Prüfungsprotokoll zur Diplomprüfung

Praktische Informatik

Prüfungsgebiete:

- Modelle der Datenexploration (Seidl, WS 04/05)
- Betriebssysteme I (nach Buch von Tanenbaum (2. Auflage))
- Datenkommunikation
- Verteilte Systeme

Prüfer:	Prof. Dr. Thomas Seidl & Prof. Dr. Otto Spaniol
Beisitzer:	kein weiterer
Datum:	16.03.2006 (09:00 Uhr)
Dauer:	knapp über 45 min.
Note:	1.3

Modelle der Datenexploration

Seidl: Für was haben wir denn alles Ähnlichkeitsmodelle verwendet?
I: Für Bilder, Formen, Sequenzen und Graphen.
Seidl: Ok, wie macht man das denn ganz einfach bei Bildern?
I: *Zunächst hab ich erklärt, dass man Features extrahieren muss, damit man schließlich Vektoren hat, die man vergleichen kann. Habe weiter erwähnt, dass man beispielsweise im RGB-Farbraum je 2^n Features wählen könnte, also z. B.* $4 \times 4 \times 4 = 64$ *Features.*
Seidl: Und womit vergleicht man diese Vektoren? Was wäre zunächst das nahelegenste?
I: Die euklidische Distanz. Aber die hat Nachteile...
Seidl: Und zwar?
I: Die einzelnen Features werden nicht untereinander verglichen, haben jedoch unterschiedliche Abstände im Farbraum. Daher nimmt man Quadratische Formen.
Seidl: Ja, schreiben sie die mal hin...
I: *Hab die hingeschrieben und mich dabei peinlicherweise gleich mal vertan (Summen- und Vektorschreibweise vertauscht). Nach dem Prof. Seidl mich darauf hingewiesen hab, hab ich das dann schließlich korrigiert.*
Spaniol: *Jetzt hat sich Prof. Spaniol kurz mal eingeschaltet.* Und was ist daran jetzt quadratisch?
I: Die Quadratische Form ist eine verallgemeinerte Euklidische Distanz. Für den Spezialfall, dass man für A die Einheitsmatrix nimmt, erhält man gerade die Euklidische Distanz.
Seidl: Und was ist daran jetzt besser als bei der Euklidischen Distanz?

I: *Ich konnte ihm nicht mehr sagen, als dass die einzelnen Komponenten auch untereinander verglichen werden. Aber mit dieser Antwort schien er etwas unzufrieden. Vielleicht wollte er das noch etwas prägnanter und formaler hören.*
Seidl: Ok, was gibt es denn für Anfragetypen?
I: ϵ-Anfragen,...
Seidl: Wie sieht so eine ϵ-Anfrage aus?
I: Meinen sie formal die Ergebnismenge?
Seidl: Ja.
I: *Hingeschrieben:* $\mathrm{sim}_\epsilon(q) = \{o \in \mathrm{DB} \mid \mathrm{d}(o, q) \leq \epsilon\}$
Seidl: Und diese Anfrage ist wovon abhängig?
I: Ja, von ϵ und q. *(kopfkratz)*
Seidl: Ja, und was gibt es für weitere Anfragetypen?
I: Nächste-Nachbar-, k-nächste-Nachbar und Rankinganfragen. *Hab auch erklärt, was das ist.*
Seidl: Wie macht man jetzt solche Anfragen? Die Anfrageobjekte sind ja irgendwie auf der Festplatte gespeichert...
I: Sie wollen jetzt wahrscheinlich auf einen Index hinaus.
Seidl: Na ja, das braucht man ja nicht unbedingt. Was könnte man auch machen?
I: Sequentieller Scan. Dann hat man aber $O(n)$ Zugriffe.
Seidl: Genau, und weil der nicht so toll ist, will man eigentlich doch einen Index. Den R-Baum haben wir in der Vorlesung glaube ich behandelt. Können sie mal beispielhaft erklären, wie im R-Baum Knoten gespeichert sind?
I: *Hingemalt und erklärt, wie im Skript.*
Seidl: Und was hat man da jetzt für eine Komplexität?
I: $O(\log n)$.
Seidl: Ist das bei Bäumen immer so? Was ist im R-Baum $O(\log n)$?
I: *Überlege.* Die Länge eines Pfads von der Wurzel zu den Blättern bei einer Gesamtzahl von n Knoten im Baum.
Seidl: Was muss für den Baum gelten, damit das gilt?
I: ...
Seidl: Der Baum darf nicht entartet sein. Kann denn ein R-Baum entarten?
I: Hmm, mir fällt leider gerade kein Grund ein, warum das nicht passieren kann... Wenn man immer Punkte an der selben Stelle einfügt...
Seidl: *Prof. Seidl hat erklärt, dass man das vermeiden kann. Er meinte aber auch, dass er das in der Vorlesung vielleicht nicht unbedingt erwähnt hat.* Aber es gibt noch eine andere Entartung bei R-Bäumen, die auftritt...
I: Ja, bei hohen Dimensionen tendieren die Hyperrechtecke dazu, sich zu überlappen.
Seidl: Warum ist das schlimm, wenn sich die Rechtecke überlappen?
I: *Hab mir noch irgendwas zurechtgestammelt, etwa dass die Rechtecke dann alle ähnliche Abstände zum Anfragepunkt haben. Dass Prof. Seidl damit unzufrieden schien, könnte aber auch mit meiner miserablen Darstellung zu tun gehabt haben.*
Seidl: Was kann man tun, wenn man einen Index mit weniger Dimensionen haben will?
I: Dimensionsreduktion.
Seidl: Das macht man wie?

I: Da haben wir in der Vorlesung vieles gemacht. Beispielsweise Diskrete Fourier Transformation oder Wavelettransformation… *Hab dann mal die DFT erklärt.*
Seidl: Kann man die DFT auch sinnvoll auf DNA-Sequenzen anwenden?
I: *Nach rumstammeln hab ich dann gesagt, dass das man dort vielleicht nicht unbedingt die feinen Details wegwerfen will. Weiß aber nicht, ob das richtig ist.*
Seidl: Es gibt ja noch die Hauptkomponentenanalyse. Was schmeißt man denn da weg?
I: *Irgendwie PCA erklärt. Prof. Seidl guckt etwas skeptisch.* Wenn sie dazu noch Details wissen wollen, dürfen sie aber ruhig nachfragen.
Seidl: Ne, ich muss ja noch Betriebssysteme Prüfen.

Betriebssysteme

Seidl: Was ist ein Prozess?
I: Programm in Ausführung mit Registerinhalten, verwendeter Speicher, geöffnete Dateien. Ein Prozess kann mehrere Zustände annehmen…
Seidl: Ja, aber was hat das Betriebssystemen mit den Prozessen zu tun?
I: Ein Betriebssystem hat hauptsächlich zwei Aufgaben: Maschine mit veredelter Hardware und Ressourcenverwaltung. Das Betriebssystem verwaltet die Prozesse.
Seidl: Muss denn ein Prozess von dem Betriebssystem wissen?
I: Nun, ein Prozess will ja auf die Ressourcen zugreifen, die das Betriebssystem verwaltet. Es muss also die Systemaufrufe für das Betriebssystem kennen.
Seidl: Mit der Ressourcenverwaltung gibt es ja auch so seine Probleme. Woran denken sie da?
I: Deadlocks.
Seidl: Was ist das?
I: Eine Gruppe von Prozessen befindet sich in einem Deadlock, wenn jeder Prozess dieser Gruppe auf ein Ereignis wartet, dass nur ein anderer Prozess dieser Gruppe auslösen kann.
Seidl: Und was kann man dagegen tun?
I: Nun, also erstmal könnte man gar nichts tun, da Deadlocks relativ selten auftreten. So wird das z. B. bei Linux oder Windows gemacht. *Das fand Prof. Seidl nicht so passend formuliert (siehe unten). Dann habe ich die weiteren Möglichkeiten erklärt, in der Reihenfolge, wie sie im Tanenbaum stehen. Als ich schließlich auf den Angriff auf den wechselseitigen Ausschluss und Spooling zu sprechen kam, hat er mich unterbrochen.*
Seidl: Sehen sie, man tut also auch in der Praxis durchaus im Vorfeld eine ganze Menge, um Deadlocks zu vermeiden. Wie sieht es denn bei dem Arbeitsspeicher aus? Wie vermeidet man da Deadlocks?
I: Das hängt natürlich vom Speicherverwaltungsmodell ab… aber in der Regel verwendet man heutzutage Paging.
Seidl: Können sie Paging erklären?
I: Hmm, da weiß ich jetzt ja gar nicht, wo ich anfangen soll… Nun, also erstmal hat man einen virtuellen Adressraum… *Ich kam dann irgendwie zur Seitentabelle und hab erklärt, wie Seiten auf Seitenrahmen abgebildet werden. Insgesamt hab ich das aber wenig souverän runtergestammelt.*
Seidl: Und wann werden jetzt Seitenrahmen ausgelagert?

I: Das Betriebssystem prüft, dass immer genug freie Seitenrahmen zur Verfügung stehen.
Seidl: Also wann ist das nicht mehr der Fall?
I: Wenn der Speicher voll ist.
Seidl: Na ja, wenn der Speicher voll ist, ist ja noch alles in Ordnung. Ausgelagert wird, wenn der Speicher voll ist und noch ein zusätzlicher Seitenrahmen benötigt wird. *Auch wenn Prof. Seidl darauf hinauswollte, bin ich mir nicht sicher, ob das so stimmt. Ich meine mal gelesen zu haben, dass ein 'Paging Daemon' im Hintergrund dafür sorgt, dass immer freie Seitenrahmen zur Verfügung stehen.* Nun, und die einzelnen Strategien für die Seitenersetzung kennen sie ja sicherlich auch alle. Gut, dass reicht mir dann. *Zu Prof. Spaniol:* Machen sie weiter?

Datenkommunikation & Verteilte Systeme

Spaniol: Sie kennen ja sicherlich IP. Haben sie Geräte, die eine IP-Adresse besitzen?
I: Ja, mein Rechner hat natürlich eine.
Spaniol: Kennen sie die auswendig?
I: 134.130.48.23
Spaniol: Das ist ja schon eine spezielle Darstellung, und zwar?
I: Dezimaldarstellung?
Spaniol: Nein.
I: ...
Spaniol: Also in ihrer Darstellung können z. B. nicht alle Ziffern vorkommen, z. B. kann dort keine 9 vorkommen. Das ist die Oktaldarstellung. *Ich halte das für falsch. Wahrscheinlich hat Prof. Spaniol irgendwas ganz komisches verstanden, als ich meine IP-Adresse gesagt habe, oder aber ich habe Prof. Spaniol während der Prüfung falsch verstanden. Ich hatte allerdings wenig Chance da noch einzuhaken, da er sofort weitergeredet hat.* Das ist also eine für den Menschen angenehme Darstellung. Kennen sie eine Verzeichnis, in dem man IP-Adressen unter für uns Menschen sinnvollen Namen nachschlagen kann? *Prof. Spaniols Fragestellung war glaube ich sogar noch verwirrender, so dass ich zunächst nicht drauf gekommen bin.*
I: DNS
Spaniol: Steht für was?
I: Domain Name Service.
Spaniol: Wieviel IP-Adressen gibt es, theoretisch?
I: Moment... eine IP-Adresse ist 32 Bit lang, also theoretisch...
Spaniol: Genau, 2^{32}. Ich habe während Prüfungen schon die wildesten Spekulationen gehört, wieviel 2^{32} denn sei. Irgendwann kommen die Studenten dann meist darauf, dass das mehr als 100 ist. Was meinen sie denn, wieviel das ist?
I: Also auf jeden Fall weit über eine Millionen.
Spaniol: Das ist richtig. Aber überlegen sie mal... wieviel sind denn 2^{30}?
I: Öhm...
Spaniol: Oder erstmal 2^{10}. Das sollte man als Informatiker kennen.
I: 1024.
Spaniol: Ja, also sagen wir mal grob 1000. Was sind dann 2^{30}? Jetzt müssen sie Potenzrechnung können.

I: Das ist dann... mal 3, nein... Moment...
Spaniol: Hoch 3, also 1000 mal 1000 mal 1000, also eine Milliarde. Wobei man das nicht mit dem Amerikanischen verwechseln darf, die nennen das 'one billion'. So, und wieviel sind jetzt 2^{32}?
I: Das ist dann 2^{30} mal 2^2,... also nochmal mal 2 *Nein!*
Spaniol: Mal 4. Also etwa 4 Milliarden IP-Adressen. Nun, haben die eine Struktur? Kann man z. B. anhand der IP-Adresse sehen, wo sich ein Rechner befindet?
I: Ja, das geht.
Spaniol: Richtig. Woraus setzt sich denn eine IP-Adresse zusammen?
I: Der erste Teil adressiert das Netzwerk. Man unterscheidet dabei verschiedene Klassen je nachdem, wieviel Stellen das für die Adressierung des Netzwerks verwendet werden.
Spaniol: Ja, genau. IP-Adressen besitzen also eine hierarische Struktur. Und der zweite Teil der IP-Adresse, was adressiert der?
I: Die einzelnen Rechner in dem Netzwerk.
Spaniol: Genau. IP-Adressen sind ja knapp. Können denn alle IP-Adressen genutzt werden?
I: Nein, aus verschiedenen Gründen. Zum einen sind bestimmte Adressen für spezielle Zwecke reserviert, z. B. die Klasse D Adressen für Multicast. Zum anderen haben sind aufgrund der Strukturierung viele Adressen nur für sehr kleine Klasse C Netze nutzbar...
Spaniol: Außerdem wurden IP-Adressen in der Anfangszeit des Internets sehr verschwenderisch vergeben. Der Bundesstaat Massachusetts *(ich meine den hat er genannt)* hat z. B. mehr IP-Adressen als ganz China... Nun gut, es gibt aber auch nur etwa 4 Milliarden IP-Adressen. Das sind etwas weniger als es Menschen gibt. Was kann man jetzt dagegen tun?
I: Zum einen könnte man natürlich IPv6 einführen.
Spaniol: Ja, wie groß ist der Adressraum da?
I: IPv6 hat 128 Bit Adressen, macht also 2^{128} Adressen.
Spaniol: Ja, mit IPv6 könnte man viele tolle Sachen machen. Aber warum ist das so schwer das IPv4-Protokoll zu ersetzen? Warum können sich nicht einfach zwei Leute sagen, wir benutzen jetzt IPv6?
I: Weil dann auch alle Router, über die die Pakete laufen, IPv6 verstehen müssen.
Spaniol: Ja. Man müsste die ganze Welt an einem Tag umstellen. Das geht natürlich nicht. Man braucht also eine Migrationsstrategie. Wie könnte die aussehen?
I: Also, natürlich müsste man erstmal beide 'Protocol Stacks' in den Routern implementieren... Zum anderen könnte man IPv4 Router überbrücken, indem man IPv6-Pakete tunnelt.
Spaniol: Tunneln, das heißt?
I: Router packen IPv6-Pakete in IPv4-Pakete ein, um sie auch über IPv4-Router weiterleiten zu können.
Spaniol: Genau, man schreibt also einfach noch einen IPv4-Header davor und schickt die Pakete dann weiter. *Prof. Spaniol hat noch ein paar Gründe eingeworfen, warum IPv6-Geräte so rar sind.* Also, was kann man denn ansonsten machen, wenn man kein IPv6 hat?
I: Network Address Translation bzw. Network Address Port Translation.

Spaniol: Genau, das funktioniert wie?
I: Also angenommen, eine Firma hat nur einige wenige IP-Adressen…
Spaniol: Nicht einige wenige. Sagen wir nur genau eine.
I: Ok, dann geht das nur über die Ports. *Ich hab noch ein wenig weiter erklärt.*
Spaniol: Genau, wir erweitern also den Adressraum durch Hinzunahme der Ports um zusätzliche 16 Bit. Was sind denn die Vor- und Nachteile von NAT?
I: Also an Vorteilen, fällt mir jetzt die höhere Sicherheit ein, da die hinter dem NAT-Router liegende Netzwerkstruktur nach außen versteckt wird.
Spaniol: Ja. Man erhält eine Firewall-Funktionalität. Das ist von Unternehmen natürlich durchaus so gewünscht. Und die Nachteile?
I: Nun, also erstmal muss natürlich der Router auch auf Schicht 4 arbeiten. Das könnte sich schlecht auf die Performance auswirken…
Spaniol: Na gut, aber das ist eher ein kosmetisches Problem. Die Schichten werden vermischt…
I: Dann man kann die Rechner hinter dem NAT-Router nicht direkt adressieren. Das muss man aber z. B. wenn man einen Server hinter einem NAT-Router betreiben will…
Spaniol: Ja, aber das ist in so einem Unternehmen ja durchaus noch so erwünscht. Ein Problem ist, dass der gesamte Netzverkehr über diesen NAT-Router laufen muss, dieser also zum Bottleneck wird. Was könnte man dagegen machen?
I: Redundanz. Noch einen zweiten bereitstellen…
Spaniol: Ja, das könnte man machen. Und wenn der eine Router ausfällt, wird der andere geschaltet. Was entsprechen IP und TCP für OSI-Schichten?
I: IP entspricht Schicht 3, TCP Schicht 4.
Spaniol: Wofür steht TCP?
I: Transmission Control Protocol.
Spaniol: Wozu braucht man das überhaupt, wenn man doch schon mit IP Rechner weltweit adressieren kann?
I: Mit IP kann man nur einzelne Pakete durchs Netz verschicken. TCP bietet eine netzunabhängige Ende-zu-Ende Betrachtung des Datenstroms.
Spaniol: TCP ist im Gegensatz zu IP also verbindungs…
I: …orientiert.
Spaniol: Um diese Verbindungsorientiertheit zu erreichen, muss man ja irgendwie verhindern, dass zuviele IP-Pakete geschickt werden, die dann verworfen werden. Wie verhindert TCP das?
I: TCP hat eine Überlastkontrolle. Benutzt wird der sogenannte Slow Start Algorithmus. Man fängt mit einem Segment an und dann verdoppelt man immer die Anzahl der Segmente, die man aufeinmal sendet…
Spaniol: Genau, man verdoppelt. Die Transferrate wächst also exponentiell…
I: …bis zu einer bestimmten Grenze…
Spaniol: …und von da an nur noch linear. Was glauben sie, wo der der Slow Start Algorithmus schlecht funktioniert?
I: Hmm…
Spaniol: In Funknetzen. Was meinen sie warum?

I: In verdrahteten Netzen pendelt sich eher ein konstanter Wert ein. Geht ein Paket verloren, kann man davon ausgehen, dass das Netz überlastet ist. Bei Funknetzen können Pakete auch häufig nur durch Störungen verloren gehen, auch wenn das Netz noch nicht ausgelastet ist. Die Übertragungsqualität schwankt stark je nach Wetter und anderen Einflüssen.
Spaniol: TCP ist ja verbindungsorientiert. Kennen sie auch ein verbindungsloses Protokoll auf Schicht 4?
I: UDP
Spaniol: Steht für was?
I: User Datagram Protocol.
Spaniol: Für welche Anwendungen verwendet man das beispielsweise?
I: Da fallen mir spontan Routing-Protokolle ein.
Spaniol: Ich meinte Anwendungen. Also z. B. Telefon- und Videodaten in Echtzeit. Warum nimmt man dafür kein TCP? Ein zuverlässiger Datentransport ist da doch auch durchaus wünschenswert.
I: Aber ein nachgereichtes verlorenes Datenpaket würde einem bei Echtzeitanwendungen nichts mehr bringen. Dann ist es zu spät.
Spaniol: Ja, man ist auf niedrigen Delay angewiesen und will beim Telefonieren keine Verzögerung von ein oder zwei Sekunden in der Leitung haben. Was könnte man denn dafür tun, um Verbindungsorientiertheit zu erreichen?
I: Buffering oder Traffic Shaping, wie z. B. Leaky Bucket, Token Bucket,...
Spaniol: Na ja, aber dadurch erhält man keine Verbindungsorientiertheit. Man könnte Pakete markieren, so dass sie immer den gleichen Weg durchs Netz nehmen können.
I: MPLS.
Spaniol: Heißt was?
I: Multi-Protocol Label Switching.
Spaniol: Ja, man schreibt einfach ein Label vor den IP-Header...
I: ...und die weiteren Pakete folgen dann auf dem selben Weg.
Spaniol: Ist das Label denn immer bei jedem Router das gleiche?
I: Öhm, ich glaube das ist in einem Teilnetz schon so...
Spaniol: Nein, die Pakete werden bei jedem Router umgelabelt. Sonst könnten Labels auch mehrfach vergeben werden. So kann das nicht passieren. So, jetzt müssen wir noch langsam auf die verteilten Systeme zu sprechen kommen. Was definieren OSI-Schichten 5 und 6?
I: Schicht 5 ist die Sitzungsschicht: Synchronisation, Token-Vergabe, Wiedereinstiegspunkte,...
Spaniol: Und Schicht 6?
I: Darstellungsschicht: Aushandlung der Transfersyntax,...
Spaniol: ...Darstellung von Zahlen und Zeichen,...
I: ...sowie Datenkompression und Verschlüsselung.
Spaniol: Und womit simuliert man diese Schichten in verteilten Systemen?
I: Middleware.
Spaniol: Nennen sie mal eine.
I: CORBA

Spaniol: Steht wofür?
I: Common Object Request...
Spaniol: ... Broker Architecture. *Prof. Spaniol hat dann noch zwei Sätze dazu gesagt.* Nehmen wir mal an, wir habe n verschiedene Sprachen, die man jeweils in alle anderen Sprachen übersetzen will. Wieviele Wörterbücher braucht man dann?
I: Hui... Sind das nicht n^n. *Da habe ich wohl sehr unüberlegt geantwortet.*
Spaniol: Um Gottes Willen. Also, sagen wir mal, wir haben 10 Sprachen.
I: *... Ich habe nur daran gedacht, dass ich jetzt eh nicht geügend Zeit habe, darüber nachzudenken. Dabei macht mir Kombinatorik doch eigentlich richtig Spaß.* Hmm, das kann eigentlich nicht so schwer sein?
Spaniol: Nö, ist es auch nicht... Nun ja, $\binom{n}{2}$. Aber wie kann man es besser machen?
I: Man einigt sich auf eine Sprache, die zur Übersetzung verwendet wird...
Spaniol: Beispielsweise auf Englisch...
I: Man übersetzt zunächst nach Englisch und dann von Englisch in die gewünschte Sprache.
Spaniol: Wieviele Wörterbücher braucht man dann?
I: $2n$.
Spaniol: $\frac{2n}{2}$, also n reichen auch. *Bei Prof. Spaniol kann man also mit Wörterbüchern jeweils in beide Richtungen übersetzen. Den tieferen Grund für diese Sichtweise kenne ich aber nicht. Ich glaube, er hat in der Prüfung dazu nichts weiter gesagt.* Können sie die Begriffe 'Stub' und 'Marshalling' erklären?
I: Ja, die Stubs hat man beim Remote Procedure Call, RPC, oder auch bei RMI. Wenn ein Client in einem verteilten System Daten anfordert, führt er einen für ihn transparenten Systemaufruf auf dem Stub auf. Der Client Stub stellt dann fest, ob die Daten lokal vorhanden sind, und wenn nicht, werden sie quasi per 'Marshalling' an den Server Stub übergeben... *Keine Garantie, dass das so ganz formal korrekt ist.*
Spaniol: Was sagt ihnen IDL?
I: Interface Definition Language...
Spaniol: *Eh ich anfangen konnte zu erklären, hat Prof. Spaniol schon gesagt, wofür man so eine Sprache benutzt.*
I: Das ist bei CORBA ganz toll, da dort aus dem IDL-Code zunächst nur maschinenunabhängiger Code in höheren Programmiersprachen erzeugt wird, der dann erst auf dem Zielsystem übersetzt wird. *Das konnte ich auch nicht mehr ganz aussprechen, denn dann...*
Spaniol: So, für mich ist die Prüfung damit jetzt zuende.

Anmerkungen:
Ich habe *nicht* direkt nach der Prüfung meine Note erhalten. Nach mir kam noch ein Kommilitone mit einer ähnlichen Prüfungskombination dran (Datenexploration, Datenkommunikation, MoBKo). Die Noten haben wir dann zusammen erhalten. Er hatte eine 1.0, ich eine 1.3 bekommen. Die Prüfer meinten aber, dass meine Note so an der Grenze zur 1.7 sei. Sie meinten weiter, dass die Fachgruppe Informatik Kombi-Prüfungen ja fördern wolle. Mit der Note bin ich mehr als glücklich. Ich hätte nach der Prüfung eher 1.7, vielleicht sogar 2.0 erwartet. Gelernt habe ich für die Prüfung knapp 6 Wochen. Das war zwar knapp, hat aber gereicht. Wenn man im Stoff weniger gut drin ist, sollte man vielleicht eher mehr Zeit einplanen.
Prof. Seidl stellt eigentlich fast immer die selben Fragen (vergleiche andere Prüfungsprotokolle). Eine Ausnahme war hier vielleicht die Frage nach dem Entartung des R-Baums (und das steht nicht mal in seinem Skript). Aber bei der Menge an Prüfungsprotokollen von ihm, kann man sich meiner Meinung nach schon darauf verlassen, dass er keine abenteuerlichen Themen aus seinem Skript fragt. Dennoch habe ich in seinem Prüfungsteil schätzungsweise schlechter abgeschnitten. Das liegt daran, dass man ihm immer relativ komplizierte Sachverhalte am Stück möglichst eloquent erklären muss. Falls man das nicht so gut aus dem Stegreif kann, sollte man sich vor der Prüfung überlegen, wie man z. B. den R-Baum, Anfragen im R-Baum, die Hauptkomponentenanalyse oder Paging am besten erklärt. Ich habe für seinen Prüfungsteil zu viel in Breite gelernt und mich auf seine Standardfragen nicht gut genug vorbereitet.
Prof. Spaniol stellt das Gegenstück dar. Er redet selber sehr viel während seiner Prüfung, so dass man kaum zu Wort kommt. Hier sollte man versuchen, möglichst viel gegenanzureden. Es reicht Stichworte von ihm aufzugreifen oder passende Stichworte einzuwerfen. Mehrere Sätze am Stück redet man bei ihm in der Prüfung nicht. Mit diesem unvorhersehbare Prüfungsstil kam ich deutlich besser zurecht. Für so eine Prüfung sollte man sehr in die Breite lernen. Er springt so schnell von einem Thema zum nächsten, dass Details bei der Prüfung unter den Tisch fallen. Und wenn man ein Detail mal nicht weiß (z. B. umlabeln von Paketen bei MPLS) geht das in der Gesamtheit der Unmengen an Fragen eher unter. Die Frage nach den 2^{32} und die Kombinatorikfrage zur Übersetzung zeichnen wahrscheinlich den Prüfungsstil von Prof. Spaniol aus. In einem Protokoll hab ich gelesen, dass er auch mal gefragt hat, wie denn die O-Notation definiert sei. Solche Fragen sind zwar nicht schwer, aber treffen einen absolut unvorbereitet. Anscheinend fallen die aber auch nicht so schwer ins Gewicht, schließlich hat mir sein Prüfungsteil sogar die 1.3 gerettet. Es gibt allerdings das Gerücht, dass Prof. Spaniol sehr launisch sei und auch schlechte Noten vergibt, wenn er nicht gut drauf ist. Ich kann natürlich nicht sagen, ob an dem Gerücht etwas dran ist. Vielleicht hat auch Prof. Seidl die nötige gute Laune mit in die Prüfung hineingetragen. Nichtsdestotrotz hatte ich während der Prüfung bei Prof. Spaniol eher das Gefühl, dass einem nichts passieren kann, solange man in die Breite gelernt hat, eine gute Übersicht über den Stoff hat, sich mit gesundem Menschenverstand an seinen Diskussionen beteiligt und immer schön nickt, wenn er am reden ist. Mir hat die Prüfung Spaß gemacht.
Ich kann daher beide Professoren als Prüfer weiterempfehlen, besonders in der Kombination.

Zum Lernen verwendete Literatur:

- Folien (manche Leute nennen das Skript) zur Vorlesung Modelle der Datenexploration
  `http://www-i9.informatik.rwth-aachen.de/lehre/hauptstudium/datenexploration/ws2004/`
- Andrew S. Tanenbaum: Moderne Betriebssysteme (2. Auflage)
  (Ich habe öfter schon gehört, dass die deutsche Übersetzung von dem Buch so schlecht sein soll. Ich konnte das nicht nachvollziehen und kann somit auch die deutsche Ausgabe weiterempfehlen.)
- Folien zur Vorlesung Datenkommunikation
  `http://www-i4.informatik.rwth-aachen.de/content/teaching/lectures/sub/datkom/WS03-04/`
  `http://www-i4.informatik.rwth-aachen.de/content/teaching/lectures/sub/datkom/WS05-06/`
- Andrew S. Tanenbaum: Computernetzwerke bzw. Computer Networks (3. & 4. Auflage)
  (habe hauptsächlich nach den Folien gelernt, im Tanenbaum habe ich nur ein paar mal drin rumgeblättert)
- Folien zur Vorlesung Verteilte Systeme (Distributed Systems)
  `http://www-i4.informatik.rwth-aachen.de/content/teaching/lectures/sub/vs/vsSS04/`
- Alte Skripte vom i4
  `http://www-i4.informatik.rwth-aachen.de/content/teaching/scripts/downloads/lokale_netze.pdf`
  `http://www-i4.informatik.rwth-aachen.de/content/teaching/scripts/downloads/verteilte_systeme.pdf`
  (habe ich nur mal überflogen)
- Die Prüfungsprotokolle…
  - von www.s-inf.de
  - aus der Fachschaft. Für Protokolle von Prüfungen von Prof. Spaniol in den Ordnern „Informatik Diplom Praxis“ schauen. Aktuelle Protokolle von Prof. Spaniol sind leider eine Rarität.

Übrigens, das sind nur Gedächtnisprotokolle, d. h. die Dialoge sind nur grob rekonstruiert. Besonders der Dialog mit Prof. Spaniol, bei dem wir sehr häufig auch gleichzeitig geredet haben, lässt sich unmöglich vernünftig protokollieren. Ich finde lediglich, dass sich so ein konstruierter Dialog viel cooler liest und dass man so einen besseren Eindruck von dem Fragestil der Professoren bekommt.
Man sollte übrigens vermeiden, während der Prüfung das Wort 'quasi' zu verwenden. Das ist so ein Tick von mir, über den sich Prof. Spaniol konsequent hergemacht hat: „Nein, das ist nicht nur quasi so… das ist so.“ Im Prüfungsprtokoll findet sich das Wort quasi nicht, was verdeutlicht, wie gekünstelt dieses Protokoll doch ist. :)

Wünsche Euch viel Erfolg bei Euren Prüfungen!