

Hauptdiplomklausur Informatik

März 1996

Teil: Rechnernetze–Praktikum

Name: Vorname:

Matrikel-Nr.: Semester: Fach:

Hinweise:

- a) Bitte füllen Sie sofort den Kopf des Deckblatts aus.
- b) Überprüfen Sie Ihr Klausurexemplar auf Vollständigkeit (12 Seiten).
- c) Tragen Sie Ihre Lösungen soweit möglich direkt in die Klausur ein.
- d) Es sind keine Hilfsmittel zugelassen.
- e) Zeit: 67 Minuten

Aufgabe	max. Punktezahl	Punkte
1	20	
2	29	
3	18	
Summe	67	

Aufgabe 1 [4+4+4+8 = 20 Punkte] TCP/IP und UDP/IP

- a) [4 Punkte] TCP ist das verbindungsorientierte, UDP das verbindungslose Transportprotokoll des Internet. Welche Eigenschaften besitzt TCP verglichen mit UDP, die den Begriff der Verbindungsorientiertheit rechtfertigen? Mit welchen Mitteln lassen sich diese Eigenschaften realisieren?
- b) [4 Punkte] Nennen Sie je zwei typische Anwendungen, die auf TCP bzw. UDP beruhen!

- c) [4 Punkte] Begründen Sie, warum das verbindungslose Transportprotokoll UDP besser für eine *Video*-Übertragung geeignet ist als TCP!

- d) [8 Punkte] Schreiben Sie ein Programmstück, das es einem auf TCP Sockets basierenden Server erlaubt, für eine neue Verbindung einen neuen Prozeß zu starten. Der alte Prozeß wartet auf weitere Verbindungsaufbauwünsche. Verwenden Sie die Routinen `accept()` und `fork()`. Der Aufruf von `fork()` liefert eine Prozeß-ID als Ergebnis (Typ: `pid_t`). Ein Prozeß kann an der Prozeß-ID feststellen, ob er der Vater oder der Sohn ist; im ersten Fall gibt sie die ID des Sohnprozesses an, im zweiten hat sie den Wert 0. Verwenden Sie folgende Vorgaben:

```
int socket1,socket2;
....
socket1 = socket(...);
bind(socket1,...);
listen(socket1,5);
/* Jetzt kommt Ihr Code!! */
```

Aufgabe 2 [6+8+15 = 29 Punkte] *Remote Procedure Call*

- a) [6 Punkte] Beschreiben Sie anhand einer Graphik das Modell des entfernten Prozeduraufrufs. Gehen Sie insbesondere auf die Unterschiede zum lokalen Prozeduraufruf ein.

- b) [8 Punkte] Wie wird beim Sun-RPC die Kommunikation realisiert, d.h., was passiert, wenn eine entfernte Prozedur aufgerufen wurde? Gehen Sie auf den Weg der Daten vom Client zum Server und den Rückweg ein.

- c) [15 Punkte] Gegeben sei folgendes Problem: ein Programm A benötigt die Summe zweier Produkte ganzer Zahlen. Leider ist es aus unerfindlichen Gründen nicht in der Lage, diese Berechnung selbst durchzuführen. Glücklicherweise bietet ein anderes Programm B diesen Dienst als entfernte Prozedur an. Es erwartet als Eingabe den Typ **Komponenten**, der aus vier ganzen Zahlen a , b , c und d besteht, und gibt das Ergebnis $e = a * b + c * d$ ebenfalls als Ganzzahl zurück.

Erstaunlicherweise ist jedoch auch dieses zweite Programm nicht in der Lage, sämtliche Berechnungen selbst durchzuführen. Vielmehr kann es selbst nur Zahlen addieren. Aber wir haben erneut Glück: ein drittes Programm C ist in der Lage, zwei Zahlen miteinander zu multiplizieren. Es bietet diesen Dienst ebenfalls als entfernte Prozedur an und erwartet zwei Ganzzahlen als Eingabe. Als Ergebnis liefert es das Produkt dieser Zahlen als Ganzzahl.

1. [3] Skizzieren Sie kurz, wie sich dieses Problem mit RPC lösen läßt.

2. [6] Schreiben Sie die *beiden* RPC-Beschreibungsdateien auf für den Sun-RPC (also in XDR), die die Schnittstellen für die entfernten Prozeduren enthalten. Die erste Prozedur soll `addiereProdukte` heißen, die zweite `multipliziere`.

3. [6] Schreiben Sie die Implementierung der Funktion `addiereProdukte` im Programm B (in der Sprache C). Mittels `rpcgen` haben Sie bereits den Rahmen generiert, der Ihnen vorgegeben wird.

```
int *
addiereprodukte_1(argp, rqstp)
    Komponenten *argp;
    struct svc_req *rqstp;
{
    static int  result;

    /*
     * insert server code here
     */

    return (&result);
}
```

Aufgabe 3 [4+5+4+5 = 18 Punkte] Estelle

- a) [4 Punkte] Welche Vorteile kann die Verwendung von formalen Sprachen bei der Systementwicklung haben?
- b) [5 Punkte] Welche Funktionen muß ein Werkzeug unterstützen, das die Entwicklung einer ablauffähigen Implementierung aus einer formalen Spezifikation ermöglicht?

- c) [4 Punkte] Beschreiben Sie die generelle Struktur einer Estelle-Spezifikation (möglichst anhand einer Graphik)!

d) [5 Punkte] Setzen Sie den folgenden Transitionsteil eines Estelle-Moduls in eine Zustands-Transitionstabelle um!

```
TRANS
  FROM s1 to s2
    WHEN sap.msg1 begin
      output sap.msg2;
    end;

  FROM s1 to same
    provided p1=TRUE begin
      p1:=FALSE;
    end;

  FROM s2 to s3
    WHEN sap.msg3
    provided p2 = TRUE begin
      output sap.msg4;
    end;

  FROM s3 to s1
    WHEN sap.msg4 begin
    end;
```
