AFB! - SEHSCHLANGE

Verfasser: Jens Köhler
Quelle: Fachschaft Mathematik/Informatik
Erstellungsdatum: 24.08.1999
Fach: Informatik 3./4. Semester
Vor-/Hauptdiplom: Vordiplom
Studienfach: Informatik
Prüfer: Prof. Joubert
Beisitzer: Vogel
Prüfling: unbekannt
Datum: 11.07.1991
Dauer: 45 Minuten
Note: unbekannt
Prüfungsklima: Angenehm. Prof. Joubert legt nicht nur Wert auf Fachwissen, sondern auch auf strukturierte Darstellung und übergeordnete Zusammenhänge
Prüfungsablauf:

- Turing-Maschine
- Von-Neumann-Rechner /-Architektur
- Betriebssysteme allgemein (Aufgaben, Funktionen).
- warum Parallelrechner
- Verbesserungsmöglichkeiten der Von-Neumann-Rechner
- Betriebssysteme Parallelrechner (Aufgaben, Organisationsformen)
- Prozessor- Synchronisation (Bsp: Matrix-Vektor, DGL-Iteration)
- obj ^.meth(x, y); mit procedure meth$(var\ a : \ldots ; b : \ldots)$; virtual; Ablauf auf Assemblerebene erläutern
- VMT (wann, was)
- Unterschied MACRO-/UP-Aufruf (Assembler) (wann welches)
- Kann MACRO-Wdh-Anweisung (REPT) mit „Variable" aufgerufen werden? ($\rightarrow$ nein, Wert muß zur Compilierzeit bekannt sein.
- Warum können Standard-Sortieralgorithmen (aus Bibliotheken) selten direkt in eigenen Programmen verwendet werden? (jeweils ganz spezielle Anforderungen/Umgebung, z.B. spezielle Datenstrukturen in eigenem Programm...)
- Lösung? (Problem der Wiederverwendbarkeit): Objektorientierte Programmierung. (Erläuterung, z.B. Implementierung eines solchen 'abstrakten Sortierverfahrens'?)
- Wie kann der Rechner jeweils die 'richtige' Methode anwenden? (virtuelle Methoden, Erläuterung des Verfahrens der VMT...)
- Woher weiß ein Destruktor-Aufruf, wieviel Speicher freizugeben ist? (Speicherbedarf eines Objektes auch in VMT enthalten)
- Parameterübergabe bei virtuellen Methoden (anhand letztendlicher Assembler-Realisierung) ? (Übergabe auf Stack vom aufzurufenden Programmteil,

muß bei allen virtuellen Methoden daher gleich sein, keine Konflikte, da Methodenköpfe im virtuellen Fall identisch sein müssen!)

- Sortierverfahren: Einteilung? (bottom-up, top-down, hardsplit-easyjoin, easysplit-hardjoin + Erläuterung)
- Erläuterung von heap-Sort (top-down, hardsplit, Anschauliche Erklärung des Verfahrens anhand eines Beispiel-Baumes...)
- Welche Zeitkomplexität hat heap-sort? ($n \cdot log_2 n$)