

Hauptdiplomklausur Informatik

September 1992 Teil: Rechnernetz-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 (13 Seiten).
- c) Es sind keine Hilfsmittel zugelassen.
- d) Zeit: 67 Minuten

Aufgabe	max. Punktezahl	Punkte
1	22	
2	25	
3	20	
Summe	67	

Aufgabe 1 [22 Punkte] *Client-Server-Konzept in TCP/IP*

Im folgenden ist ein simpler Arithmetik-Server und das Skelett eines entsprechenden Clients abgedruckt.

a) [4 Punkte] Wozu dienen die folgenden Aufruf im Code des Servers?

1. getservbyname

2. gethostbyname

3. bind

4. listen

b) [3 Punkte] Erweitern Sie den Server um die Operation '*'.

c) [10 Punkte] Erweitern Sie den Client, so daß er den Ausdruck $50 + 4 * 3$ mit Hilfe des Servers berechnet.

d) [5 Punkte] Angenommen der Server und der Client laufen auf Maschinen mit unterschiedlicher Byteanordnung. Funktioniert die Kommunikation trotzdem? (Kurze Begründung)

Server-Code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/stat.h>
6 #include <sys/socket.h>
7 #include <netinet/in.h>
8 #include <netdb.h>

9 #define MAXHOSTNAME 32
10 void arithmetik(int);

11 extern int errno;
12 main(argc, argv)
13 int argc; char**argv;
14 { int s;
15     struct servent *sp;
16     struct hostent *hp;
17     struct sockaddr_in sa, isa;
18     char localhost[MAXHOSTNAME+1];

19     if((sp = getservbyname("klausur", "tcp")) == NULL)
20         { printf("No such service: klausur\n"); exit(1);
21         }
22     sa.sin_port = sp->s_port;
23     gethostname(localhost, MAXHOSTNAME);
24     if((hp = gethostbyname(localhost)) == NULL)
25     { printf("Error when trying to get localhost info\n"); exit(1);
26     }
27     bcopy((char*)hp->h_addr, (char*)&sa.sin_addr, hp->h_length);
28     sa.sin_family = hp->h_addrtype;
29     bzero(sa.sin_zero, 8);
30     if((s = socket(hp->h_addrtype, SOCK_STREAM, 0) ) < 0)
31     { perror(" socket "); exit(1);
32     }
33     if(bind(s, &sa, sizeof(sa), 0) < 0)
34     { perror(" bind 1 "); exit(1);
35     }
36     listen(s, 5);
37     for(;;)
38     { int i=sizeof(isa);
39         int t;
40         if((t = accept(s, &isa, &i)) < 0)
41             { perror(" accept ");
42             }
```

```

43     arithmetik(t);
44     close(t);
45 }
46 }
47 void arithmetik(int t)
48 { int a, b, l, erg;
49   char op;
50   char buf[20];
51   l=read(t, buf, 20);
52   buf[l]=0;
53   a=atoi(buf);
54   read(t, &op, 1);
55   l=read(t, buf, 20);
56   buf[l]=0;
57   b=atoi(buf);
58   buf[l]=0;
59   switch(op)
60     { case '+':
61       erg = a+b;
62       break;
63       .
64       .
65       .
66       .
67       .
68       .
69       .
70     }
71   sprintf(buf, "%d", erg);
72   write(t, buf, strlen(buf));
73 }

```

Client-Code:

```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <sys/socket.h>
4 #include <netinet/in.h>
5 #include <netdb.h>
6 #include <ctype.h>

7 #define MAXHOSTNAME 32
8 void arithmetik(int);

9 main(argc, argv)
10 int argc; char**argv;
11 { int s;
12     struct servent *sp;
13     struct hostent *hp;
14     struct sockaddr_in sa;
15     char *host;

16     /* Ueberpruefen, ob ein Host angegeben wurde */
17     if (argc>1)
18     { host = argv[1];
19       argv++; argc--;
20     } else
21     { printf("Bitte Host angeben\n"); exit(1);
22     }
23     /* Portnummer holen */
24     if((sp = getservbyname("klausur", "tcp")) == NULL)
25     { printf("No such service: klausur\n"); exit(1);
26     }
27     sa.sin_port = sp->s_port;
28     /* Information ueber den eigenen Rechner abholen: */
29     if((hp = gethostbyname(host)) == NULL)
30     { printf("Unknown Host:%s\n", host); exit(1);
31     }
32     /* Setzen der Parameter fuer die Verbindung: */
33     bcopy((char*)hp->h_addr, (char*)&sa.sin_addr, hp->h_length);
34     sa.sin_family = hp->h_addrtype;
35     bzero(sa.sin_zero, 8);
36     if((s = socket(hp->h_addrtype, SOCK_STREAM, 0) ) < 0)
37     { perror("socket"); exit(1);
38     }
39     /* Verbindungsaufbau: */
40     if(connect(s, &sa, sizeof(sa), 0) < 0)
41     { perror("Client"); exit(1);
42     }
};
```

```
43     arithmetik(s);
44     close(s);
45 }

46 void arithmetik(int s)
47 {
48 .
49 .
50 .
51 .
52 .
53 .
54 .
55 .
56 .
57 .
58 .
59 .
60 .
61 .
62 .
63 }
```

Aufgabe 2 [25 Punkte] *Estelle*

Sie haben die Aufgabe, die Sendeinstanz eines *Sliding-Window-Protokolls* in Estelle zu spezifizieren und mit Hilfe des NIST-Estelle-Compilers zu implementieren. Die Fenstergröße betrage 3, der Übertragungskanal sei absolut fehlerfrei. Vorgegeben seien folgende Vereinbarungen:

```
type    UserDataTypes = ...;
channel SlidingWindow(user, provider);
  by user      : DTreq(msg: UserDataTypes);
channel TransferChannel(sender, receiver);
  by sender    : DT(msg: UserDataTypes);
  by receiver  : ACK;
module Sender systemprocess;
  ip    SAP    : SlidingWindow(provider);
       PEER   : TransferChannel(sender);
end;
```

- a) [7 Punkte] Spezifizieren Sie einen `body SenderKonv` für das Modul `Sender`. Dabei sollen Sie nur einen *endlichen* Automaten *ohne* Verwendung von Variablen angeben. Zeichnen Sie nur das Zustandsübergangdiagramm Ihres Automaten; Sie brauchen keinen Estelle-Text zu schreiben.

- b) [7 Punkte] Erstellen Sie nun einen `body SenderErweitert`. Es soll sich dabei um einen *erweiterten* endlichen Automaten handeln. Auch hier genügt ein (erweitertes) Zustandsübergangsdiagramm.
- c) [4 Punkte] Angenommen, Ihre beiden Automaten liegen in Estelle geschrieben vor und sind Bestandteil einer Spezifikation. Sie übersetzen diese Spezifikation mit dem NIST-Compiler. Welche Dateien werden generiert und was ist in ihnen abgelegt? (Gefragt sind nur die Dateien, die einen direkten Bezug zu den beiden Modulbodies `SenderKonv` und `SenderErweitert` haben).

d) [3 Punkte] Vergleichen Sie die beiden Varianten bzgl. Laufzeit, Skalierbarkeit (z.B. variable Fenstergrößen) und Speicherplatzbedarf zur Laufzeit.

e) [4 Punkte] Der NIST-Compiler verfolgt eine bestimmte Strategie bei der Code-Generierung. Erläutern Sie diese, indem Sie das Grundprinzip kurz beschreiben. Gibt es noch andere Strategien? Erläutern Sie Ihre Alternative(n).

Aufgabe 3 [20 Punkte] *ISODE, ROS*

ISODE ist eine Entwicklungsumgebung für OSI-Protokolle und -Anwendungen.

a) [2 Punkte] Welche OSI-Schichten stellt ISODE zur Verfügung?

b) [2 Punkte] Wie sind diese Schichten zugänglich, d. h. wie sehen die Programmierschnittstellen zu ihrer Anwendung aus?

- c) [7 Punkte] ISODE unterstützt die Implementierung von *remote operations* durch eine Reihe von Werkzeugen. Wie wird dabei das Problem der unterschiedlichen Datenrepräsentationen verschiedener Rechner gelöst? Beschreiben Sie kurz das Prinzip und detailliert, wie die Implementierung durch den Werkzeugeinsatz unterstützt wird.

d) [3 Punkte] Ist es mit einer solchen Implementierung möglich, die *context management functional unit* der Darstellungsschicht zu verwenden? Begründung.

- e) [6 Punkte] Beschreiben Sie die drei Fehlersemantiken von ROS. Welcher Verwaltungsaufwand muß jeweils auf der Seite des Senders und des Empfängers betrieben werden, um diese Fehlersemantiken zu implementieren?