

Prüfungsprotokoll Praktische Informatik

Semester: SS 05
Prüfer: Prof. Seidl
Prüfungsgebiete: 1.) Einführung in Datenbanken (nach Kemper/Eickler 5. Auflage, Kap. 1-7, 9, 13-16)
2.) Indexstrukturen für Datenbanken (nach Folien vom SS 04)
3.) Betriebssysteme (nach Tanenbaum, 2. Auflage, Kap. 1-6, 8, 10,11)
Allgemeines: Insgesamt herrschte während der gesamten Prüfung eine lockere Atmosphäre. Zu Beginn konnte ich aussuchen, mit welchem Gebiet ich anfangen wollte. Die Prüfung hat ziemlich genau 45 Minuten gedauert. Mir kam es so vor, als wäre Betriebssysteme etwas kürzer geprüft worden als der Rest, sicher kann ich das aber nicht sagen.

1.) Einführung in Datenbanken

- F: Wie geht man vor, wenn man eine Datenbank entwirft?
A: Anforderungsanalyse, konzeptueller Entwurf, Implementationsentwurf, physischer Entwurf
- F: Welche Konzepte gibt es für den konzeptuellen Entwurf?
A: ER – Modell und UML – Modell; auf Unterschiede hingewiesen (Methoden und Generalisierung in UML, UML besser, falls oo. Datenbank erwünscht)
- F: Machen Sie mal ein ER – oder UML – Diagramm für Vorlesungen und Studenten!
A: ER – Diagramm aufgemalt mit Attributen, Beziehung hören und Funktionalität N:M
- F: Wie sehen die zugehörigen Relationen aus?
A: drei Tabellen aufgemalt (hören: {[MatrNr, VorlNr]}, Vorlesungen {[VorlNr, Titel]} und Studenten: {[MatrNr, Name, Semester]}); auf Schlüssel hingewiesen
- F: Was ist denn eigentlich ein Schlüssel?
A: minimale Menge von Attributen, aus der man mittels funktionaler Abhängigkeiten alle Attribute des Schemas erhalten kann; falls Menge nicht minimal: Superschlüssel
- F: Die Uni schlägt für die obige Datenbank das Schema [MatrNr, Name, VorlNr, Titel] vor. Ist das eine gute Idee?
A: Nein. Hier sind nicht zusammengehörige Informationen in einer Relation vereint, wodurch es zu Anomalien kommen kann. Update -, Einfüge-, und Löschanomalien erklärt; Normalformen zur Lösung des Problems erwähnt
- F: Welche Anfragesprachen an relationale Datenbanken kennen sie?
A: SQL, RA, TK, DK
- F: Wie sieht denn eine Anfrage in SQL aus, um alle Matrikelnummern von Studenten zu erhalten, die die Vorlesung Datenbanken hören?
A: SELECT h.MatrNr FROM hören h, Vorlesungen v WHERE h.MatrNr = v.MatrNr AND v.Titel = 'DB'; hier habe ich erklärt, wo hier der Join zwischen hören und Vorlesungen realisiert wird

F: Wie sieht die Anfrage in RA/DK/TK aus?
A: hingeschrieben; erwähnt, daß im TK Variablen an Tupel einer Relation und im DK die Variablen an einzelne Attribute gebunden sind

F: Welche weiteren Operationen in RA kennen sie, und welche Entsprechungen in SQL gibt es?
A: Vereinigung (UNION / UNION ALL), wobei UNION implizit Duplikate eliminiert und die Schemagleichheit vorausgesetzt ist, Umbenennung von Relationen (in SQL im FROM-Teil) und Attributen (in SQL im SELECT-Teil durch Schlüsselwort AS), Mengendifferenz und kartesisches Produkt

2.) Indexstrukturen für Datenbanken

F: Braucht man Indexstrukturen?
A: Theoretisch nicht, da immer ein sequentieller Scan auf den Daten möglich ist. Wenn die Effizienz jedoch sehr wichtig ist, kann man darauf eigentlich nicht verzichten.

F: Komplexität eines sequentiellen Scans?
A: linear, also $O(n)$

F: Welche Indexstrukturen für eindimensionale Daten kennen sie?
A: Hashing, Bitmap, B(*)-Bäume, näher auf Hashing eingegangen (Eigenschaften einer Hashfunktion, offenes / geschlossenes Hashing, Sondierungsverfahren und jeweils die Verbesserung eines solchen Verfahrens gegenüber einem anderen erwähnt; Problem bei statischem Hashing erwähnt und auf Möglichkeit, dynamisches Hashing zu verwenden, hingewiesen, wozu er jedoch nichts weiteres wissen wollte)

F: Wozu gibt es denn die Bitmap?
A: für Attribute mit kleinem Wertebereich (z.B. Bitvektor „männlich“); auf Effizienz und Auswertung von Leseanfragen und Problematik bei Schreiboperationen hingewiesen

F: Wie ist denn ein B-Baum definiert?
A: Definition genannt, Knoten skizziert, Komplexität für Suche ($O(\log_{k+1}(n))$) und Zusammenhang zwischen Verzweigungsgrad und Suchkomplexität erwähnt

F: Kann denn ein B-Baum entarten?
A: Nein, da er, wenn überhaupt, dann nach oben wächst und somit immer balanciert bleibt.

F: Wie groß ist denn normalerweise eine Seite?
A: zwischen 512 Byte und 8 KB, auf jeden Fall im KB – Bereich

F: Wie berechnet man denn die Minimalbelegung k der Knoten im B-Baum?
A: hier hatte ich absolut keine Ahnung, Prof. Seidl hat dann irgendwas davon gesagt, daß man die Größe der Daten ja kenne und auch die Größe einer Seite auf der Platte. Daraus könne man zunächst bestimmen, wie viele Daten auf eine Seite passen würden. Weiter weiß ich jetzt leider auch nicht mehr.

- F: Welche Indexstrukturen gibt es denn für Daten, deren Koordinaten nicht bekannt sind?
- A: metrische Daten; beruhen auf Abschätzungen (Symmetrie, Definitheit, Nichtnegativität und Dreiecksungleichung); als Beispiel den M-Baum genannt und erklärt, wie ein Knoten davon aussieht (Routingobjekte, Radian der Routingobjekte, Distanzen zum Vorgänger und Verweise auf Teilbäume); gezeigt, welches Kriterium es gibt, um abzuschätzen, ob in einen Teilbaum abgestiegen werden muß (siehe Folie 5-16)

3.) Betriebssysteme

- F: Was ist ein Prozeß?
- A: Abstraktion eines laufenden Programms; hat Adreßraum und Prozeßtabelleneintrag; Adreßraum enthält Programm, Programmdatei, Variablen, Befehlszähler usw.; Prozeßtabelle speichert Befehlszähler, Speicherbelegung, Kellerzeiger usw., damit bei Wiedereinlagerung des Prozesses die CPU weiß, wo sie weitermachen muß und auch die Zwischenergebnisse wieder kennt
- F: Ich würde einen Prozeß nicht als Abstraktion, sondern eher als Konkretisierung eines Programms sehen. Warum?
- A: Weil das Programm ja gerade ausgeführt wird und somit quasi eine Instanz des Programms darstellt.
- F: Warum benötigt man ein Betriebssystem? Könnten die Prozesse nicht auch so arbeiten?
- A: Betriebssystem als Ressourcenverwalter erwähnt (hier kann ich mich leider nicht mehr erinnern, was ich sonst noch so gesagt habe)
- F: Wie benutzt ein Prozeß E/A – Geräte?
- A: Normalerweise durch einen Systemaufruf oder (z.B. in UNIX) durch Spezialdateien. Betriebssystem führt die E/A aus und schaltet dann in den Benutzermodus zurück.
- F: Welche Bedingungen gibt es denn für Deadlocks?
- A: Wechselseitiger Ausschluß, Hold – And – Wait – Bedingung, Ununterbrechbarkeit und zyklisches Warten; alle vier Bedingungen kurz erklärt
- F: Wie werden Deadlocks denn behandelt?
- A: In Windows und Unix eigentlich gar nicht (Vogel-Strauß-Methode), ansonsten gibt es noch die Möglichkeit, die Ressourcen vorsichtig zuzuteilen, Deadlocks zu erkennen und zu beheben oder eine der vier Bedingungen unmöglich zu machen.
- F: Wie funktioniert denn die Seitenadressierung?
- A: Falls kein virtueller Speicher unterstützt wird, wird die Seitenadresse auf den Speicherbus gelegt und der Inhalt aus dem Speicher geladen. Sonst geht die Seitenadresse zuerst zur MMU, die die Seitentabelle verwaltet. Falls Seite nicht im Speicher ist, muß sie von der Platte geladen werden. Eventuell muß dann eine andere Seite ersetzt werden. Dann habe ich noch die Seitenersetzungsalgorithmen FIFO, SC, Clock, NRU, NFU erwähnt und erklärt.