

Prüfungsfragen BS/VS 2005

Betriebssysteme und Verteile Systeme, Prof. Schröder-Preikschat, Beisitzer: Dr. Olaf Spinczyk Juli 2005

Bemerkungen zu Prüfung und Prüfer

- Ergebnis: 1,3
- Angenehme Atmosphäre
- Wenn man auf etwas nicht kommt, führt der Prüfer einen auf das Ergebnis hin

Fragen

Betriebssysteme

- Semaphore, P() und V() Operation, Semantik
- P und V als kritischer Abschnitt. Wie synchronisieren (Interrupts sperren, Unterbrechungstransparente Synchronisation, Elementarbefehle (cas und so), incl)
- unter welcher Randbedingung ist incl verwendbar? (Monoprozessorsystem)
- Probleme mit Semaphorwarteschlange (Prioritätsverletzung, Prioritätsumkehr)
- Was kann man dagegen tun (Scheduler Semaphorewarteschlange verwalten lassen, Prioritätsvererbung)
- Deadlocks, Bedingungen (1-4, direkt, indirekt)
- Was kann man dagegen machen (prevention, avoidance, recognition, Auflösen)
- Was davon ist praxisrelevant ?

Verteilte Systeme

- Deadlock durch RPC - Callback (Analog Beispiel mit den Türmen von Hanoi), warum ? (Ein Faden ist gleichzeitig Dienstanbieter und Dienstnehmer)
- Was kann man dagegen tun (Zustandsmaschine, Zwei Fäden)

- Was ist besser und warum (Zwei-Fäden-Lösung, da remote invocation send und damit nichtgleichberechtigte Kommunikation beibehalten wird)
- RPC Semantiken (Maybe, At-Least-Once, At-Most Once, Exactly Once, erklären)
- Szenario: Ein Geldautomat ist Dienstnehmer, der Bankserver Dienstanbieter. Abheben als idempotente Operation formulieren (setkontostand (getkontostand()-betrag))

Middleware / Verteilte Systeme Dr.-Ing. Jürgen Kleinöder Juli 2005

Bemerkungen zu Prüfung und Prüfer

- Ergebnis: 1,0
- Die Prüfungsatmosphäre war sehr angenehm, allerdings könnten die Fragen etwas konkreter sein.

Fragen

Thematisch ging die Prüfung hauptsächlich über Middleware-Stoff, wobei der Stoff von Verteilte Systeme sich sehr stark überlappt, v.a. beim Thema RPC/Fernaufruf.

- Warum ist Objektorientierung sinnvoll? — Datenorientierte Programmierung skaliert nicht. ⇒ Objekte, Klassen, Typen, Abstraktion, Kapselung, ...
- Wie funktionieren Objekte im verteilten Fall? — Irgendjemand muß die gleiche Schnittstelle auf einem Server anbieten.
- Wie funktioniert ein lokaler Funktionsaufruf und wie im verteilten Fall? — Bei einem lokalen Funktionsaufruf werden die Parameter über den Stack übergeben und dann die neue Methode aufgerufen. Im verteilten Fall müssen die Parameter auch übergeben werden (by-value — Serialisierung, by-reference — Callback-Möglichkeit). Bei Referenzen besteht das Problem, daß der Client auch Aufrufe entgegennehmen muß und damit als Server

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage.
--

fungiert. Dazu braucht er entweder einen Zustandsautomaten oder er muß mehrere Threads erzeugen.

- Wo kommen die Objekte eigentlich her? — Irgendjemand muß sie erzeugen, z.B. explizit der Server (oder Client bei .NET) oder sie kommen aus einer Datenbank usw.
- Wie funktioniert es bei Corba/EJB (ganz allgemein)? Wie sind die Unterschiede? — Bei EJB ist alles (z.B. Transaktion, Sicherheit, ...) transparent, die Konfiguration wird mit einer XML-Datei festgelegt. Bei Corba passiert dies alles explizit.
- Was kann schiefgehen bei der Kommunikation zwischen Client und Server? — Anfrage geht verloren, Antwort geht verloren, ... Gemeint waren im Endeffekt die Aufrufsemantiken:
 - maybe — Ist das sinnvoll? Ja, denn manchmal ist es egal, ob man eine Antwort bekommt.
 - at-least-once — Idempotente Operationen, aber das geht nicht immer. Lösung: Berechnung am Client statt auf dem Server — keine relativen Aufrufe (z.B. statt Hebe 20 Euro ab. → Setze Kontostand auf 80 Euro.)
 - at-most-once — Unterschied at-most-once/maybe: gar nicht so groß...
 - exactly-once — Am Ende des Tages braucht man das fast immer und die Anwendungsschicht muß sich darum kümmern, daß es wirklich genau einmal ausgeführt wird (Stichwort Transaktionen ist nicht fehl am Platz). Praktisch kann man exactly-once aber nie garantieren (z.B. Serverausfall).

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage.
--