1. Analysieren Sie folgendes C-Programm. Welche Fehler können Sie identifizieren (Zeilennummer + Fehlerbeschreibung)? Sie müssen den Code nicht ändern oder erweitern, sondern nur die Fehler beschreiben und die resultierenden möglichen Probleme aufzeigen.
2. Klassifizieren Sie den Servertyp des untenstehenden C-Programms anhand der folgenden Kategorien und begründen Sie Ihre Entscheidung:
   - stateful/stateless
   - connection oriented/connectionless
   - iterative/concurrent

```
01  #include "myheader.h" /* alle notwendigen Includes */
02  #define BUF 1024
03  #define PORT 6543
04
05  int main (void) {
06    int cs, ns;
07    socklen_t addrlen;
08    char buffer[BUF];
09    struct sockaddr_in addr, cl;
10    pid_t pid;
11
12    cs = socket (AF_INET, SOCK_STREAM, 0);
13
14    addr.sin_family = AF_INET;
15    addr.sin_addr.s_addr = "127.0.0.1";
16    addr.sin_port = PORT;
17
18    bind (cs,(struct sockaddr *)&addr,sizeof (addr));
19    listen (cs, 5);
20
21    addrlen = sizeof (struct sockaddr_in);
22
23    while (1) {
24      printf("Waiting for connections...\n");
25      ns = accept (cs, (struct sockaddr *) &cl, &addrlen);
26      if ((pid=fork())==0) {
27        do {
28          recv (ns, buffer, BUF, 0);
29          printf ("Message received: %s\n", buffer);
30        } while (strncmp (buffer, "quit", 4)  != 0);
31        close(cs);
32      }
33    }
34    close (cs);
35    return EXIT_SUCCESS;
```

= inet_addr("127.0.0.1"); es wird die IP in numerischer Darstellung benötigt, nd als Zeichenkette

= htons(PORT); Portnummer in Netzwerkbyte order

BUF-1 /0 null terminator
buffer[BUF-1 ='/0'; weil man sonst irgendwann auf sachen zugreift, auf die man nicht zugreifen sollte

close(ns); error handling, endlosschleife

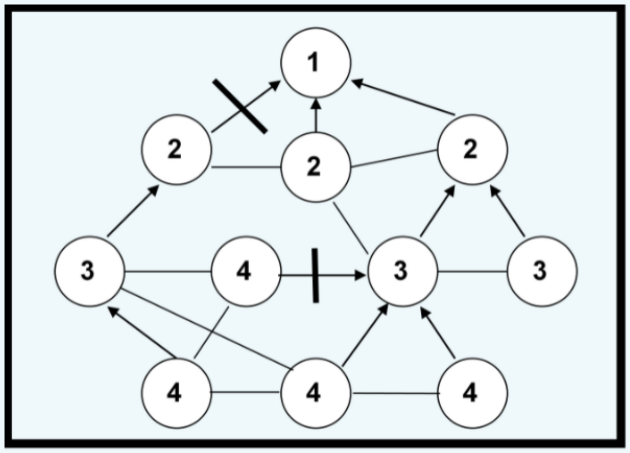
Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

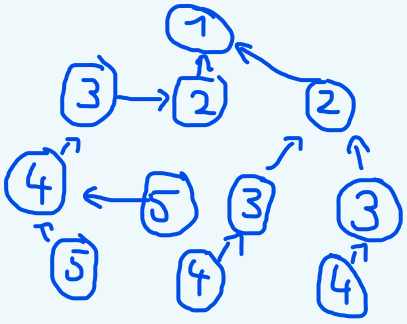
NTP

Reorganisieren Sie das folgende NTP Netzwerk, sodass der Ausfall der Verbindungen (markiert mit einem dicken Querstrich) bestmöglich kompensiert wird.

Legende:

- Kreis mit Ziffer: Knoten mit entsprechendem Stratum
- Pfeil: verbundener Peer
- Linie: mögliche Netzwerkverbindung





Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

LDAP

1. Geben Sie einen syntaktisch korrekten LDAP Suchfilter an, der alle Einträge des LDAP Verzeichnisses mit der Suchbasis „dc=technikum-wien,dc=at" liefert, deren Nachname (Attribut sn) mit M oder N beginnen und die in der Organisationseinheit (Attribut ou) BMR oder BWI zu finden sind.
2. Wie unterscheidet sich die Ausgabe wenn die Abfrage nicht anonym sondern mit Authentifizierung durchgeführt wird?

allgemein (&(objectClass=*)(|(sn=M*)(sn=N*))(|(ou=BMR)(ou=BWI))(dc=technikum-wien,dc=at))

mit ldapsearch ldapsearch -x -b "dc=technikum-wien,dc=at" "(&(objectClass=*)(|(sn=M*)(sn=N*))(|(ou=BMR)(ou=BWI)))"

ldapsearch -x -D "cn=deinBenutzername,dc=deineDomain,dc=com" -W -b "dc=technikum-wien,dc=at" "(&(objectClass=*)(|(sn=M*)(sn=N*))(|(ou=BMR)(ou=BWI)))"

getopt

1. Ein Programm soll als Optionen -q und -c CONFIGDATEI sowie die Angabe eines Verzeichnisses haben. Wie muss die getopt() Funktion aufgerufen werden, um die Kommandozeile zu analysieren?
2. Wie ermitteln Sie den Verzeichnisnamen in diesem Beispiel?

./deinProgramm -q -c konfig.txt /pfad/zum/verzeichnis

```
while ((opt = getopt(argc, argv, "qc:")) != -1) {
    schleifenkörper
```

```
if (optind < argc) {
    directory = argv[optind];
}else error handling
```

Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

Zeitsynchronisation

1. Berechnen Sie die neue Zeit T_C eines Clients nach dem Algorithmus von Christian. Die Sendezeit des Clients T_0=23115. Die Empfangszeit des Clients T_1=23127. Der Reply des Servers enthält die Serverzeit T_S=23120.
2. Skizzieren Sie diesen Request mittels Zeitachsen für Client und Server. Zeichnen Sie die obenstehenden Zahlen und Ihr Ergebnis in die Skizze ein.
3. Wie groß wäre der Fehler bei einer idealen symmetrischen Übertragungsdauer und wie groß bei einer asymmetrischen Übertragungsdauer?

Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

fork

1. Beschreiben Sie die exakte Ausgabe auf stdout. Ist die Ausgabenreihenfolge variabel oder immer ident? Begründen Sie ihre Antwort.
2. Welche Programmierrichtlinie für parallele Prozesse wird verletzt? Erweitern Sie das Programm entsprechend.

```
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main()
6  {
7       pid_t pid;
8       int a=12;
9       int b=5;
10
11      pid = fork();
12      switch (pid)
13      {
14              case -1:
15                      printf("fork failed"); return -1; break;
16              default:
17          sleep(1);
18                      a=3;
19                      b=5;
20                      printf("1: a: %d b: %d\n", a, b);
21                      break;
22              case 0:
23                      sleep(3);
24                      b=a+1;
25                      printf("2: a: %d b: %d\n", a, b);
26                      break;
27      }
28      a++;
29      b--;
30      printf("3: a: %d b: %d\n", a, b);
31      return 0;
32 }
```

immer gleich

1: a: 3 b: 5
3: a: 4 b: 4
2: a: 12 b: 13
3: a: 13 b: 12

gemeinsam genutze ressourcen werden nicht geschützt
möglich mit mutex

4 pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
24 u 18 u 28 pthread_mutex_lock(&mutex);
26 u 21 u 32 pthread_mutex_unlock(&mutex);

oder

zombie prozesse
schutz: wait(NULL) in zeile 17
parent wartet bis child terminiert ist

LDAP

Ein LDAP Server hat folgende Schemadefinition gespeichert:

```
objectclass A
  MUST a, b, c
  MAY  d, e, f

objectclass B
  MUST a, d, x
  MAY  b, y

objectclass C SUP A
  MUST d
  MAY  x, y, z
```

Legende:

- Großbuchstaben: Object Classes
- Kleinbuchstaben: Attribute
- MUST = notwendig, MAY = optional, SUP = abgeleitet von

Welche der folgenden Entries sind gültig?

Select one or more:

- [x] **Implementierte Object Classes:** A, B, C
  Gesetzte Attribute: a, x, d, c, b, y
- [ ] **Implementierte Object Classes:** A
  Gesetzte Attribute: a, c, b, g
- [x] **Implementierte Object Classes:** B, C
  Gesetzte Attribute: a, b, c, d, x, y
- [ ] **Implementierte Object Classes:** C
  Gesetzte Attribute: a, b, c, d, e, f, x, y
- [ ] **Implementierte Object Classes:** A, B
  Gesetzte Attribute: a, b, d, f, e

Socketprogrammierung

networkShort und networkLong

Wozu dienen die Funktionen ntohs(), ntohl()? Warum gibt es keine Funktion ntohstr() für Strings? Begründen Sie Ihre Antwort.

The ntohs() function converts the unsigned short integer netshort from network byte order to host byte order.

wandeln Werte zwischen Netzwerkbyteauftrag und dem Hostbyteauftrag um; Datenaustausch zwishcne rechnern mit unterschiedlicher Architektur (Big Endian und Little Endian); kein string weils kein problem is, wie die chars gereiht sind;

IPC

Welche Unterschiede gibt es zwischen Named und Unnamed Pipes? Nennen Sie 3 Unterschiede!

named pipes haben einen namen im dateisystem
persistenz, sie bestehen fort, solange sie nicht gelöscht werden
müssen expliziet mit mkfifo erstellt werden

haben keinen namen, entstehen idr mit pipe() bei verwandten prozessen eltern/kind dank fork
existieren nur für die dauer der verbindung
sind idr einfacher erstellt und zu verwenden weil sie automatisch als prozesskommunikation erstellt werden

IPC sind nicht dafür ausgelegt große daten zu übertragen, normalerweise auf einige bytes beschränkt; hauptzweck is benachrichtungen über ereignisse zu senden, nicht ganze daten übertragen

Eignen sich Signale zur Kommunikation zwischen Prozessen und zur Übermittlung größerer Datenmengen? Warum / warum nicht?

sind nicht zuverlässig und können verloren gehen (blöd bei kritischen daten
nicht blockierend, senden blockiert den sende rnormalerweise nicht, kann zu timing problemen führen (Stichwort Synchronisation)
kein standardmechanismus für dateninegrität

besser: pipes, message queues, (sockets oder shared memory)

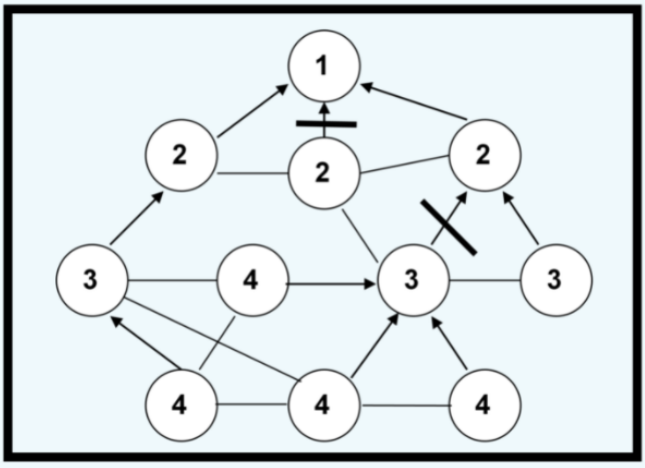
Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

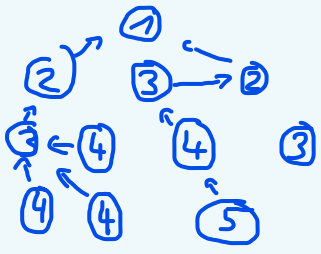
NTP

Reorganisieren Sie das folgende NTP Netzwerk, sodass der Ausfall der Verbindungen (markiert mit einem dicken Querstrich) bestmöglich kompensiert wird.

Legende:

- Kreis mit Ziffer: Knoten mit entsprechendem Stratum
- Pfeil: verbundener Peer
- Linie: mögliche Netzwerkverbindung





Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

Buffergröße: wenn der buffer zu klein ist, wird nur ein teil der daten gesendet und der rest wird abgeschnitten
schrittweise füllen und mehrmals senden; das muss aber auch dementsprechend behandelt werden
send() gibt die anzahl der tatsächlich gesendeten bytes zurück, das soll überprüft werden.

Socketprogrammierung

Was ist beim Aufruf von `send()` zu beachten, wenn größere Datenmengen (z.B. ein Paket mit 100kB) über einen Socket übertragen werden sollen?

senden kann verzögert werden
kann programm blockieren

LDAP

1. Geben Sie einen syntaktisch korrekten LDAP Suchbefehl an, der alle Einträge des LDAP Verzeichnisses mit der Suchbasis „dc=technikum-wien,dc=at" liefert, deren Vorname (Attribut givenname) mit C oder D beginnen und die in der Organisationseinheit (Attribut ou) MSE oder MGS zu finden sind.
2. Wie kann die Suche beschleunigt werden, wenn Sie annehmen, dass alle gesuchten Einträge nur eine Ebene unter dem Teilbaum „ou=People,dc=technikum-wien,dc=at" gespeichert sind?

1) (&(objectclass=*)(|(givenname=C*)(givenname=D*))(|(ou=MSE)(ou=MGS))(dc=technikum-wien.at,dc=at))
2) (&(objectClass=*)(|(sn=M*)(sn=N*))(|(ou=BMR)(ou=BWI))(ou=People)(dc=technikum-wien,dc=at))
sucht jz nur auf ebene ou=People

IPC

ldapsearch -x -LLL -b -s one
sucht nur eine ebene drunter, nicht rekursiv in unterbäumen

Warum können verwandte Prozesse (Eltern und Kindprozess) nicht über globale Variablen Daten austauschen und kommunizieren? Begründen Sie ihre Antwort!

nachn fork() hat child eine kopie des gesamten Addresraumes des parentprocess, einschließlich der globalen variablen
dadurch sind die globalen variablen unabhängig, änderungen haben keine auswirkung
werte werden mitkopiert
es gibt keine synchronisation

fork

1. Beschreiben Sie die exakte Ausgabe auf stdout. Ist die Ausgabenreihenfolge variabel oder immer ident? Begründen Sie ihre Antwort.
2. Erweitern Sie das Programm entsprechend, dass keine Zombie Prozesse entstehen können.

unterschiedlich, je nachdem was zuerst bearbeitet wird; weil kein sleep() ?

```
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main()
6  {
7       pid_t pid;
8       int a=0;
9       int b=0;
10
11      pid = fork();
12      switch (pid)
13      {
14              case -1:
15                      printf("Fork failed"); return -1; break;
16              case 0:
17                      a=3;
18                      b=5;
19                      printf("1: a: %d b: %d\n",a,b);
20                      exit(0);
21              default:
22                      b=a+1;
23                      printf("2: a: %d b: %d\n",a,b);
24                      break;
25      }
26      a++;
27      b--;
28      printf("3: a: %d b: %d\n",a,b);
29      return 0;
30 }
```

von den Folien kopiert

```
pid_t childpid;
while(childpid=waitpid(-1, NULL, WNOHANG)){
   if((childpid == -1) && (errno != EINTR)) {
      breck;
   }
}
```

IPC

Wie können Sie in der Shell eine Named Pipe löschen, wie ein Message Queue? Geben Sie beide Kommandos mit korrekter Syntax an.

rm my_pipe
können nicht direkt gelöscht werden, wird mit beendigung des prozesses gelöscht

ipcs zum auflisten
ipcrm -q IPC:ID

Socketprogrammierung

so wie bereits oben; Unterschied String und char

Wozu dienen die Funktionen ntohs(), ntohl()? Warum gibt es keine Funktion ntohc() für Characters? Begründen Sie Ihre Antwort.

Socketprogrammierung

wie bei send()
buffergröße, teilweise empfangen, blockieren

Was ist beim Aufruf von `recv()` zu beachten, wenn größere Datenmengen (z.B. ein Paket mit 100kB) über einen Socket empfangen werden sollen?

getopt

1. Ein Programm soll als Optionen -x, -v und -f <SOURCEFILE> haben. Wie muss die getopt() Funktion aufgerufen werden, um die Kommandozeile zu analysieren?
2. Wie verhindert man, dass eine Option mehrfach verwendet werden kann?

wie im theoreibeispiel: mit variable, die mitzählt und entsprechendem code

Socketprogrammierung

Listening Socker zum "lauschen" auf eigehende Verbindungen; ist da um Verbindungen auf einem bestimmten Port zu akzeotieren und die Anfrage an separate Sockets weiterzuleiten
Accepted Socket für jede angenommene Verbindung wird ein neuer Socket erstelltt, hier findet die eigentliche Kommunikation statt
bidirectional data transmission

Warum benötigt ein TCP Server 2 Sockets für eine Verbindung? Ist dies auch bei UDP der Fall? Begründen Sie Ihre Antwort.

UDP ist verbindungslos, es gibt keinen dedizierten listening und accepted socket.ein einzelner socket macht beides
unidirectional data transmission

IPC

primitiv: exit status
ein einzelnes byte, das bei beendigung festgestellt wird

pipes:child schreibt, parent liest; einfach, geringer overhead; nachteil: einweg, begrenzte datenmenge

Welche Möglichkeiten haben Sie kennen gelernt, Ergebnisse (z.B. Berechnungen) eines Kind-Prozesses an den Elternprozess zurück zu liefern? Nennen Sie 3 unterschiedliche Varianten und beschreiben Sie deren Vor- und Nachteile.

signals easy,schnell,
keine datenübertragung, unzuverlässig

message queues zuverlässig, asynchron (nachricht kann abgelegt werden ohne darauf zu warten, dass sie sofort abgeholt wird (mailbox))
nachteil: komplexer, mehr overhead, begrenzte größe von queus

LDAP

1. Geben Sie einen syntaktisch korrekten LDAP Suchbefehl an, der alle Einträge des LDAP Verzeichnisses mit der Suchbasis „dc=technikum-wien,dc=at" liefert, deren Nachname (Attribut sn) mit A oder B beginnen und die in der Organisationseinheit (Attribut ou) BIF oder BID zu finden sind.
2. Wie kann die Suche beschleunigt werden, wenn Sie annehmen, dass alle gesuchten Einträge nur eine Ebene unter dem Teilbaum „ou=People,dc=technikum-wien,dc=at" gespeichert sind?

(&(objectClass=*)(|(sn=A*)(sn=B*))(|(ou=BIF)(ou=BID))(dc=technikum-wien,dc=at))

(&(objectClass=*)(|(sn=A*)(sn=B*))(|(ou=BIF)(ou=BID))(ou=People)(dc=technikum-wien,dc=at))

Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

Zeitsynchronisation

1. Berechnen Sie die neue Zeit T_C eines Clients nach dem Algorithmus von Christian. Die Sendezeit des Clients T_0=35335. Die Empfangszeit des Clients T_1=35351. Der Reply des Servers enthält die Serverzeit T_S=35340
2. Skizzieren Sie diesen Request mittels Zeitachsen für Client und Server. Zeichnen Sie die obenstehenden Zahlen und Ihr Ergebnis in die Skizze ein.
3. Wie groß wäre der Fehler bei einer idealen symmetrischen Übertragungsdauer und wie groß bei einer asymmetrischen Übertragungsdauer?

Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

fork

1. Beschreiben Sie die exakte Ausgabe auf stdout. Ist die Ausgabenreihenfolge variabel oder immer ident? Begründen Sie ihre Antwort.
2. Welche Programmierrichtlinie für parallele Prozesse wird verletzt? Erweitern Sie das Programm entsprechend.

```
1  #include <sys/types.h>
2  #include <unistd.h>
3  #include <stdio.h>
4
5  int main()
6  {
7       pid_t pid;
8       int a=0;
9       int b=0;
10
11      pid = fork();
12      switch (pid)
13      {
14              case -1:
15                      printf("Fork failed"); return -1; break;
16              case 0:
17                      a=3;
18                      b=5;
19                      printf("1: a: %d b: %d\n",a,b);
20                      break;
21              default:
22                      sleep(3);
23                      b=a+1;
24                      printf("2: a: %d b: %d\n",a,b);
25                      break;
26      }
27      a++;
28      b--;
29      printf("3: a: %d b: %d\n",a,b);
30      return 0;
31 }
```

1: a:3 b:5
3: a:4 b:4
2: a:0 b:1
3: a:1 b:0

immer gleich

zugriff auf gleiche ressourcen
schutz mit mutex

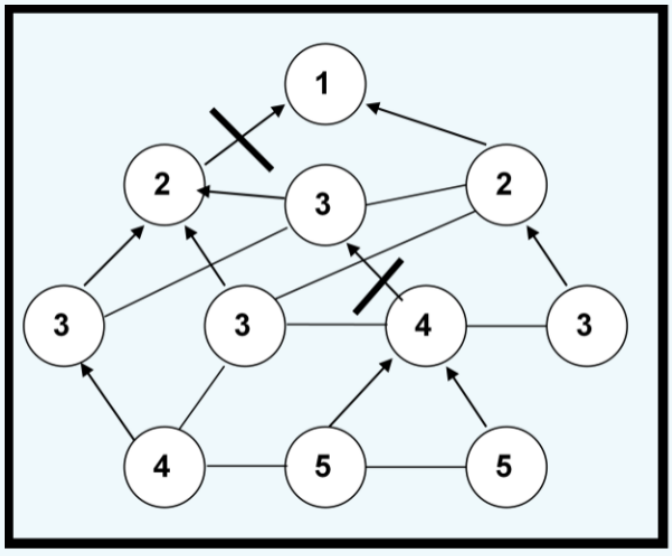
siehe oberes beispiel

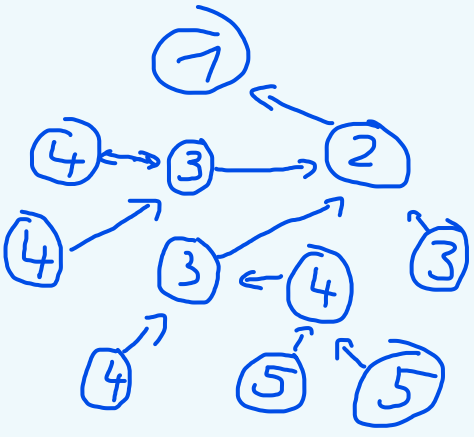
NTP

Reorganisieren Sie das folgende NTP Netzwerk, sodass der Ausfall der Verbindungen (markiert mit einem dicken Querstrich) bestmöglich kompensiert wird.

Legende:

- Kreis mit Ziffer: Knoten mit entsprechendem Stratum
- Pfeil: verbundener Peer
- Linie: mögliche Netzwerkverbindung





getopt

1. Ein Programm soll als Optionen -r, -i <INPUTFILE> und -o <OUTPUTFILE> haben. Wie muss die getopt() Funktion aufgerufen werden, um die Kommandozeile zu analysieren?
2. Wie können INPUTFILE und OUTPUTFILE weiter verarbeitet werden?

IPC

wie mailbox, nachricht wird abgelegt und man kann weiter machen

Was passiert, wenn eine Message Queue beschrieben wird und kein lesender Prozess wartet? Was passiert bei einer Named Pipe? Welchen Vorteil bietet eine Message Queue gegenüber einer Named Pipe?

bei named pipes wird gewartet bis gelesen wird; der schreibende prozess blockiert;
in manchen fällen kann der empfangende prozess ein SIGPIPE signal empfangen (muss man implementieren)

Vorteil: message queue blockier nd
nachteil named pipe muss schreib und leseprozess synchronisieren um blockaden zu vermeiden

LDAP

sehr ähnlich zu erstem ldap filter

1. Geben Sie einen syntaktisch korrekten LDAP Suchfilter an, der alle Einträge des LDAP Verzeichnisses mit der Suchbasis „dc=technikum-wien,dc=at" liefert, deren Nachname (Attribut sn) mit D oder F beginnen und die in der Organisationseinheit (Attribut ou) BMR oder BWI zu finden sind.
2. Wie unterscheidet sich die Ausgabe wenn die Abfrage nicht anonym sondern mit Authentifizierung durchgeführt wird?

Socketprogrammierung

1. Analysieren Sie folgendes C-Programm. Welche Fehler können Sie identifizieren (Zeilennummer + Fehlerbeschreibung)? Sie müssen den Code nicht ändern oder erweitern, sondern nur die Fehler beschreiben und die resultierenden möglichen Probleme aufzeigen.
2. Klassifizieren Sie den Servertyp des untenstehenden C-Programms anhand der folgenden Kategorien und begründen Sie Ihre Entscheidung:
   - stateful/stateless
   - connection oriented/connectionless
   - iterative/concurrent

```
01  #include "myheader.h" /* alle notwendigen Includes */
02  #define BUF 1024
03  #define PORT 1234
04
05  int main (void) {
06    int cs, ns;
07    socklen_t addrlen;
08    char buffer[BUF];
09    struct sockaddr_in addr, cl;
10
11    cs = socket (AF_INET, SOCK_DGRAM, 0);
12
13    addr.sin_family = AF_INET;
14    addr.sin_addr.s_addr = "10.0.0.1";
15    addr.sin_port = nthl(PORT);
16
17    bind (cs,(struct sockaddr *)&addr,sizeof (addr));
18    listen (cs, 5);
19
20    addrlen = sizeof (struct sockaddr_in);
21
22    while (1) {
23      printf("Waiting for connections...\n");
24      ns = accept ( cs, (struct sockaddr *) &cl, &addrlen );
25      if (ns > 0)
26      {
27        strcpy(buffer,"Please enter your command:\n");
28        send(ns, buffer, strlen(buffer)+1,0);
29      }
30      do {
31        recv (ns, buffer, BUF, 0);
32        printf ("Message received: %s\n", buffer);
33      } while (strncmp (buffer, "quit", 4)  != 0);
34    }
35    close (cs);
36    return EXIT_SUCCESS;
37  }
```

= inet_addr("10.0.0.1");

nths(PORT);

Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

Zeitsynchronisation

1. Berechnen Sie die neue Zeit T_C eines Clients nach dem Algorithmus von Christian. Die Sendezeit des Clients T_0=78002. Die Empfangszeit des Clients T_1=78080. Der Reply des Servers enthält die Serverzeit T_S=82014
2. Skizzieren Sie diesen Request mittels Zeitachsen für Client und Server. Zeichnen Sie die obenstehenden Zahlen und Ihr Ergebnis in die Skizze ein.
3. Wie groß wäre der Fehler bei einer idealen symmetrischen Übertragungsdauer und wie groß bei einer asymmetrischen Übertragungsdauer?

Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

Question 32

Not yet answered

Marked out of 8.00

Flag question

Socketprogrammierung

1. Analysieren Sie folgendes C-Programm. Welche Fehler können Sie identifizieren (Zeilennummer + Fehlerbeschreibung)? Sie müssen den Code nicht ändern oder erweitern, sondern nur die Fehler beschreiben und die resultierenden möglichen Probleme aufzeigen.
2. Klassifizieren Sie den Servertyp des untenstehenden C-Programms anhand der folgenden Kategorien und begründen Sie Ihre Entscheidung:
   - stateful/stateless
   - connection oriented/connectionless
   - iterative/concurrent

```
01  #include “myheader.h” /* alle notwendigen Includes */
02  #define BUF 1024
03  #define PORT 6543
04
05  int main (void) {
06    int cs, ns;
07    socklen_t addrlen;
08    char buffer[BUF];
09    struct sockaddr_in addr, cl;
10    pid_t pid;
11
12    cs = socket (AF_INET, SOCK_STREAM, 0);
13
14    addr.sin_family = AF_INET;
15    addr.sin_addr.s_addr = INADDR_ANY;
16    addr.sin_port = PORT;
17
18    bind (cs,(struct sockaddr *)&addr,sizeof (addr));
19    listen (cs, 5);
```

1. Beschreiben Sie die exakte Ausgabe auf stdout. Ist die Ausgabenreihenfolge variabel oder immer ident? Begründen Sie ihre Antwort.
2. Erweitern Sie das Programm entsprechend, dass keine Zombie Prozesse entstehen können.

```c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int main()
{
      pid_t pid;
      int a=0;
      int b=0;

      pid = fork();
      switch (pid)
      {
              case -1:
                      printf("Fork failed"); return -1; break;
              case 0:
                      a=3;
                      b=5;
                      printf("1: a: %d b: %d\n",a,b);
                      exit(0);
              default:
                      b=a+1;
                      printf("2: a: %d b: %d\n",a,b);
                      break;
      }
      a++;
      b--;
      printf("3: a: %d b: %d\n",a,b);
      return 0;
}
```

getopt

1. Ein Programm soll als Optionen -q und -c CONFIGDATEI sowie die Angabe eines Verzeichnisses haben. Wie muss die getopt() Funktion aufgerufen werden, um die Kommandozeile zu analysieren?
2. Wie ermitteln Sie den Verzeichnisnamen in diesem Beispiel?

bereits gelöst

IPC

Eignen sich Signale zur Kommunikation zwischen Prozessen und zur Übermittlung größerer Datenmengen? Warum / warum nicht?

bereits gelöst