Modulprüfung 1661 / 1801
Mathematik BSc, Nebenfach Informatik

Prüfer: Prof. Dr. Jörg Haake
Beisitzer: Dr.-Ing. Dirk Veiel
Termin: 16.01.2017
Note: 1,0

Prof. Haake beginnt mit einer kurzen Einführung in den Prüfungsablauf: Die Prüfung dauert 25 Minuten, besteht für den Studiengang Mathematik Bsc mit Nebenfach Informatik aus den Teilen Datenstrukturen I sowie Betriebssysteme und Rechnernetze. Jeder Teil sollte mindestens mit einer 4 bestanden werden. Außerdem fragt er noch, ob dies meine erste Prüfung bei der Fernuni ist, was ich bejahe. Und dann geht's auch schon los.

Datenstrukturen:

Was ist denn ein Algorithmus?
Ich: Ein Algorithmus ist mathematisch betrachtet die Implementierung einer Funktion auf algorithmischer Ebene, d.h. die Funktion definiert, was berechnet wird, und der Algorithmus beschreibt, was berechnet wird.

Ja, ganz richtig. Und was ist dann der Unterschied zwischen einem Programm und einem Algorithmus?
Der Algorithmus wird gebraucht, um einem Menschen einen Berechnungsweg mitzuteilen, damit er den dann in einer beliebigen Programmiersprache implementieren kann. Der Algorithmus muss also für Menschen verständlich und unabhängig von den Details einer Programmiersprache sein. Das Programm ist dagegen eine Sequenz von Befehlen und implementiert einen Algorithmus. *Dann hole ich noch kurz ein bisschen aus und beschreibe das Schaubild am Anfang vom Kurs Datenstrukturen. Prof. Haake nickt immer wieder zustimmend.*

Genau, alles richtig. Und woraus besteht so ein Algorithmus?
Aus Kontrollstrukturen wie z.B. if-/else oder Schleifen, aus Variablen der verwendeten Datentypen wie z.B. int-, boolean- oder Arrayvariablen, aus sprachlichen Anweisungen, was mit diesen Variablen passieren soll und ggf. dem Schlüsselwort return mit dem gelieferten Ergebnis.

Wie ermittelt man denn, ob ein Algorithmus effizient ist?
Das macht man mit der O-Notation. Also mit der Laufzeit.

Ja genau. Und wie kommt man darauf? Wie misst man diese Laufzeit?
Hier ging es um den Abschnitt im Kurstext von S. 6 bis ca. S. 12. Man abstrahiert von der Hardware, auf der eine Implementierung des Algorithmus betrieben wird. Dann werden die Elementaroperationen nicht mehr unterschieden... usw.

Nun hatten Sie ja eben die O-Notation erwähnt. Was genau besagt die denn?
Die O-Notation beschreibt, dass eine Laufzeitfunktion f maximal so schnell wächst wie eine Referenzfunktion g.

Ganz genau. Aber warum braucht man heutzutage diese Laufzeit- und Platzbedarfanalysen überhaupt noch? Die Geräte werden schließlich immer leistungsfähiger.
Weil ja parallel zum technischen Fortschritt auch die Anforderungen an Algorithmen und Datenstrukturen immer umfangreicher werden. **Naja, und insbesondere?** *Hier stand ich auf dem Schlauch.* **Was braucht denn ein Algorithmus?** Laufzeit? **Ja, aber was noch?** Platzbedarf? **Naja, womit arbeitet der Algorithmus denn?** *Ich war leider immer noch ahnungslos; Blackout.* **Na, mit Eingabedaten.** *Ach ja, stimmt ja. Dann ergänzte Prof. Haake noch das Stichwort Big Data: Die Datenmengen haben sich vervielfacht.*

Welche Laufzeit hat denn eine einfache Schleife?
O(n). **Genau, und wenn ich in der Schleife noch eine weitere habe?** Dann $O(n^2)$.

Richtig. Und wenn ich dann so eine Laufzeit habe wie z.B. $2n^2 + 4n + 12$, was macht die O-Notation?
Die lässt die additiven und multiplikativen Konstanten weg.

Ja, ganz richtig. Wie ist das denn beim Sortieren einer Menge. Wie ist da die beste Laufzeit?
Das wäre beim Sortieren durch Fachverteilen, da kann unter bestimmten Voraussetzungen lineare Laufzeit erreicht werden. Bei allgemeinen Sortierverfahren, also Schlüsselvergleichsverfahren, braucht man immer $\Omega(n \log n)$ Schritte. *Ob ich hier dann auch den Entscheidungsbaum erklärt habe, weiß ich leider nicht mehr.*

Wie ist das denn beim Suchen auf einer Menge. Welche Laufzeiten haben wir da?
Das ist ja die Operation member des Datentypen Dictionary. Primitive Implementierungen, das Hashing und der binäre Suchbaum haben im worst case lineare Laufzeit O(n). **Warum?** Weil der Suchbaum entarten kann. Beim Hashing, wenn alle Elemente in einem Behälter liegen. **Genau, also die maximale Anzahl von Kollisionen passieren.** Ja, und Laufzeit O(log n) gibt es bei der sequentiell geordneten Liste im Array wg. der binären Suche, beim binären Suchbaum im average case und beim AVL-Baum im worst case.

Ja, stimmt. Was ist denn der AVL-Baum?
Das ist ein binärer Suchbaum mit einer speziellen Struktur.
Genau, bei dem für jeden Knoten die beiden Teilbaumhöhen nur um 1 verschieden sein dürfen. Gibt es auch noch bessere Laufzeit fürs Suchen auf der Menge? Ja, es gibt auch ideale konstante Laufzeit O(1). **Und wann?** Beim Bitvektor, bei kleinem Grundtyp-Wertebereich, und beim Hashing im average case, wenn die Schlüssel- und Behälteranzahl ungefähr gleich groß sind und die Hashfunktion die Schlüssel gleichmäßig über die Behälter verteilt. **Und wann kann die Laufzeit beim Hashing auch im worst case konstant sein?**
Da bin ich verwirrt und kann nur raten. Prof. Haake muss die Frage selbst beantworten und schließt mit der Erkenntnis: **...das wäre dann aber praktisch ein Bitvektor.**

Dann ist die Zeit für die Datenstrukturen auch schon um, und es folgen gleich die Betriebssysteme

Was ist denn ein Prozeß?
Das ist der Ausführungsverlauf eines Programms, d.h. der Prozeß wird im laufenden Rechnerbetrieb gestartet, ist also ein Programm, das sich in der Ausführung befindet. Dafür werden die Befehle und Nutzdaten in den Hauptspeicher geladen, bilden da dann den Adreßraum und werden während der Ausführung sukzessive in die CPU geladen; Befehle werden ausgeführt und Nutzdaten werden verarbeitet. Dabei kann der Prozeß das Betriebssystem auch mit einer Ein-/Ausgabeaufforderung beauftragen und mit dem Benutzer in Kontakt treten.

Ja, ganz richtig. Und das Betriebssystem braucht zur Verwaltung eines Prozesses den Prozess…?
...kontrollblock. **Genau. Woraus besteht der?** Das sind die CPU-Registerinhalte, insbesondere der Befehlszähler, Basis- und Grenzregister. **Und sonst?** Die eindeutige Prozessnummer, dann System- und Benutzermodus, eine Liste mit den geöffneten Dateien. Im Mehrbenutzerbetrieb noch die Rechendauer, die der Prozess bereits in der CPU verbracht hat, und die Priorität des Prozesses. **Und beim Paging?** Da ist die Seitentabelle für die Seiten des Adreßraums wichtig.

Genau. Welche Adressierung braucht man denn im Systemmodus? *Hier habe ich keine Ahnung, worauf Prof. Haake hinauswill und frage, ob er die Frage noch mal umformulieren könnte. Er erwähnt dann die verschiedenen Arten der Adressierung: Relative, logische, physische, absolute Adresse.* **Welche dieser Adressen verwendet ein Prozess im Systemmodus?** Physische Adressen. **Genau. Weil es im Systemmodus keinen Speicherschutz geben muß.**

Wie läuft denn der Speicherschutz im Benutzermodus? Falls die Adreßräume als zusammenhängende Folge von Speicherzellen realisiert sind, wird zuerst geprüft, ob die relative Adresse $\geq$ dem Wert im Grenzregister ist. **Was passiert dann?** Dann ist sie zu groß für den Adressraum und eine Ausnahme wird ausgelöst. **Ja, und wenn das nicht der Fall ist?** Dann liegt die Zelle im Adressraum und die physische Adresse wird berechnet aus Basisregister plus Grenzregister. **Und beim Paging?** Da gibt es die Seitentabelle, die zeigt zu jeder Seite den Seitenrahmen und damit die erlaubten Bereiche an. **Genau.**

Und wie funktioniert die Berechnung der physischen Adresse beim Paging?

Man nimmt die höherwertigen Bits der logischen Adresse, die bilden die Nummer der Seite, auf der das Wort steht. Dann schlägt man damit in der Seitentabelle in dem Eintrag mit Index Seitennummer nach. **Was steht da drin?** Die Seitenrahmennummer. **Und was macht man damit?** Die konkateniert man mit den niederwertigen Bits der logischen Adresse. Dann hat man die physische Adresse. **Richtig.**

Dann geht's weiter mit den Rechnernetzen

Welche Protokolle sind in der Vermittlungsschicht? Das Internet Protocol, das DHCP und das ICMP.

Ja, genau. Was macht das IP-Protokoll?
Das verpackt ein Transportschichtsegment in ein IP-Datagramm und sorgt dann für die Weiterleitung des Datagramms im Netzwerkgerät.

Wie erfolgt denn die Weiterleitung?
Das IP-Protokoll ermittelt aus der Routingtabelle die IP-Adresse des Zielnetzwerks. **Nein nein – das stimmt so nicht. Was wird ermittelt?** *Nach kurzem Zögern:* Die Basisadresse! **Naja, die Netzwerkadresse.** Oh ja, stimmt. Bei der sind die Bits des Host-Teils auf 0 gesetzt. **Ja. Und wie wird diese Adresse ermittelt?** Die ermittelt man aus der IP-Adresse des Datagramms, anhand der Netzwerkmaske.

Ganz genau. Was liest man aus der Routing-Tabelle noch aus? Die Hops bzw. Distanz, also die Netzwerke, die auf dem Pfad zum Zielnetz passiert werden müssen. **Genau, und bei Routern noch die IP-Adresse der Router-Schnittstelle.**

Und wie kommt das IP-Protokoll an die Adresse ran? Dazu muss die Adresse aus dem Segmentheader ausgelesen werden. Dafür muss das Paket als Ganzes vorliegen.

Was ist denn, wenn ein Datagramm nicht weitergeleitet werden kann? Wie findet das IP-Protokoll das heraus?
Mittels Fehlererkennung. **Nein, genauer. Was findet man denn im Header eines IP-Datagramms?** Also, die IP-Adresse des Empfängers, die des Absenders, ein Längenfeld, d.h. wie viele Bytes genutzt werden, außerdem noch Informationen zur Fragmentierung und Defragmentierung, darunter die Identifikations-ID des Datagramms, ein Flag und ein Fragmentation Offset, dann die Lebensdauer im Time-To-Live-Feld, und eine Möglichkeit zur Fehlererkennung… **Naja, die Checksum.**

Welche Teile des IP-Datagramms werden denn mit der Checksum geprüft?
Nur der Header, weil nur er zur Weiterleitung des Datagramms wichtig ist.

Und wenn die Prüfung negativ ausfällt?
Dann verwirft der Router das Datagramm und informiert den Quellhost mit einer ICMP-Warnnachricht. **Ja, wie soll er das denn machen? Er weiß doch gar nicht, ob die IP-Adresse nicht fehlerhaft ist. Er wird das Paket natürlich einfach verwerfen.** *Oh je, stimmt. Das hätte ich im Normalzustand gewusst.*

Was berechnet die Routen denn?
Die Routen werden über Routingprotokolle berechnet. **Und was ist da das Besondere?** Die Routingprotokolle sind auf der Anwendungsschicht angesiedelt, und nicht auf der Vermittlungsschicht, d.h. hier wird das Internet-Schichtenmodell eigentlich etwas verletzt. **Genau, und warum macht man das?** Weil die Routingprotokolle zur Pfadberechnung Nachrichten austauschen und dafür die Dienste der Transportschicht nutzen. **Ganz genau.**

Dann ist die Zeit auch schon vorbei und ich werde kurz raus ins Sekretariat geschickt, bevor mir Prof. Haake dann das Ergebnis mitteilt.

Fazit:
Auch ich kann nur bestätigen, dass Prof. Haake ein ausgesprochen fairer Prüfer ist. Ich hatte einen gehörigen Respekt vor dieser Prüfung, hatte mich im Vorfeld akribisch drauf vorbereitet und war extrem nervös. Dafür muss Prof. Haake ein Gespür besessen haben, sodass er die Blackouts nicht zu stark bewertete. Man kann also in der Prüfung durchaus mal etwas neben sich stehen; wenn man gut vorbereitet ist und den Großteil der

Fragen beantworten kann, wird Prof. Haake dies auch berücksichtigen. Die Prüfung besteht dabei aus einer Kette schneller und prägnanter Fragen, wodurch die 25 Minuten wie im Nu vergehen. Prof. Haake signalisiert sofort, wenn man eine Frage ausreichend beantwortet hat. Ist dabei der Eindruck entstanden, dass man das Themenfeld beherrscht, dann fährt er mit einem anderen Thema fort. Anderenfalls hakt er nach. Ich habe nur selten längere Ausführungen machen müssen, meistens reichte es, zu zeigen, dass man das Themenfeld beherrscht, z.B. bei den Laufzeiten beim Suchen auf einer Menge.
Ich hoffe, das Protokoll kann bei der Vorbereitung etwas helfen, und wünsche allen Mitstudierenden viel Erfolg und gutes Gelingen!

Bachelorprüfung „Softwaresysteme“

Prüfer: Prof. Dr. Jörg Haake
Kurse: 1671 Datenbanken I & 1801 Betriebssysteme und Rechnernetze
Termin: 12.02.2007
Dauer: etwa 25 Minuten
Note: 2.0

Ich kann den anderen Protokollschreibern anschließen: Prof. Haake ist nett und schafft eine angenehme Atmosphäre. Er fragt sehr ruhig und zielgerichtet, es ist eigentlich immer klar, auf was er abzielt.
Die Protokolle sind eine sehr gute Vorbereitung für die Prüfung. Die meisten Fragen, die mir gestellt wurden, sind aus den vorliegenden Protokollen bekannt gewesen. Dieses Protokoll sollte daher auch eher als Ergänzung verstanden werden!

Teil 1) Betriebssysteme

- Was sind Prozesse und Programme? Was sind Threads und woraus bestehen sie?
- Was ist ein kritischer Abschnitt? Wo ist der kritische Abschnitt bei der Leser/Schreiber-Problematik? (Ein Prozess schreibt ein Wort in den Speicher, ein anderer Prozess liest das Wort)
  [Wichtig: Das Problem besteht hierbei, wenn der schreibende Prozess die Zeitscheibe verliert, nachdem <u>nur einzelne Bits</u> geschrieben wurden. Der lesende Prozess kann dann nur falsche Daten lesen. Damit ist der gesamte Schreib- und Lesevorgang jeweils ein kritischer Abschnitt.]
- Was passiert bei einem Interrupt genau? Wie wird ermittelt, welcher Prozess ausgeführt werden soll?
  [Hier wollte er insbesondere darauf hinaus, dass die Startadresse über einen Vektor ermittelt wird.]

Teil 2) Rechnernetze

- Welche Protokolle gibt es auf der Transportschicht?
- Welches Protokoll würden Sie auf der Transportschicht für ein Internet-mp3-Radio verwenden? Was hat dies für Folgen?
  [Wichtig: Bei der Verwendung von UDP ist auf der Empfängerseite ein (kleiner) Puffer notwendig, der die Pakete kurzzeitig aufnimmt und in der richtigen Reihenfolge abspielt.]

Teil 3) Datenbanken

- Welche Nachteile haben Dateisysteme?
- Welche Datenbankschichten gibt es und woraus bestehen sie?
  [Die interne Schicht enthält neben Sekundär-Index und Zugriffspfaden auch Informationen über die Datenrepräsentation, also bspw. wieviele Bits für ein Integer-Feld benötigt werden u.ä.]
- Welche Datenbankmodelle gibt es?
  [Alles aufzählen, auch wenn es nicht im Kurstext genauer erläutert wird.]

Generell ist es wichtig, dass ihr nicht nur die Antworten aus den Prüfungsprotokollen nehmt und auswendig lernt, sondern den gesamten Sachverhalt versteht. Überlegt euch, wie die Themen zusammenhängen und vor allem, was sich dahinter verbirgt, z.B. woraus die einzelnen Schichten bestehen (→ Thema Datenbanken: Was ist in welcher Datenbankschicht enthalten).
Die Benotung ist fair, auch wenn ich mich anfangs geärgert hatte, da ich die Themen sehr gut vorbereitet hatte. Nur an ein paar Stellen hing ich dann leider und daher wurde es „nur“ eine 2.0.
Aber auch von mir eine Empfehlung, die Prüfung bei Prof. Haake zu belegen. Viel Erfolg!

Prüfung in Praktischer Informatik (1661/1801)

Termin: 08.01.2007
Dauer: 25 Minuten
Prüfer: Prof. Haake
Note: 1,0

Ganz am Anfang des Kurses Datenstrukturen hatten wir ein Schaubild, da waren die drei Ebenen der Betrachtung von Algorithmen bzw. Datenstrukturen angegeben, welche Ebenen waren das?
Programmierung, Algorithmik und Mathematik (für Mathematik musste er mir etwas helfen).

Und so ein Algorithmus mathematisch betrachtet ist?
Eine Funktion, die bestimmten Eingaben bestimmte Ausgaben zuordnet.

Woraus besteht ein Algorithmus?
Aus den einzelnen Verarbeitungsschritten.

Und was gilt für diese Schritte?
Sie müssen jeder für sich verständlich und in endlicher Zeit durchführbar sein. Nicht gefordert ist hingegen die Endlichkeit des gesamten Algorithmus.

Können Sie ein Beispiel geben für einen nicht terminierenden Algorithmus?
Z.B. in der theoretischen Informatik zu bestimmen, ob ein bestimmtes Element in einer aufzählbar rekursiven Menge enthalten ist. Wenn es enthalten ist, terminiert das Verfahren, wenn nicht, dann läuft es endlos. **Ja, so etwas mehr aus dem praktischen Leben vielleicht noch eine Ampelsteuerung, die sollte ja auch nicht terminieren.**

Wie misst man denn die Laufzeit eines Algorithmus?
Das real zu messen ist schwierig, dafür muss man ihn implementieren und auf einer realen Maschine messen. Dies wird man in der Regel jedoch nicht wollen, die Ergebnisse hängen ja nun stark ab von der Implementierung und der verwendeten Maschine. Deshalb abstrahiert man auf ein mathematisches Modell, z.B. Touring-Maschine, Register-Maschine oder RAM und zählt die Elementaroperationen (Zuweisungen, Vergleiche, Array-Zugriffe, mathematische Operationen).

Und wie geht man dann weiter vor?
Auch diese Ergebnisse vereinfacht man weiter, man betrachtet nur noch die Elementaroperationen in ihrer Gesamtzahl, teilt die Eingaben in Komplexitätsklassen und diese in Gruppen (Best, Worst, Avarage) auf. Dann schaut man, wie sich diese Zahl verändert bei Veränderung der Komplexität.

Wenn man dann etwas rausbekommt wir 23n^2 + 4n +28, was fängt man damit an?
Es interessiert ja nur die Größenordnung, eine obere Laufzeitabschätzung, man behält also nur den dominierenden Summanden, und auch davon nur den nicht konstanten Anteil, in dem Fall n^2. Das ist dann die Ordnung.

Eine Ordnung O(n^2), wie könnte die zustande kommen?
Z.B. durch ein einfaches Sortierverfahren, wobei jedes Element mit jedem Vergleichen wird, also zwei ineinander geschachtelte Schleifen.

Und so was wie 2n?
Durch einfachen Durchlauf, und mit jedem Element wird irgendwas gemacht.

Bei einer Menge, wenn man feststellen will, ob da ein bestimmtes Element drin ist, welche Ordnung hat das?
Auf einer unsortierten Menge muss man die Menge durchlaufen, im Worst-Case die gesamte, im Mittel die halbe, es hat also O(n). Bei sortierten Mengen kann man binär suchen, dann O(log n).

Geht das nicht auch schneller?
Wenn die Mengen in einen Hash einsortiert sind, dann ja, dann geht es im Mittel mit O(1), wenn der Hash nicht zu voll ist. Bei einem offenen Hash können ja die Ketten sehr lang sein.

Und wenn man O(1) garantieren will?
Man muss einen geschlossenen Hash verwenden, ein hinreichend kleines Universum, wenig Eingaben, viele Fächer, und eine gute Kollisions-Strategie (Das war eher ein gemeinsames Fazit von ihm und mir).

So ein Abstrakter Datentyp, der wird ja durch Axiome definiert, was ist der Vorteil davon?
Es ist ein logischer Ansatz, er kann beweisbar richtig, widerspruchsfrei und vollständig sein. **Bei kleinen Datentypen aber nur.** Außerdem definiert es nur die Aspekte, die man definieren möchte, man kann Aspekte offen lassen.

Und wenn der ADT vollständig und widerspruchsfrei ist, ist die Algebra dann eindeutig?
Es gibt immer noch beliebig viele Implementierungen, die aber isomorph sind.

Es kamen noch ein oder zwei Fragen zu ADT, das Thema schien er zu mögen.

Dann springen wir mal in den anderen Kurs. Wir hatten ja Scheduler kennen gelernt. was für Strategien gibt es da?
Im Batchbetrieb First Come First Served oder Shortest Job First, im Time Sharing Betrieb Zeitscheiben, im einfachsten Fall mit Round Robin, komplizierter mit Prioritäten oder ähnlichem.

Gut, Zeitscheiben, wenn ein Job denn gerade läuft, wer unterbricht ihn dann, wenn seine Zeitscheibe abgelaufen ist?
Der Scheduler.
Und wie?
Per Interrupt.
Wie kann er das denn, wenn es nur ein Ein-Prozessor-System ist? Er läuft ja gerade gar nicht.
Durch einen Timer.
Richtig, ein Timer-Interrupt, wie läuft das denn dann ab? Was macht der Prozessor?
Der Interrupt wird an den Prozessor als Signal angelegt, der aktuelle Befehl wird noch zuende bearbeitet, der Kontext gesichert, dann wird im Interrupt-Vektor nachgesehen, was zu tun ist, und es wird der Scheduler gestartet.

So ein Betriebssystem soll ja sicherstellen, dass niemand fremden Speicher beschreiben kann. Wie macht es das?
Im einfachsten Fall, wenn es nur zusammenhängende Adressräume gibt, gibt es ein Grenzregister, wenn versucht wird, eine Adresse jenseits dieser Grenze zu lesen, wird ein Interrupt ausgelöst, und das OS kann prüfen, ob der Zugriff berechtigt ist. Wenn der Speicher segmentiert ist, so ist ja in jedem Segment vermerkt, welcher Prozess oder auch Prozesse darauf zugreifen können.

Gut, dann noch ein wenig zu Netzwerken. Im Internet gibt es ja das IP-Protokoll, was ist dessen Aufgabe?
Es verbindet einzelne Rechner miteinander, schafft ihnen einen transparenten Kanal, mit dem sie kommunizieren können.

Dazu muss es ja geroutet werden, unter Umständen durch mehrere Transit-Systeme, und es muss ja ein Pfad ermittelt werden, benutzen eigentlich alle Router im Internet das gleiche Routing-Protokoll.
Nein, das wäre alleine wegen der Vielzahl nicht durchsetzbar, und größere Organisationen wollen das auch gar nicht. Man hat das Netz in autonome Systeme unterteilt, die ihre eigenen Routing-Protokolle fahren, untereinander wird mit einem Inter-Routing-Protokoll vermittelt, wir haben das BGP4 kennengelernt.

Wie läuft das denn jetzt ab, wenn mein Rechner Ihrem was schicken will, wir hier in der FU ja RIP benutzen?
Ihr Rechner wird es an den nächsten Router schicken, diese Router haben sich verständigt, welches der günstigste Pfad zum günstigsten Gateway ist, dann geht das durch evtl. ein paar Router bis zum Netz der T-Com und deren Router haben sich mit welchem Protokoll auch immer verständigt, wie das Paket zu mir kommt.

Und ist sichergestellt, dass die Pakete immer die gleiche Route nehmen?
Nein, es könnte sein, im Netz Router existieren, die per Lastverteilung über verschiedene Routen schicken, oder dass während der Übertragung der Pfad ausfällt, und ein alternativer Pfad genutzt werden muss.

Was tut der Router denn dann, wenn ihm ein Pfad ausfällt?
Er wird den entsprechenden Pfad in seiner Distanztabelle mit Unendlich markieren, und diese Information an die Router propagieren auf die Art, die das Routing-Protokoll vorsieht.

Und was macht er mit dem Paket, was er gerade Routen wollte?
Er muss es verwerfen und schickt eine ICMP-Nachricht zurück.
Keine andere Möglichkeit?
Falls er noch einen alternativen Pfad kennt, kann er es darüber rausschicken.

Gut, und wenn der Router keinen kennt, das Paket verwerfen muss, ist dann sichergestellt, dass keine weiteren Pakete mehr in die Sackgasse laufen?
Nein, das Problem ist, dass in vielen Routing-Protokollen diese schlechten Informationen nur relativ langsam propagiert werden, es also schon einige Iterationen dauern kann. Das nennt man das Count-To-Infinity Problem.

Wir haben ja über TCP gesprochen, dass Pakete dann erneut gesendet werden. Wenn sie gar nicht zugestellt werden, passiert das ja nach einem Timeout. Jetzt muss man sich ja überlegen, wie groß dieser Timeout-Wert sein soll. Wenn nun auf der anderen Seite

jemand dringend auf die Pakete wartet, dann könnte man den Timeout ja klein wählen, damit die Pakete schneller neu gesendet werden. Was könnte da zum Problem werden?
Wenn die Pakete eh schon langsam fließen, weil zB eine Leitung verstopft ist, dann würde man durch diese Taktik noch mehr Pakete erzeugen, das Problem also noch weiter vergrößern, es würde immer voller werden.

Und bei zu großem Timeout-Wert?
Das gleiche Resultat aber durch andere Ursache. Das Sendefenster wäre ausgelastet, man hat keine freien Sendenummern mehr, und die Timer sind noch nicht abgelaufen, also kann man keine Pakete mehr schicken.

Wenn der Empfänger nur einen Teil der Pakete verloren hat, was könnte er dann tun?
Er kann per Selektive Resend Request den Sender zum erneuten Senden auffordern und hoffen, dass dieses Pakete ankommt, bevor er in einen Timeout läuft. Dann werden nur die verlorenen oder verstümmelten Teile neu gesendet

Insgesamt war die Prüfungsathmosphäre sehr angenehm, ich kann Herrn Prof. Haake als Prüfer empfehlen. Man bekommt zu jeder Frage sofort ein Feedback, wird also nicht hängen gelassen, und weiß ob es gut läuft. Wenn er nicht ganz zufrieden ist, ihm die Antwort also nicht reicht, fragt er nach, auch sehr gezielt um einem auf die Sprünge zu helfen. Auch während man die Antwort gibt, nickt er, so dass man weiß, dass man auf dem richtigen Weg ist. Er lässt einen schon ausreden und auch ein wenig um das Thema herum erzählen, erwartet also nicht nur kurze und einsilbige Antworten und stellt dann auch teils bewusst etwas offene Fragen. Mir schien es, als bevorzugt er eher einen Überblick des Wissens, nicht unbedingt die letzten Details, wie ein bestimmter Algorithmus funktioniert oder ähnliches.

Mit Sicherheit hab ich die in oder andere Frage noch vergessen, aber zumindest sollten alle Themen und ihre Tiefe vorhanden sein.

Viel Glück allen Nachfolgenden.

Gedächtnisprotokoll
Fachprüfung Praktische Informatik 01661 / 01801
Prüfer: Prof. Dr. Haake
11.12.2006
Note 1,0

Zur Situation:
Die Atmosphäre war –im Gegensatz zu mir- ruhig und entspannt und es war eher ein Fachgespräch am runden Tisch als das, was ich bisher (außerhalb der FernUni) als mündliche Prüfung kennen gelernt habe.
Nach einer kurzen Aufwärmphase und Erklärung zum Ablauf der Prüfung kann ich mich an folgende Fragen erinnern:

Datenstrukturen:

Welche 3 Ebenen von Datenstrukturen werden im Kurs behandelt?
- Siehe Schaubild auf der 1.KE

Was wird auf Ebene der Mathematik betrachtet?
Wie heißt denn das was da aufgestellt wird?
- Die Antwort wäre hier „Funktion“ gewesen

Was ist ein Algorithmus?
Was ist ein Datentyp?
Welche Bestandteile hat so ein Datentyp?
Wenn ich nun so einen Datentyp nicht auf eine bestimmte Trägermenge beschränken will, was kann ich dann machen?
- ADT

Was ist denn ein ADT?
Gibt es den auch in der Praxis?
Wenden wir uns nun den Algorithmen zu.
Wie kann ich denn messen wie lange ein Algorithmus zur Bearbeitung braucht?
Wenn ich das nicht berechnen kann, wie kann ich dann vorgehen?
Wie sind denn die Abstraktionsschritte?
Wenn ich eine unendliche Eingabemenge habe, muss ich dann für jede Eingabe überprüfen wie lange der Algorithmus läuft?
Was ist denn die Eingabekomplexität genau?
Brauchen wir bei der heutigen technischen Entwicklung –die Rechner werden ja immer schneller und haben immer mehr Speicher- eigentlich diese Laufzeitberechnungen noch?
Gibt es eigentlich auch Probleme von denen es wünschenswert ist, wenn sie nicht gelöst werden können?
- Hier wollte er auf Verschlüsselung hinaus.

Betriebssysteme:

Was ist die Aufgabe eines Betriebssystems?
Wenn ich einen Rechner habe und viele Hersteller von Festplatten, wie kann das trotzdem so funktionieren, dass der Benutzer auf die Daten zugreifen kann?
-Konstrukt des virtuellen Gerätes

Wie wird eine i/o Anforderung abgearbeitet?
Wie wird denn der Prozessinhalt gerettet?
Welche Struktur ist dazu notwendig?
- Auf den Prozesskontrollblock kam ich erst nach etwas Nachdenken

Rechnernetze:

Was sind die Aufgaben eines Routers?
Wie leitet er ein Datagramm weiter?
Woher bekommt denn der Router seine Informationen?
Was macht das IP Protokoll?

Die Verteilung der Fragen war ungefähr 50% Datenstrukturen und jeweils 25% Rechnernetze und 25% Betriebssysteme.
Abschließend kann ich Prof. Dr. Haake als Prüfer nur empfehlen, ich empfinde die Benotung als sehr fair und hatte nach meinen Anlaufschwierigkeiten nicht mit dieser Note gerechnet.
Alles Gute für eure Prüfung.

Bachelorprüfung Praktische Informatik

Prüfer: Prof. Dr. Jörg Haake
Kurse: 1661 und 1801
Termin: 18.10.2005
Dauer: 25 Minuten

Fragen zu 1661:

- Was ist ein Algorithmus?
- Für wen schreibt man einen Algorithmus?
- Woraus besteht ein Algorithmus?
- Was ist ein Datentyp?
- Wie kann man ihn beschreiben/darstellen?
- Wie ist der Begriff, wenn es mehrere verschiedene Algebren gibt, die denselben abstrakten Datentyp befriedigen?
- Wie kann ich so einen polymorphen Datentypen in einer objektorientierten Programmiersprache implementieren?
- Wie ist denn die Komplexität des Problems eine n-elementige Menge zu sortieren?
- Es gibt Algorithmen, die O(n log n) erreichen. Wie schaffen sie das?
- Woher kommt eigentlich der Logarithmus in der Laufzeitfunktion? (*intuitive Erklärung reichte hier aus – keine mathematische Herleitung.*)
- Wenn ich eine Menge sooft wie möglich durch 3 teile, welcher Logarithmus ist das dann?
- Welche Datenstruktur verwenden sie, um zu prüfen ob ein bestimmtes Element in einer Menge enthalten ist?

Fragen zu 1801 (Betriebssysteme):

- Was ist der Unterschied zwischen einem Programm und einem Prozeß?
- Wie heißt die Datenstruktur, welche die Prozesse verwaltet?
- Welche Scheduling-Strategien kennen sie?
- Wann ist *Round-Robin* sinnvoll?
- Nutzen denn alle Prozesse ihre Zeitscheibe voll aus?
- Wie kann ein Prozess seine Zeitscheibe vorzeitig freigeben?
- In welchen Fällen ist *shortest-job-first* sinnvoll?

Fragen zu 1801 (Rechnernetze):

- Nennen sie die fünf Schichten des Internet-Schichten-Modells aus dem Kurs!
- Nennen sie zu jeder Schicht ein typisches Protokoll und wer mit wem kommuniziert!
- Geben sie mir ein Beispiel für zwei Anwendungen die über HTTP kommunizieren!
- Wie schafft es die Vermittlungschicht zwei Hosts miteinander zu verbinden?
- Was passiert wenn in einem Netzwerk ein Router ausfällt?
- Wir haben einen dezentralen Routing-Algorithmus. Wie weit verbreitet sich diese neue Information dann im Netzwerk?
- Was ist Multicasting?
- Müssen denn alle Router auf dem Weg multicast-fähig sein?
- Wie funktioniert denn dieses *tunneln*?

Prüfungsprotokoll 1661/1801

Datum 14.3.2005
Prüfer: Prof. Haake
Beisitzer: hab ich vergessen
Zeit: 15:00

Die ganze Prüfung lief in einer freundlichen Atmosphäre ab. In meinem Erststudium (nicht an der Fernuni) hatte ich bei solchen Prüfungen oft eine herablassende Art der Profs erlebt, und auch den unterschwelligen Wunsch der Prüfer, durch möglichst „bescheuerte“ Fragen herauszustellen, wie superschwer „ihr“ Fachgebiet ist.
Hier keine Spur davon. Alles war wie gesagt sehr nett und fair, in dem Sinne: wer sich gut vorbereitet, kriegt hier auch eine eins(hat bei mir dann auch erfreulicherweise geklappt ☺)
Die Fragen und Antworten sind unvollständig aus dem Gedächtnis wiedergegeben, ohne Anspruch auf Vollständigkeit oder Richtigkeit, insbesondere meine Antworten waren natürlich viel ausführlicher.

Was ist eine Algorithmus?
siehe script.
Für wen schreibt man einen Alg.?
Für einen Menschen, deshalb Mischung aus Alltagsspr. Und Prog-Spr.
Warum nicht Prog-Sprache?
Zu speziell, stellung im Schaubild erklärt.
Was macht die Qualität eines Alg. aus?
siehe Scipt
Kann man all diese Dinge gleichzeitig erfüllen?
Nein, - trade-off
Beispiel für Trade-Off?
Bubble-sort – DAC
Wie kommt man denn zur O-Notation?
siehe script
Wie ist denn die Laufzeit fürs suchen auf einer Menge?
O(n)
Wie muss man eine Menge dafür implementieren?
Dictionary
was, wenn die Menge gar nicht der Grösse nach sortiert werden kann?
Hashing.
Wie ist denn die Laufzeit beim hashing?
je nachdem wie effektiv, O(1) bis O(n)

Nun kamen wir zu Betriebssystemen.

Unterschied Prozess-Programm?
Prozess=Prog, das ausgeführt wird.
was gehört alles zu einem Prozess?
siehe Script, der Prof wollte noch hören: Die Dateien, die der prozess geöffnet hält.
Welche Segmente hat so ein P.?
Daten / Prog / Stack
was teilen Threads?
siehe script
wie kann man P. synchronisieren?
switch- var, Nachteile erwähnt, Semaphor erklärt
in welcher Reihenfolge werden denn Betriebsmittel an die wartenden P. verteilt?
war mir unklar, ich riet: Zufall oder Prioritäten, der Prof brachte mich dann noch auf das naheligende FCFS

Wir kamen zum Thema Netze.

schichtmodell mit dazugehörigen Protokollen.
erklärt
Wird im WWW überall das gleiche Routing Protokoll verwendet
Nein, erklärt AS/ Inter AS / Intra AS / RIP / BGP4
Was passiert denn also genau, wenn ich eine Nachricht von Host1 an Host2 schicke?
Intra AS Host1-> Gateway -> Inter AS->Gateway-> Intra AS Host2->Ziel Host

Das wars, nach einer extrem kurzen Wartezeit wurde ich wieder reingerufen, bekam meine Note (1.0) mitgeteilt und verliess guter Dinge die Fernuni wieder. Prof Haake kann ich als Prüfer auf jeden Fall empfehlen, ich glaube dass man sich bei guter Vorbereitung bei ihm auch einer guten od. sehr guten Note sicher sein kann. Er hat bei mir, obwohl ich so gut wie alles beantworten konnte, keinen ehrgeiz entwickelt, solange "in die Tiefe" zu gehen, bis ich irgendwann hätte passen müssen, was ihm sicherlich gelungen wäre. Durch diese Vorgehensweise kommt bei der Prüfung auch wenig Stress auf, man hat die ganze Zeit das Gefühl, dass es ok ist, wie man antwortet.

Gedächtnisprotokoll
Fachprüfung Praktische Informatik 01661/01801
Prof. Dr. J. Haake
Beisitzer Prof. Dr. J. Keller
14.03.2005
Note 1,3

Am Anfang erklärte mir Prof. Haake, wie sich die Prüfung zeitlich gliedern wird (jeweils 1/3 der 25 Minuten für Datenstrukturen/Algorithmen, Betriebssysteme und Rechnernetze). Dann ging es mit der ersten Frage los:

Datenstrukturen/Algorithmen:

Was ist ein Algorithmus?
Ein Verfahren zur Lösung eines Problems. Es besteht aus einzelnen Schritten und Anweisungen, die die Ausführung dieser Schritte kontrollieren. Jeder dieser Schritte muss mit endlichem Aufwand in endlicher Zeit terminieren. Algorithmen werden in einer Mischung aus programmiersprachlichen Elementen und allgemeiner Sprache beschrieben.

Ist es wichtig, an welche Programmiersprache Algorithmen angelehnt werden?
Nein, Algorithmen sollen in jeder Programmiersprache implementierbar sein, insbesondere sollte man keine Tricks benutzen, die nur eine spezielle Programmiersprache anbietet.

Wozu werden die Algorithmen eigentlich gebraucht?
Um jemanden mitzuteilen, wie etwas berechnet wird.

Welche Kriterien muss so ein Algorithmus erfüllen?
Das K.O. Kriterium ist die Korrektheit. Darüber hinaus sollen Algorithmen leicht implementierbar, leicht verständlich und sparsam bezüglich Rechenzeit und Speicherbedarf sein.

Können alle diese Kriterien gleichzeitig erfüllt werden?
Nein, bezüglich der 3 letztgenannten Kriterien gibt es “trade-offs”, z.B. ist ein einfacher Algorithmus meistens ineffizient.

Können Sie da auch ein Beispiel angeben?
Sortieralgorithmen: SelectionSort ist einfach implementierbar, hat aber $O(n^2)$ Laufzeit, Quicksort ist komplizierter und hat nur $O(n * \log n)$ Laufzeit.

Wie kann die Qualität eines Algorithmus angegeben werden?
Durch Angabe der Laufzeit und des Speicherbedarfs. Dann habe ich das ganze Verfahren erklärt, von Zählen der Elementaroperationen über die Abstraktionsschritte bis zur O-Notation.

Welchen Sortieralgorithmus nehmen Sie dann bei n = 5, Quicksort oder SelectionSort?
SelectionSort, nein, ein Moment... (hab dann ein wenig im Kopf gerechnet)... doch Quicksort.

Was muss man also bei kleinen n beachten?
Das die O-Notation erst für große n eine Orientierung gibt. Bei kleinen n soll man sich die konstanten Aufwände genauer anschauen.

Was ist ein Datentyp?
Eine oder mehrere Objektklassen zusammen mit den zugehörigen Operationen.

Welche Möglichkeiten gibt es, die Semantik eines DT anzugeben?
Entweder als Algebra (mathematisch genaue Notation) oder als abstrakter Datentyp (Axiome).

Was beschreiben die Axiome?
Hierauf kam ich mit Hilfe des Professors: Axiome geben nicht an, wie etwas berechnet wird, sondern was für das Ergebnis gelten muss.

Und was ist der Vorteil eines ADT?
Es können polymorphe Datentypen spezifiziert werden, d.h. mehrere Algebren können die Axiome erfüllen.

Mit welchem Konstrukt werden denn polymorphe DT in z.B. Java implementiert?
Hier habe ich ziemlich verunsichert Vererbung, mit einer kleinen Hilfe dann Interface genannt.

Betriebssysteme:

Was ist ein Prozess?
Ein Programm, das sich gerader in der Ausführung befindet. Ein Prozess hat seinen Kontext (Registerinhalte, insbesondere Befehlszähler und Stack sowie Adressraum)

Was fehlt noch?
Datensegment.

Welche Zustände kann ein Prozess annehmen?
Erzeugt-bereit-rechnend-blockiert-beendet.

Was für ein Problem gäbe es denn, wenn ein blockierter Prozess direkt in den Zustand "rechnend" übergehen konnte?
Hier brauchte ich auch einen kleinen Denkanstoß: die Schedulingstrategie (Fairness) wird durcheinander gebracht.

Welche Schedulingstrategien kennen Sie denn?
SJF, FCFS und Time-Sharing (round robin).

Was passiert, wenn bei dem round-robin Zuteilung die Zeitscheibe zu klein ist?
Der Overhead für den Prozesswechsel fällt sehr stark ins Gewicht.

Rechnernetze:

Zählen Sie die Internetschichten auf und geben Sie zu jeder Schicht Name, Aufgaben, Beispiel eines Protokolls und die Entitäten, zwischen denen eine Verbindung hergestellt wird.
Hier habe ich das Internetschichtenmodell laut Kurstext wiedergegeben.

Sie haben zu der Vermittlungsschicht das IP Protokoll genannt. Was fehlt noch?
Routing.

Was passiert, wenn ich einen neuen Router ans Netz anschliesse?
Der Router macht sich im Netzwerk bekannt (ich weiss nicht, ob das so stimmt ;-)), dann erfolgt der Austausch der Kosten untereinander und Aufbau der Routingtabelle.

Und wie sieht so eine Routingtabelle aus?
Zu jedem erreichbarem Netzwerk gibt es einen Eintrag, die Spalten sind dann die Netzwerkmaske, Kosten und... (kleine Hilfe des Professors: Der Router hat doch so viele Anschlüsse..) das Interface

Wie baut denn der Router seine Routingtabelle auf?
Hier wollte ich sagen, dass es verschiedene Algorithmen gibt, nach denen die Tabellen aufgebaut werden, aber erwartet wurde das: Der Router bekommt doch von seinen Nachbarn ihre Tabellen zugeschickt. Wie macht er daraus seine eigene Tabelle? Professor Haake hat das dann selbst beantwortet: Die Tabellen werden zusammengefügt, sortiert und die ungünstigen Routen gestrichen.

So, das war's. Professor Haake stellt die Fragen in einem praktischen Zusammenhang ("Sie implementieren ein Buchungssystem für Lufthansa..."), die Fragen in dem Protokoll sind von mir konkretisiert worden. Bei einigen Fragen bat ich ihn, die Frage ein wenig umzuformulieren, was meistens geholfen hat, aber keinen Einfluss auf die Note hatte. Angesichts einiger kleiner Unsicherheiten und der letzten Frage bin ich mit der Note absolut zufrieden und kann Professor Haake uneingeschränkt weiterempfehlen.

mündl. Prüfung 1661 / 1801
Prof. Haake
19.4.2004
Ablauf:
Es ging ohne großes Geplänkel gleich zur Sache.
Da Prof. Haake sich mit dem Prüfungsstoff (Netzwerke) vertan hatte, ging es sofort mit Rechnernetzen (und etwas mehr ...) los:

Erläuterung des ISO/OSI -Referenzmodell und Protokolle.
Ich habe zusätzlich die Geräte der jeweiligen Schichten erläutert und deren Aufgabe kurz skizziert.
Habe auch kurz den Sinn und Zweck dieses Schichtenmodell genannt.

Wozu dient ein Router? Was passiert, wenn man einen neuen Router anschliesst?
Routingtabelle wird aufgebaut, Austausch der Kosten untereinander, ermitteln der kürzesten Wege
Was macht ein Router bei eingehenden Requests?
Was enthält die Routingtabelle?

(nun kamen wir zu Stoff, der wohl nicht mehr zu 1801 gehört)
wie funktioniert ein RPC? Stichworte Call by Reference, call by Value usw. ;-)
Hier wurde er von der Beisitzerin aufmerksam gemacht, das wir uns bei 1801/1661 befinden :-)
Ein wenig konnte ich hier aus der täglichen Praxis jedoch wohl beisteuern.

dann Datenstrukturen:

Was ist ein Algorithmus? abstrakt formulierte Problemlösung ...
Laufzeitbestimmung eines Algorithmus ? Elementaroperationen, Komplexitätsklassen, Beachtung dominierender Ops ....man lese nach
O-Notation - betrachtet die Laufzeitentwicklung für n -> unendlich (hier habe ich schon ziemlich geschwommen - er hat Wert darauf gelegt, das dann konstante Aufwände entfallen könne, aber ...)
Welcher Algorithmus bei Sortierung von n=5 mit O(n^2) und O(nlogn) ??? Gut wäre gewesen, wenn ich geantwortet hätte, das dann die konstanten Aufwände bei kleinen n doch eine Rolle spielen und dann ggf. ein bei grossen n besserer Algorithmus nicht genommen wird (zur Verdeutlichung: man zeiche zwei Graphen der O-Funktionen, bis zum Schnittpunkt der lt. O-Notation schlechtere A. interessanter.

da wir dort lange diskutiert haben, ging es auch schon mit den Betriebssystemen weiter:

Was ist ein Betriebssystem? Benutzer- und Systemschnittstelle ...
Prozesssynchronisation, Hardwareverwaltung usw...
Wie sind die Zustände eines Prozesses? erläutern der einzelnen Zustände und Übergänge, habe bei den Übergängen und genauen Begriffen etwas "Nachhilfe" bekommen
Wie kommunizieren die Prozesse mit dem BS? Interrupts
welche gibts? Hard-/SoftwareInterrupts
Zwischendurch sind wir kurz abgeschwiffen auf DMA, wie ein Plattenzugriff vom Prozess angefordert wird, der DMA von der CPU angestoßen wird und der DMA-Controller sich wieder zurückmeldet.
Wie läuft so ein Interrupt genau ab? Woran erkennt man welchen I.? hier fehlte es bei mir vor allem am "genau", es hätte der Interruptvektor erläutert werden sollen, das der aktuell laufende Prozess gerettet wird war noch klar, aber wie erfährt das BS den aktuellen Befehl? Wer es genau wissen will, sollte im Script sich diese Seite genau zu Gemüte führen.
Dann gibt es noch LWP's? Was ist das? Habe LWP erläutert (eigene Befehlsstacks, geteilter Speicher usw.)
Wie verhindern, wenn mehrere Threads schreiben, dass diese sich gegenseitig überschreiben? Wenn wie aus der Pistole "Semaphor" gesagt worden wäre, alles i.O., aber das fiel mir erst nach einigem Hin- und Her ein.

Fertig, kurz raus zu seiner Sekretärin, die immer ein paar nette Worte über hatte.

Wieder zu Herr Haake, wo wir uns über die sehr gute Netzwerkprüfung und die nicht so guten Datenstruktur- und BS-Prüfungen unterhalten haben, alles in allem 2,7.
Die Bewertung ist fair, meine Nervosität bringt mir jedes mal diese nicht nachvollziehbaren und oben erwähnten Gedächtnislücken mit.
Das Protokoll ist natürlich nicht 100% vollständig, gibt hoffentlich MitstudentInnen eine Orientierung.

Viel Glück!

www.stud.fernuni-hagen.de/q6557341/

Protokoll zur mündlichen Prüfung für praktische Informatik (1801-1661) vom 22.9.03 bei Prof. Dr. Haake

Prüfer:	Prof. Dr. Haake
Termin:	22.09.03
Dauer:	ca. 30 min
Note:	1.0

Zunächst gab Prof. Haake eine kurze Einführung über den Ablauf der Prüfung:

Die Prüfung wird etwa 25-30 min dauern. Es werden die Themen Betriebssysteme, Netzwerke und Datenstrukturen geprüft. Für jedes Thema wird gleichviel Zeit aufgewandt. Falls eine Frage nicht beantwortet werden kann, so soll man nicht zögern, dies zu sagen. Die Fragen die gestellt werden haben unterschiedlichen Schwierigkeitsgrad, so dass die fehlende Antwort nicht immer gleich schlimm ins Gewicht fallen muss. Die Note setzt sich aus der Summe aller Antworten zusammen.

Datenstrukturen:

Frage: Was ist eine Datenstruktur?

(Hier habe ich ziemlich weit ausgeholt, und das Schaubild aus dem Kurs erklärt.) Es gibt eine mathematische, algorithmische und programmiertechnische Ebene. Mathematische Ebene ist die abstrakteste, die programmiertechnische die konkreteste, dazwischen die algorithmische. Innerhalb der Ebenen gibt es jeweils wieder ein konkretes und ein abstraktes Modell. Auf der algorithmischen Ebene dann den ADT und den Algorithmus.

Frage: Wie sieht so ein ADT aus?

Ein ADT ist auf der syntaktischen Ebene gekennzeichnet durch eine Signatur. Darin enthalten der Name des ADT, die sorts (Objekte des ADT- normalerweise Mengen) und die ops (Operationen, die auf den sets ausgeführt werden können?

Frage: Wie sieht das genau aus?

Es werden für jede Operation die Eingabe- und Ausgabeobjekte festgelegt.
Beispiel (contains: list x elem -> bool)

Frage: Wie erfolgt die Beschreibung auf semantischer Ebene?

Hier werden durch Axiome Eigenschaften der Operationen definiert. Unterschied zu Algebra ist, dass dort bei gleicher Signatur auf semantischer Ebene mit mathematischer Genauigkeit gearbeitet werden muss, während hier auf die Definition einzelner Eigenschaften beschränkt

werden kann. Dies bietet Vorteile bei der Implementierung, da dann mehr Spielraum für die Entwickler.

Frage: Welche Probleme können bei ADT auftreten?

Nicht alle ADT lassen sich leicht erstellen. Je nach Problemstellung kann es sogar ausgesprochen schwierig werden Definitionen zu finden. Bei steigender Komplexität ist wiederum die Implementierbarkeit schwierig, da dann ähnliche Probleme wie bei einer Algebra auftreten können.

Frage: Was kann noch für ein Problem auftreten?

Mehr fällt mir dazu leider auch nicht ein.

Antwort vom Prof.: Die Axiome müssen richtig sein. Wenn Sie über hundert Axiome haben, kann dies durchaus ein schwieriges Problem darstellen.

Frage: Was ist ein Algorithmus?

Ein Algorithmus ist ein Verfahren zur Lösung eines Problems. Er ist gekennzeichnet durch Einzelschritte, die genau beschreibbar sind, und jeweils in endlicher Zeit terminieren.

Frage: Entspricht ein Algorithmus einer Programmiersprache?

Nein. Es wird eine Mischung aus programmiersprachlichen Elementen und allgemeiner Sprache verwendet. Im Vordergrund steht die Verständlichkeit zur Erläuterung des jeweiligen Verfahrens.

Frage: Welche Eigenschaften kann ein Algorithmus haben?

Algorithmus hat bestimmte Laufzeit und Speicherbedarf.

Frage: Was noch?

Weiß nicht genau, worauf Sie hinauswollen?

Antwort vom Prof.: Ein Algorithmus muss auch leicht implementierbar sein.

Frage: Erläutern Sie das Verfahren zur Laufzeitbestimmung?

Es werden eine Reihe von Abstraktionsschritten vorgenommen. Zunächst wird der Algorithmus in Elementaroperationen eingeteilt. Dies sind arithmetische Operationen und Zuweisungen. Für diese wird ein Kostenwert von 1 angenommen. Dann werden Komplexitätsklassen gebildet.

Frage: Was sind Komplexitätsklassen genau?

(kleine Unsicherheit) Komplexitätsklassen sind Eingabemengen, die gleiches Laufzeitverhalten haben.

(Jetzt wollte ich weiter mit O-Notation.)

Frage: Welche Zwischenschritt fehlt noch?

(kleine Unsicherheit, mit Hilfe vom Prof.) Es wird eine Eingabemenge aus der jeweiligen Komplexitätsklasse ausgewählt, und für diese die Anzahl der Elementaroperationen festgelegt.

Frage: Wie geht es dann weiter?

(jetzt konnte ich endlich mit der O-Notation) Die Anzahl der Elementaroperationen stellt häufig eine Funktion in Abhängigkeit von der Eingabemenge dar. Von dieser Funktion kann man die konstanten Werte eliminieren. Setzt sich die Funktion aus mehreren Teilfunktionen zusammen, so wird nur die dominante Funktion betrachtet.
(Jetzt wollte ich noch die Definition der O-Notation genauer erläutern, dabei hat mich der Prof. dann aber unterbrochen.)

Frage: Was beschreibt die O-Notation?

Die O-Notation beschreibt eine obere Laufzeitgrenze für den Algorithmus.

Frage: Gibt es noch andere Möglichkeiten?

Man kann die untere Laufzeitgrenze eines Algorithmus beschreiben. Man bezeichnet diese als $\Omega(n)$ -Notation. Diese liegt vor, wenn gilt $f(n) \geq c \cdot g(n)$. Sind beide Notationen identisch, spricht man von einem asymptotisch optimalen Algorithmus.

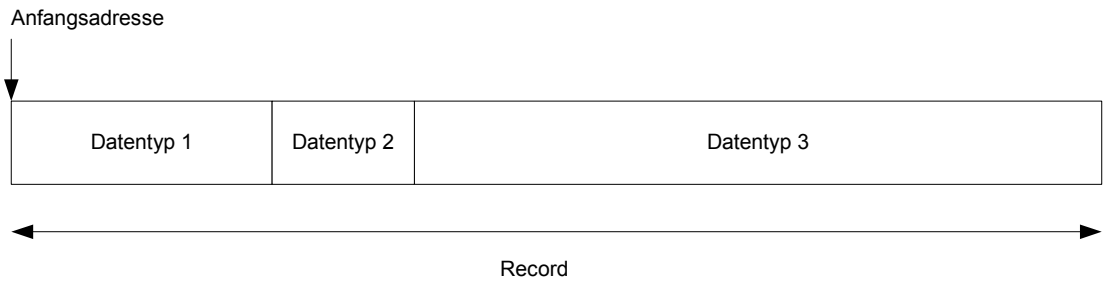
Frage: Welche zusammengesetzten Datentypen kennen Sie?

Array als Darstellung gleicher Datentypen, Record als Darstellung verschiedener Datentypen und Set als Unterbereichstyp.

Frage: Wie erfolgt die Darstellung eines Records im Hauptspeicher?

Es werden die Repräsentationen der einzelnen Datentypen im Hauptspeicher aneinandergefügt.

(habe dann noch folgendes Schaubild gezeichnet)



Frage: Wie kann man dynamische Strukturen aufbauen?

Hierfür ist der Datentyp des Zeigers notwendig. Zeiger speichern eine Adresse auf ein Objekt im Hauptspeicher. Sie bieten die Möglichkeit auf Datenstrukturen vom eigenen Typ zu zeigen. (Dereferenzierung) Es besteht daher die Möglichkeit die jeweiligen Instanzen der Struktur miteinander zu verbinden. (Linked, double-linke list, Baum)

Betriebssysteme

Frage: Was sind die Aufgaben des Betriebssystems?

CPU-Scheduling, Hauptspeicherverwaltung, Terminalverwaltung, Verwaltung von Peripheriegeräten ...

Frage: Erläutern Sie die Prozesszustände?

Es gibt erzeugt, bereit, rechnend, blockiert, beendet.
(Habe dann noch das Schaubild aus dem Kurstext mit den Zustandsübergängen gezeichnet. Wichtig, dass die Pfeile korrekt sind, da nur bestimmte Zustandsübergänge möglich sind)

Frage: Wie kann es zu Änderungen der Prozesszustände kommen?

Dies kann mit Hilfe von Interrupts geschehen. (siehe Kurstext was passiert wenn ein read-Befehl bearbeitet wird. Habe das Beispiel mit kleineren Lücken beschrieben. Zu jedem Interrupt gehört eine Interruptroutine die .... usw.)

Frage: Was sind Threads im Unterschied zu normalen Prozesse?

Ein Prozessraum hat unter anderem Datensegment und Programmsegment. Threads teilen sich diese in einem gemeinsamen Prozessraum. Sie haben jedoch eigene Register sowie Stacks.

Frage: Nennen Sie typische Anwendungen für Threads?

Typische Anwendung ist z.B. eine Client-Server Architektur. Hier bearbeitet der Server die Anfragen der Clients jeweils in eigenen Threads. Er kann dadurch mehrere Anfragen parallel bearbeiten.

Frage: Was passiert, wenn mehrere Threads schreibend und lesend auf eine gemeinsame Datei zugreifen?

Hier kann unbeabsichtigtes Fehlerverhalten auftreten, da nicht vorhersehbar ist, welcher Thread wann auf die Datei zugreift?

Frage: Wie kann man dies vermeiden?

Auf Programmebene durch Einsatz eines Monitors, auf Betriebssystemebene durch Einsatz von Semaphoren. Semaphor ist ein abstrakter Datentyp von Dijkstra, der eine Variablen w für die wartenden Prozesse und count für die verfügbaren Betriebsmittel enthält. Er lässt die Prozeduren up und down zu.

Frage: Erläutern Sie die Prozeduren?

down: wenn count>0 dann count-1, sonst w+1
up: wenn w>0 dann w-1, sonst count+1

Netzwerke:

(die Zeit wurde schon etwas knapp)

Frage: Erläutern Sie die Schichten des Internet mit einzelnen Protokollen?

(etwas Schlagwortartig)

Applikationsschicht: FTP, SMTP, HTTP sind zuverlässig nicht zeitsensitiv, Realtime-Protokolle sind fehlertolerant aber zeitsensitiv

Vermittlungsschicht: TCP: verbindungsorientiert und zuverlässig, (beachtet Bitfehler, Paketverlust und Überflutungsschutz)

Frage: Was fehlt?

Keine Ahnung (stressbedingte Aussetzer)

Antwort: die Reihenfolge der Pakete wird garantiert.

(ich dann weiter)
UDP: verbindungslos, hat Checksumme, keinen Schutz vor Überflutung, Paketverlust oder falscher Reihenfolge

Transportschicht: derzeit IPv4, sowie Routingprotokolle, es gibt globale und dezentrale Routingprotokolle

(eigentlich wollte ich das Gespräch auf die Routingalgorithmen lenken, aber leider..)

Frage: Wird eigentlich überall im Internet dasselbe Routingprotokoll eingesetzt?

(ganz große Unsicherheit)

Nun ja, eigentlich nicht. (dann der rettende Gedanke) Es gibt Intra-AS und Inter-AS. Innerhalb eines Intra-AS wird immer nur ein Protokoll eingesetzt, z.B. RIP. Zwischen AS wird ein Inter-AS Protokoll eingesetzt. Mir fiel nur noch ein, dass es mit B- anfängt. (gemeint war BGP-border gateway protocoll) Ich habe es erst fälschlich als Distanzvektor-Protokoll bezeichnet, auf Nachhaken vom Prof. bin ich dann aber doch noch darauf gekommen, dass es sich um ein Pfadvektor-Protokoll handelt, bei dem immer der Weg über die AS gespeichert wird.

Das wars, kurz draußen Warten.

Prof. Haake: Ich freue mich Ihnen mitteilen zu können, dass ich Ihnen ein sehr gut (1.0) geben kann. Alles Gute für den weiteren Fortgang des Studiums.

Fazit:
Die ganz guten Noten erwischt man sicherlich nur, wenn man selber einen guten Tag hat, und der Professor auch. Taktisch sinnvoll ist auf jeden Fall, sich bei den Fragen nicht nur auf die Minimalantwort zu beschränken, sondern auch immer noch ein paar Zusatzinformationen einfließen zu lassen. Wegen der Zeitknappheit ist es im Zweifelsfall immer besser zu sagen, dass man etwas nicht weiß, oder noch einmal nachzufragen, als irgendwie herumzueiern und so die Zeit zu vertun, die man eigentlich an anderer Stelle für das Zeigen von Wissen nutzen könnte.

Empfehlenswert ist vielleicht noch die zeitnahe Belegung des Kurses 1618 (OOP-Java). Dort werden auch Begriffe wie Socket, Client-Server, Monitor, Threads und Ports behandelt. Wenn man praktische Beispiele aus diesem Kurs in die Antworten mit einfließen lassen kann, so zeigt man fächerübergreifendes Verständnis und kann sehr konkret argumentieren.

Prüfung: Praktische Informatik B.Sc. (1661 & 1801)
Termin: 22.07.2003
Prüfer: Prof. Dr. Jörg M. Haake
Dauer: 25 min.
Note: 1,7

Nach der Vorstellung und der Frage, ob es sich um die erste Prüfung für mich handele und einigen einleitenden Informationen begann das Prüfungsgespräch.

Datenstrukturen:
Was ist ein Algorithmus?
- Die Grafik vom Deckblatt kam hier gut an.

Wie kann ich die Qualität eines Alg. messen?
- Laufzeit/Speicherbedarf (direktes Messen geht nur bei Implementierungen)
- O-Notation (und die Schritte dazu)

Was sind Datentypen?
Wie werden Datentypen notiert/beschrieben?
Standarddatentypen?
Wenn Standarddatentypen nicht ausreichen?
- Array, Record, Set

Einen davon beschreiben?
Wie wird die Adresse eines Elements eines Arrays ermittelt?

Betriebssysteme
Was sind die Bestandteile eines Prozessors?
Wie werden Befehle im Prozessor abgespeichert?
Mit welchem Datentyp kann der Hauptspeicher beschrieben werden?
Wie funktioniert der Prozesswechsel?
Worin liegt der Unterschied von Programmen und Prozessen?
Wie funktioniert die Kommunikation mit Geräten?

- Schichtenmodell
- Interrupts

Rechnernetze
Internet-Schichtenmodell?
Protokolle der einzelnen Schichten?
Wie funktioniert die Namensauflösung?
- DNS-Anfrage -> Root-Server -> Authorative DNS (und zurück)

Welche Unterschiede bestehen zwischen UPD und TCP?
Welches Transportprotokoll wäre bei einem Internet-Radio am sinnvollsten?

Insgesamt war es eine sachliche und angeneme Atmosphäre. Wichtiger sei
es, auch mal eine Frage auszulassen um damit mehr Zeit zu haben, andere
Fragen korrekt zu beantworten. Prof. Haake als Prüfer kann ich uneingeschränkt weiterempfehlen.

Hinweis: Ich hab' 'n sehr mieses Kurzzeitgedächtnis - mit Sicherheit gab's noch mehr Fragen - aber die wollen mir einfach nicht mehr einfallen.

Tipp zum Lernen: Mindmaps bauen - aber auch die Grafiken im Kurstext verstehen (die fehlen nämlich nachher in den Mindmaps...)

Mündliche Prüfung Praktische Informatik (Bachelor)
1661 Datenstrukturen I + 1801 Betriebssysteme / Rechnernetze

Prüfer: Prof. Haake
Termin: 23.04.2003
Dauer: 25 Min.
Note: 2,3

Am Anfang informierte mich Herr Haake darüber, daß die Prüfung drei gleiche Teile habe (Datenstrukturen – Betriebssysteme – Rechnernetze), daß er in dieser Reihenfolge prüfen wolle und jeder Teil ca. 8 Minuten dauern würde. Das hielt er auch ein, die Dauer kontrollierte er während der Prüfung durch wiederholte Blicke auf die Uhr.

Dann ging es gleich los, ohne große Vorrede. Ich war ziemlich nervös, etwas Smalltalk hätte mich wahrscheinlich beruhigt, aber dem war nicht so, die Nervosität legte sich erst langsam im Lauf der Prüfung.

Datenstrukturen

Was ist ein Algorithmus?

> Schon hier begann ich zu schwimmen, weil die Frage so allgemein gehalten war. Erwartet wurde etwa das: Realisiert eine mathematische Funktion (und spezifiziert eine in einer Programmiersprache geschriebene Prozedur oder Funktion) zur Lösung eines Problems.

Woraus besteht ein Algorithmus?

> Gute Frage. Nach drei Anläufen hatte ich alles zusammen: Verwendete Datentypen, Befehle und Kontrollstrukturen.

Qualitätskriterien für einen Algorithmus?

> Stehen im Skript: Korrektheit, Einfachheit des Verständnisses und der Implementierbarkeit, Ressourcenbedarf und Laufzeitverhalten..

Können alle Kriterien gleichzeitig erfüllt werden?

> Nein, vgl. Skript S. 6

Wie kann man die Laufzeit eines Algorithmus bestimmen?

> Zählen der Elementaroperationen, Aufstellen einer Laufzeitfunktion in Abhängigkeit von der Größe n der Eingabemenge.

Welches sind Elementaroperationen?

> Zuweisung, Vergleich, arithmetische Operation, Arrayzugriff.

Was tut man, wenn man die Elementaroperationen nicht zählen kann?

> Abschätzen, Vereinfachen der Laufzeitfunktion.

Welche Schritte sind dazu nötig?

> Jetzt hatte er mich. Ich wußte natürlich, daß das auf die O-Notation hinauslief, ich hätte auch jede konkrete Laufzeitfunktion in die O-Notation überführen können, aber wie beschreibt man das allgemein? Daß man Konstanten wegläßt, konnte ich noch sagen. Aber er bohrte immer weiter, brachte ein Beispiel: „Was ist, wenn Sie als Eingabemenge die natürlichen Zahlen haben? Und zum Beispiel eine Abfrage, bei der im linken Zweig die geraden und im rechten die ungeraden Zahlen durchlaufen werden?„ Ich konnte mit diesem Beispiel überhaupt nichts anfangen und mußte nach einigem Hin und Her kapitulieren. Was er hören wollte, war: daß man bei parallelen Zweigen nur den Zweig betrachtet, der die höhere obere Schranke aufweist, und daß man die Menge aller Eingaben in Komplexitätsklassen aufteilt. Daß ich dieses Wort nicht gesagt habe, war böse. Aber meine Schuld, steht im Skript.

Anderes Thema: Was sind Datentypen?

Datentyp = Algebra, wird syntaktisch beschrieben durch Signatur (Objektklassen und darauf definierten Operationen) und semantisch durch Angabe der Trägermengen und Funktionen = wie Werte der Operationen berechnet werden.

Aber da gibt es doch noch eine andere Art, Datentypen zu beschreiben ...

Abstrakte Datentypen, Axiome – siehe Skript S. 25 f.

Welche einfachen Standarddatentypen sind in gängigen Programmiersprachen implementiert?

Character, Integer, Boolean ...

Was kann man tun, wenn diese vordefinierten Typen nicht reichen?

Array – Record – Set (Set fiel mir nicht ein), Skript S. 42 – 48

Zum Array: wie findet man die Adresse eines Feldelements im Hauptspeicher?

Mit ein bißchen Hilfestellung: Basisadresse + (Elementnummer –1) * Elementlänge, vgl. S. 43

Betriebssysteme

Aufgaben eines Betriebssystems?

Prozeß- und Hauptspeicherverwaltung, Schnittstelle zu Hardware, Anwendungsprogrammen und Benutzer.

Welcher Datentyp eignet sich zur Beschreibung des Hauptspeichers?

Array von Wörtern

Schildern Sie einen Befehlszyklus!

Befehlszyklus? Was ist denn damit gemeint? Hilfestellung: Am Anfang steht die Adresse des Befehls im Befehlszählregister ... Ah ja: Übertragung ins Speicheradressregister, dessen Inhalt wird auf Adressbus gelegt, damit wird die entsprechende Zelle im Hauptspeicher angesprochen, deren Inhalt wird auf den Datenbus gelegt und kommt in der CPU im Speicherinhaltsregister an, wird von da ins Befehlsregister übertragen.

Unterschied zwischen Programm und Prozeß?

Prozeß = ausgeführtes Programm + Registerinhalte + Adressraum

Unterschied Prozeß – Thread?

Threads haben gemeinsamen Adressraum

Was haben Threads nicht gemeinsam?

Registerinhalte

Und?

Stack hätte ich hier sagen sollen

Rechnernetze

Beschreiben Sie das Internet-Schichtenmodell!

Anwendungsschicht – Transportschicht – Vermittlungsschicht – Sicherungsschicht – Bitübertragungsschicht

Welche Protokolle werden auf jeder Schicht verwendet?

HTTP/FTP/SMTP – TCP/UDP – IP – Ethernet

Welche Verbindung wird auf der Transportschicht hergestellt?

Zwischen zwei Hosts, äh, zwei Prozessen ...

Zwischen zwei Hosts oder zwischen zwei Prozessen?

Zwischen zwei Prozessen, weil im Segmentheader die Portnummern übertragen werden, und die identifizieren die Prozesse.

Und auf der Ebene der Netzwerkschicht?

Da werden zwei Hosts verbunden.

Welche Dienste stellt TCP zur Verfügung?

TCP garantiert Vollständigkeit der übertragenen Nachrichten und richtige Reihenfolge, bietet Fehlerkorrektur und Flusskontrolle (letzteres ist mir nicht eingefallen, hätte ich aber erklären können)

Welchen Fortschritt bietet TCP im Vergleich zum Alternating-Bit-Protokoll?

Größere Effizienz durch aufeinanderfolgende Übertragung mehrerer Segmente und nicht Bestätigung einzelner Segmente, sondern kumulative Bestätigung.

Angenommen, ein TCP-Sender hat Platz für 4 Segmente im Puffer. 3 wurden gesendet, ein Platz ist frei. Nun trifft eine Bestätigung für das zweite Segment ein. Wieviel Platz steht daraufhin im Puffer zur Verfügung?

3, da die ersten beiden Segmente erfolgreich gesendet wurden und aus dem Puffer entfernt werden können

Wenn Sie eine Telefonie-Anwendung implementieren wollen, welches Transportprotokoll würden Sie benutzen?

UDP, da es bei Telefonie mehr auf Übertragungsgeschwindkeit ankommt und völlige Fehlerfreiheit nicht erforderlich ist.

Nachbesprechung

Zwei Minuten mußte ich draußen warten, dann wurde ich wieder hereingerufen und konnte mir die Bewertung anhören. Datenstrukturen: 3, vor allem wegen der Schwächen bei der O-Notation. Betriebssysteme: 2,3, mit dem Argument: „Bei anderen, die besser bewertet wurden, konnte ich mehr Fragen stellen.„, d.h. ich habe zu viel herumgestopselt. Rechnernetze: 1,3. Macht im Durchschnitt 2,2 – aufgerundet 2,3. Ich habe mir zwar im vorhinein ein besseres Ergebnis gewünscht, hätte mir aber auch eine 2 minus gegeben. Besser war ich nicht. Von daher fand ich die Beurteilung absolut gerecht.

Fazit

Da mir noch kein Prüfungsprotokoll zu Prof. Haake vorlag, wußte ich nicht genau, was mich erwartete. Bei der Prüfungsanmeldung hatte mir Herr Haake am Telefon gesagt: „Es reicht, wenn Sie den Kurstext verstanden haben.„ Wie dieser Satz gemeint war, habe ich erst während der Prüfung begriffen: Einerseits brauchte man nicht – wie offenbar bei Prof. Güting – Formeln aus dem Ärmel zu schütteln (z.B. zur O-Notation, zu Sortieralgorithmen, zu Semaphor-Methoden). Solcher Art Detailwissen war nicht verlangt. Es lag zwar Papier und Stift auf dem Tisch, aber ich brauchte nichts aufzuschreiben. Andererseits mußte man durchaus in der Lage sein zu reproduzieren, aber eher auf allgemeiner Ebene. Auf Fragen wie „Welche Dienste bietet TCP an?„ muß man genau 4 Punkte aufzählen. Wenn man einen vergißt und ihn auch auf Nachfrage nicht nennen kann, führt das unweigerlich zum Punktabzug.

Herr Haake ist freundlich und fair, überhaupt nicht hinterhältig, aber etwas trocken und von der Sorte „streng, aber gerecht„. Man bekommt nichts geschenkt, kann aber durchaus eine Eins schaffen – wenn man den Kurstext tatsächlich verstanden hat UND sich einen hierarchischen Überblick über den Stoff verschafft hat UND auf allgemeiner Ebene die wichtigsten Aufzählungen, Unterscheidungen, Definitionen, Untergliederungen etc. reproduzieren kann.

Prüfung: Praktische Informatik Bsc - 01661/01801
Prüfer: Professor Haake
Termin: 31.03.2003
Dauer: 25 min
Art: Videoprüfung

Themen:
Datenstrukturen
* Algorithmus
* ADT
* Abstraktionsschritte
* Grundtypen
 (ich konnte mir raussuchen welchen Typ Array, Record oder Set ich näher erklären will)

Betriebssysteme
* Hauptspeicherzugriff
* Unterschied Programm - Prozess
Hier wollte er Prozesskontext hören
* Threads

Rechnernetze
* IP-Schichtenmodell
* TCP - UDP (bei TCP ging er dann etwas in die Tiefe, Fensterprotokoll etc.)

Herr Haake wollte hauptsächlich das Grundlegende Wissen überpfrüfen, so daß ihm die wichtigsten Stichpunkte reichten.
Eine Videoprüfung ist schon sehr eigenartig, aber nach den ersten Minuten wird man schon etwas lockerer jedoch passieren immer mal wieder Störungen, so musste ich zum Beispiel am Ende meine
Prüfung per Telefon beenden.

--
EMD

Homepage: http://www.evamd.de.tf
E-Mail: EvaMD@gmx.de
Forum: http://9806.rapidforum.com

+++ GMX - Mail, Messaging & more http://www.gmx.net +++
Bitte lächeln! Fotogalerie online mit GMX ohne eigene Homepage!

Prüfung: Fachprüfung Praktische Informatik
Kurse: 01661 + 01881
Prüfer: Prof. Haake
Dauer: 30 Min.

Datenstrukturen und Algorithmen:

1.) Was ist ein Algorithmus?

2.) Wie wird die Qualität eines Algorithmus bestimmt?

3.) Wie wird die Laufzeit bestimmt?
[Hier solltest Du in der Lage sein, die Abstraktionsschritte sowie Details der O-Notation beschreiben zu können]

4.) Wie ist die Komplexität zum Suchen auf einer Menge?
[O(log n)]

5.) Und wie erreicht man das?
[z.B. binäre Suchbäume und AVL-Bäume]

6.) Erklären Sie die Datenstrukturen.

7.) Geht das nicht auch in konstanter Zeit?
[Ja, z.B. mit einem Bitvektor oder Hashing]

8.) Erklären Sie das Hashing.
[Hier solltest Du insbesondere erklären können, wie eine Laufzeit von O(1) möglich ist.]

Betriebssysteme und Rechnernetze:

9.) Was ist der Unterschied zwischen einem Programm und einem Prozess?

10.) Was gehört alles zu dem Prozesskontext?

11.) Können Teile des Prozesskontextes auch geteilt werden?
[Ja, bei Verwendung von Threads.]

12.) Was teilen sich denn Threads?

13.) Wer unterbricht einen Prozess?
[Der Zeitgeber. Er wird vom Scheduler gestartet und unterbricht ggf. den Prozess.]

14.) Was für eine Unterbrechung ist das?
[Eine Hardwareunterbrechung (der Zeitgeber ist eine Hardware!)]

15.) Wie funktionieren Unterbrechungen?

16.) Welche Unterbrechungen gibt es?

17.) Welche Scheduling Strategie würden Sie heutzutage verweden?
[Multitasking mit Round-Robin]

18.) Wie funktioniert Round-Robin?

19.) Beschreiben Sie die Internetschichten mit Namen, Protokollen und wer mit wem kommuniziert.

20.) Warum könnte ein Router ein Paket verwerfen?
[Gefragt war hier inbesondere der Ablauf der TTL]

21.) Was ist der Unterschied zwischen TCP und UDP?

22.) Was tun Sie, wenn Sie die Vorteile beider Protokolle nutzen wollen?
[Ich nehme UDP und implementiere die restlichen Funktionalitäten auf der Anwendungsschicht. (Anm.: Du solltet auch wissen welche Funktionalitäten das sind!)]

23.) Welches Prüfverfahren verwendet IP?
[Ein Checksummenverfahren]

Ich kann mich meinen Vorgängern nur anschießen, dass Prof. Haake ein ganz ausgezeichneter Prüfer ist! Er versucht eine freundliche Atmosphäre zu schaffen und setzt seinen Prüfling nicht unter Druck. Weiß man eine Antwort nicht auf Anhieb, gibt er eine Hilfestellung. Trifft man bei mehreren richtigen Antworten nicht gleich die von Ihm Gewünschte, so führt er einen langsam an die Antwort heran.
Bei der Note spielen vermutlich sowohl Qualität (der Antworten) als auch Quantität (der beantworteten Fragen) eine Rolle. Ich kann nur wiederholen: Wer gut gelernt hat, bekommt auch eine gute Note. Kleine Patzer in den Antworten wurden mir verziehen.

Viel Erfolg!