AFB! - SEHSCHLANGE

Verfasser: Jens Köhler
Quelle: Fachschaft Mathematik/Informatik
Erstellungsdatum: 24.08.1999
Fach: Informatik 3/4 (Algorithmen und Datenstrukturen)
Vor-/Hauptdiplom: Vordiplom
Studienfach: Informatik
Prüfer: Prof. Joubert
Beisitzer: Ute Vogel
Prüfling: Frank Büchler
Datum: 05.07.1991
Dauer: unbekannt
Note: 2,7
Prüfungsklima:

- sehr gut. Nachdem ich erstmal die anfängliche Nervosität abgelegt hatte, schien die gesamte Prüfung mehr ein gemütliches Gespräch zu sein.
- Prof. Joubert ist gerne bereit, Fragen umzuformulieren und gibt auch ausreichend Hilfen
- man muß damit rechnen, daß verschiedene Themengebiete in der Prüfung neu in Bezug gesetzt werden (der grobe Überblick sollte also vorhanden sein), aber es gibt keinen Grund zur Panik, da in der Regel klar ist, was er im Visier hat.
- Ute prüfte nur kurz einige Bröckchen Datenstrukturen und war fast überrascht, daß sie überhaupt noch dran kam (die Zeit war sehr schnell vorangeschritten und Joubert überlegte schon, abzubrechen)

Prüfungsablauf:: Wir haben uns ja mit der Von-Neumann-Maschine beschäftigt, wie sieht das denn nun auf der unteren Ebene mit der Ausführung eines Befehls so aus?
(1. Reaktion: Nanun, kein Turing-Test.....
2. Reaktion: Was will er jetzt? Das Zwei-Takt-Zyklus-Schema? Es bereitete mir einige Probleme, dies festzustellen, bis mir dann die berühmte Lampe leuchtete → Mikroprogrammierung und festverdrahtete Logik sollte kommen – kam dann auch! –)
Schön, nun hat unsere Von-Neumann-Maschine ja so ein paar Probleme mit der Geschwindigkeit. Was kann man da denn so gegen tun?
(Hier wollte er eine Menge hören: Busse verändern (Arten?), Cache (was für eine Speicherart ist das → assoziativ), Problem der Von-Neumann-Sprachen, E/A-Prozessoren. Auf Parallelisierung wollte er nicht hinaus.)
So langsam steuerten wir auf sein Lieblingsthema hin: Parallelrechner → Klassifizierung?
(Hier reichte ihm die Klassifizierung nach Flynn)

Sie und ihr Freund haben beide einen Rechner, die sie nun zusammengekoppelt haben → MIMD-Rechner, wie können Sie nun folgenden Prozeß parallelisieren? Hier kam er nun mit einem Riemann-Integral, das über Rechteck-Regel iteriert werden sollte.
(Zunächst hatte ich Probleme, hier überhaupt einen Ansatz zu finden, kam dann aber nach einem kleinen Anschub darauf, einfach die einzelnen Rechtecke auf die Prozessoren zu verteilen. Zum Beispiel einem Prozessor die geraden, diem anderen die ungeraden zuzuteilen.
Danach wollte er wissen, wie man diese Prozesse mit CSP synchronisieren könnte.
(Mit Parallel-Befehl, beide Prozesse initialisieren usw...)
Neues Themengebiet: Funktionen eines Betriebssystems
(Ich fragte ihn, ob er die drei oder sieben hören möchte. Er grinste und sagte: „na gut, dann nur die drei, is' kürzer.". – Die sind ja wohl allen geläufig! –)
Nun kam der gesamte Komplex der Synchronisation.
(wechselseitiger Ausschluß, Prozeß-Kooperation (Reihenfolge?), Semaphoren, je ein Beispiel, Unix → Pipeline)
Danach noch ein kurzer Abstecher in die Pipeline-Rechner (Arbeitsweise, in etwa reicht). Haben die was mit der Unix-Pipeline zu tun? (Breites Grinsen ⇒ Kein Stück!)
Dann kam einer kleiner Abschuß: Dateiverwaltung, was sind inverse Dateien?
(Wildes Raten → Volltreffer auf Lücke, war aber nicht so schlimm)
Joubert blickt auf die Uhr und stellt fest, daß die Zeit eigentlich schon vorbei ist. „Ute, willst Du noch was über Datenstrukturen oder so fragen?"
„Äh, öh – na gut, n' bissche kann ich ja noch fragen."
Wie kann eine lineare Ordnung implementiert werden? → ARRAY
Einfügen in ARRAY. (Problematisch, starre Struktur ⇒ besser Liste)
Gut, wie sieht denn so eine Liste aus?
(Kästchen malen, einfügen demonstieren usw. -einfach-)
Wir haben ja nun sämtliche Sortieralgorithmen auf ARRAYs definiert. Such Dir einen Algorithmus aus, erkläre ihn und wende ihn auf die Liste an.
(Meine Wahl fiel auf Quicksort, siehe Script. Anwenden auf die Liste macht auch keine Probleme.)
END