

F: Wie ist das mit Übergabe über Wert oder Referenz?

A: Hier Übergabe per Wert, auch Übergabe per Referenz möglich, dann „&" vor die Variable. Wird z.B. verwendet, wenn eine Funktion eine Variable außerhalb verändern soll, aber in der Funktion keine Annahme über den Variablen-Name (denn dann könnte man auch einfach global verwenden).

F: Lexikalische Struktur, was heißt das?

A: Gibt Grundstruktur der Sprache wieder, z.B. (anhand obigem Code), dass Variablen immer mit $ beginnen, das Groß-/Kleinschreibung bei Variablen unterschieden wird, sonst nicht. Anweisungen enden mit ; und können in {} zusammengefasst werden. Es gibt Schlüsselwörter wie for, function, return, if, else

F: welche Flussteueranweisungen?

A: oben schon verwendet: for, for each, if. Gibt noch switch-case:

```
switch (Ausdruck)
case(Wert1):
        Anweisung
        break;
…
```

F: Such- und Sortieralgorithmen, zuerst suchen

A: Drei Suchalgorithmen,
- erstes normale Suche: alle Elemente vergleichen
- Suche in aufsteigend sortierten Folgen, nicht direkt vergleichen, sondern gucken ob größer/gleich, nur dann vergleichen, sonst direkt weiter. Erspart einen Vergleich und bricht Suche ab, wenn an der Stelle angekommen, an der der Wert stehen sollte.
- binäre Suche in aufsteigend sortierten Folgen: nimm immer das mittlere Element (also Anfang + Ende / 2, davon Int) und schau ob größer oder kleiner, dann selbes in der Hälfte davor oder dahinter

Nachfrage beim Unterschied zwischen binärer Suche und suche in aufsteigend sortieren Folgen – Prüfer dachte dass ich das selbe zweimal erklärt habe

F: Komplexitität?

A: erster Fall einfach auszurechnen, selbe Wahrscheinlichkeit Element zu finden $\sum_{i=1}^{n} \frac{1}{n} = \frac{n+1}{2}$

F: und bei den anderen?

A: weiß ich nicht genau, die sind schneller, aber ich denke auch Ordnung n

F: na ja….. was heißt Ordnung n überhaupt?
A: n kommt nicht im Quadrat oder so vor. Könnte bei der binären Suche auch log(n) sein.
F: log(n) ist richtig. Die binäre Suche ist besser. Welche Sortieralgorithmen?
A: erste Gruppe: Sortieren durch direktes Auswählen, direktes Einfügen und Bubblesort
- direktes Auswählen: gehe durch Folge und nimm kleinstes Element und vertausche es mit erstem, also an erster Stelle, dann suche nach nächst kleinerem und mit zweitem Vertauschen usw.
- direktes Auswählen: beginne mit zweitem Element und wenn kleiner als erstes vertauschen, dann Element rechts daneben und jeweils so weit nach links schieben bis an passender Stelle
- Bubblesort: Hinterste zwei Elemente vertauschen, wenn rechtes kleiner als linkes, dann die beiden Elemente davor usw. Irgendwann gerät man an die 1 die damit ganz nach links geschoben wird. Kleine Elemente steigen also „wie Blasen im Wasser" auf. Daher Bubblesort. Danach wieder von hinten anfangen, aber schon bei vorletzten (von vorne) Element aufhören, da ja schon sortiert.

Zweite Gruppe ist Divide-and-Conquer-Verfahren. Dabei wird Problem zerlegt und rekursiv für diese Gruppe gelöst, danach zusammengesetzt. Ein Beispiel ist Quicksort, dabei nimmt man erstes Element und stellt es in die Mitte, alle kleineren links davon, alle größeren rechts. Also schon mal grob sortiert. Das dann rekursiv für die beiden Teilmengen.

(Prüfer scheint ungeduldig zu werden, also keine weitere Erklärung wie man die Kleineren links und die Größeren rechts hinbekommt.)

Betriebssysteme

F: Aufgaben eines Betriebssystems
A: für Benutzer Oberfläche, Abstraktion von Hardware um einheitliche Oberfläche für Programme zu bilden, Verwaltung von Speicher, Prozessor, Dateisystem, Ein/Ausgabesteuerung, Verwaltung der Treiber usw.
F: Verwaltung von Ressourcen schon angesprochen: Prozessverwaltung
A: zwei Arten: Kurzeit- und Langzeitscheduling. Langzeitscheduling für PC nicht so wichtig, eher Lastbegrenzung bei Servern, also hier Kurzzeitscheduling: Zuteilung des Prozessors. Dabei zwei Arten: Präemptives und Nicht-Präemptives.

Beim Nicht-Präemptiven nur Einteilung in Warteschlange, dann laufen Prozesse solange bis fertig. Methoden:
- FCFS, in Ankunftsreihenfolge. Problem: zwei Jobs mit fast gleichzeitig mit unterschiedlicher Dauer -> langes Warten wenn langer Bruchteil vor kurzem
- Weiterentwicklung: SJF, kürzester zuerst. Problem: Verhungern langer Jobs
- HRN: Prozesse mit großem Verhältnis Wartezeit / Bedienzeit *(falsch! in der Prüfung niemandem aufgefallen; richtig: Antwortzeit/Bedienzeit)* zuerst, daher auch Jobs mit langer Wartezeit
- PS

Beim Präemptiven Scheduling unterbricht Dispatcher nach jeder Zeitscheibe. Methoden. RR, SRTF, DPRR. Länge der Zeitscheibe wichtig, nicht zu kurz oder zu lang, da sonst Prozessor nur mit Umschalten beschäftig oder Jobs bleiben bis von selbst fertig.

F: Speicherverwaltung, was grundsätzliches Problem?
A: grundsätzlich: Immer zu wenig Speicher.

Bei der Speicherverwaltung zwei Ansätze: direkte Speicherbelegung und virtueller Speicher.

Direkte Speicherbelegung durch feste Tabellen (ein Bit entspricht einem Wort, 0 oder 1 je nachdem ob frei oder nicht), oder verzeigerte Listen (Ordnen nach Größe, jeweils Länge, Anfang und Zeiger auf nächstes Element angeben).

Hier betrachtet: Belegungsstrategien – Speicher durchsuchen nach passender Lücke. Methoden: FF, NF, BF, WF, QF (wenn Anforderungen bekannt), BS (Anforderung aufrunden auf 2er Potenz, dann wie QF, wenn Aufteilen einer Lücke als Partner behandeln)

Virtueller Speicher: Problem der Zerstückelung des Speichers, daher virtueller Speicher, der unendlich groß und linear ist. Umrechung von virtueller Adresse auf physikalische Adresse (und der Speicher darf dann irgendwo liegen) über Seitentabelle. Auslagern einer Seite und einlesen einer anderen wenn mehr Platz benötigt als vorhanden.

Dabei Problem des sehr großen virtuellen Adressraums, der von den meisten Prozessen gar nicht ausgenutzt wird – Lösung: Adressbegrenzung, Multileveltabellen (Trennung Existenzinformation von Umrechnung), invertierte Seitentabellen (eindeutig durch Angabe

der Prozesskennung, da mehrere Prozesse die selbe virtuelle Adresse haben können, aber natürlich verschiedenen Physikalische) und Assoziativer Tabellencache, der wie invertierte Tabelle, aber auslesen in einem Zeittakt.

(Blick des Prüfers wird ungeduldig/gelangweilt von zu viel Text, also Abbruch – aber Lob das sehr gut gelernt)

F: Netwerkdienste, wie Hostadresse / IP - Adresse

A: IP-Adresse 32 Bit, oft dargestellt durch vier dreistellige Dezimalzahlen, durch Punkt getrennt: xxx.xxx.xxx.xxx

(Zwischenfrage Prüfer, ob wirklich vier und dreistellig, war wohl verwirrt, keine Ahnung – aber klar, da vier 8-bit-Gruppen, und die zwischen 0 und 255)

Nur Klassenbasierte IP-Adressierung angeschaut, da z.B. für Klasse A Netz: erste Zahl gibt Netzwerk an, zweite Gruppe von drei Zahlen den Host *(hab ich zuerst falsch herum gesagt, wurde von Prüfer korrigiert)*. Bei Klasse B Netz: zwei Gruppen, bei Klasse C Netz: drei Gruppen. Gibt auch noch Klasse D und E, aber die nicht verwendet sondern reserviert.

Bei Klasse A beginnt erste Zahl in Binärdarstellung immer mit 0, Klasse B mit 10, Klasse C mit 110

F: Was sind Sockets

A: Wenn Dienst im Internet muss der Wissen an wen er das Ergebnis schicken kann. Dazu IP-Adressen für Rechner, und 16-Bit-Port Adresse für Dienst. Ein Socket bindet an einen Port und bildet da den Endpunkt einer Punktorientierten Verbindung zwischen Server und Client, über die dann Pakete übertragen werden können.

Datenbanken

F: Datenbanken – was ist relationale Datenbank?

A: Datenbank = Tabelle, bei relationaler Datenbank gibt es Relation auf andere Tabelle, z.B. Firma mit mehren Firmensitzen. Das wäre dann 1:n-Relation

F: Es gibt da auch Anomalien, z.B. Insert-Anomalie

A: drei Anomalien, Update wenn man neuen Eintrag eingeben will, und man weiß noch nicht Daten für alle Spalten – müssten dann lehr bleiben, evtl. problematisch. Lösung ist Normalform

F: wie sieht erste Normalform aus?

A: gibt 7 Normalformen, drei davon werden verwendet.

Erste Normalform:

- eindeutiger Spaltenname
- Spalte beschreibt Eigenschaft eindeutig
- Zeile über Primärschlüssel adressierbar
- keine doppelten Zeilen
- keine sich wiederholenden Spalten (da auslagern in andere Tabelle)
- Zellen nur atomaren Wert, also keine weitere Tabelle

F: SQL ist allgemeine Abfragesprache für Datenbanken, einen Befehl hinschreiben:

A: Auswählen aus einer Tabelle, z.B. SELECT * FROM Tabelle WHERE Spalte1=5

Gibt alle Spalten der Tabelle mit dem Namen „Tabelle" aus, aber nur die Zeile in der Spalte1 den Wert 5 hat (also z.B. durchnummerierte Zeilen, davon die fünfte)

Statt * könne man auch nur einige Spalten eingeben, oder noch tiefer gehende Filterbedingungen einbauen.

Multimedia

F: Bei der Sprachdarstellung, wie kann man das machen. – nein nicht Sprache: Musik

A: Generell die Schritte Tiefpassfilter, Abtasten, Halten, AD-Wandlung, Quantisieren

Bei Musik Zerlegung in 32 Frequenzbänder über Fouriertransformation, dann Quantisierung nach Psychoakustischem Modell (Gehör nimmt leise Töne in der nähe von lauten Tönen nicht war = Maskierung. Daher kann man leise Töne weglassen, oder Rauschen einfügen, je nach dem was leichter zu kodieren). Danach Entropiekodierung.

Bei CD-Qualität nimmt man Abtastrate von 44,1kHz da Frequenz bei verlustfreier Kodierung doppelt so hoch sein muss, wie höchste Frequenz und die ist bei Musik/Sprache 20kHz. Bei CD Quantisierung mit 16 Bit.