

- 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.
- Klassifizieren Sie den Servertyp des untenstehenden C-Programms anhand der folgenden Kategorien und begründen Sie Ihre Entscheidung:
 - o stateful/stateless
 - o connection oriented/connectionless
 - o iterative/concurrent

exit() = stdlib.h

wait() = sys/wait.h

```
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"; htonl(INADDR_ANY) inet_pton(AF_INET, "127.0.0.1", addr)
16     addr.sin_port = PORT; htons(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 (strcmp (buffer, "quit", 4) != 0);
31             close(cs); close(ns); exit(0);
32         } waitpid(child PID here) oder wait()
33     }
34     close (cs);
35     return EXIT_SUCCESS;
```

kein error handling bei allen socket functionen

htons() = host to network short

keine break condition (endloschleife)

keine error funktion bei -1 oder parent

int err = recv()

if err == -1
return -1;

buffer wird nicht buffer[BUF] = '\0'

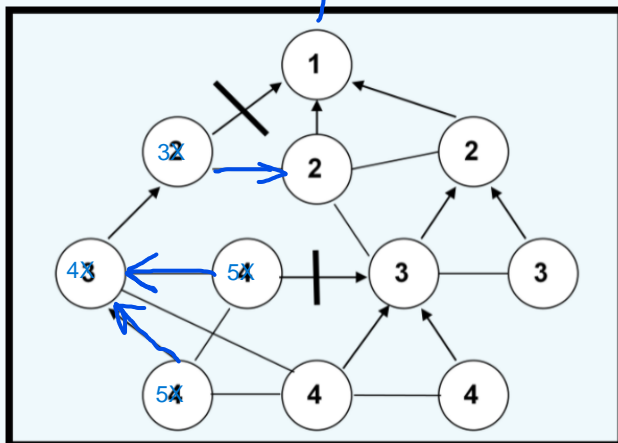
Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

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?

```
ldapsearch -x -LLL -b "dc=technikum-wien,dc=at" "(&(|(sn=M*)(sn=N*))(|(ou=BMR)(ou=BWI))))"
-x simple authentication ohne SASL      AND = (&())... -D <username>
-LLL read formatation                  OR = (|())      -W fordert password eingabe
-b searchbase
```

```
ldapsearch -x -LLL -D "uid=username,ou=people,dc=technikum-wien,dc=at" -W (wenn genauer gesucht werden soll noch -b einfügen) "(&(|(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?

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 Cristian. 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.

sleep sollte nicht benutzt werden lieber mutex oder andere synchronisationsmethoden

```
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 }
```

1:
3:
2:
3:

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:

- ☒ **Implementierte Object Classes:** A, B, C
Gesetzte Attribute: a, x, d, c, b, y
- ☐ **Implementierte Object Classes:** A
Gesetzte Attribute: a, c, b, g
- ☐ **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

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

↓ ↶ ↷ B I A ▾ I ≡ ≡ ≡ ≡ 🔗 ⚙️ 🖼️ </> ✖

ntohs() = network to host short

IPC

PID → [] → []

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

pipes(pipeArray)

mkfifo()

↓ ↶ ↷ B I A ▾ I ≡ ≡ ≡ ≡ 🔗 ⚙️ 🖼️ </> ✖

IPC

Pipes, shared memory, signals, message Queues

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

↓ ↶ ↷ B I A ▾ I ≡ ≡ ≡ ≡ 🔗 ⚙️ 🖼️ </> ✖

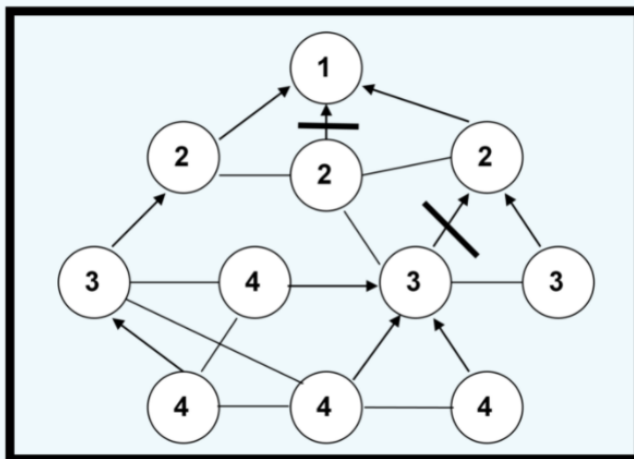
Lösen Sie die Aufgabe am Zettel und geben Sie den Zettel mit Namen und UID/Personenkennzeichen ab!

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



↓ ↶ ↷ B I A ▾ I ≡ ≡ ≡ ≡ 🔗 ⚙️ 🖼️ </> ✖

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

Socketprogrammierung

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



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?



IPC

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

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.

```
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();           parentPID > 0, childPID == 0
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 }
```

IPC

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



Socketprogrammierung

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



Socketprogrammierung

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?



Socketprogrammierung

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



IPC

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.

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?



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 Cristian. 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.

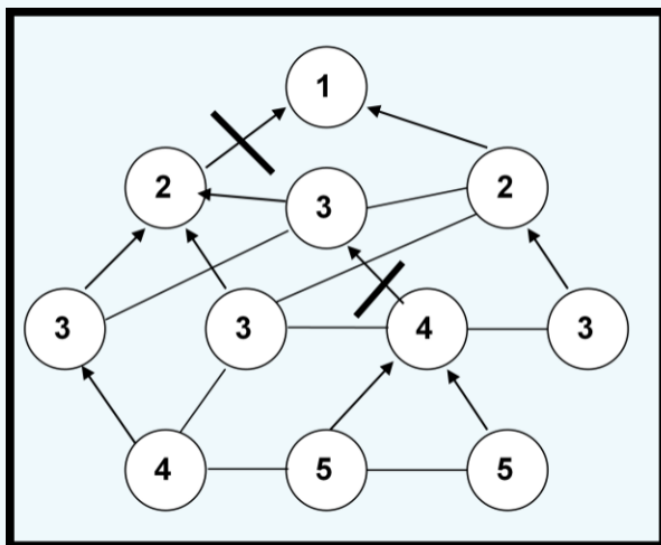
```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int main()
6 {
7     pid_t pid;
8     int a=0;
9     int b=0;
10
11     pid = fork();
12     switch (pid)
13     {
14         case -1:
15             printf("Fork failed"); return -1; break;
16         case 0:
17             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 }
```

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

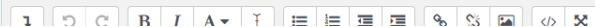


getopt

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

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?



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 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 = ntohs(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 (strcmp (buffer, "quit", 4) != 0);
34     }
35     close (cs);
36     return EXIT_SUCCESS;
37 }
```

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

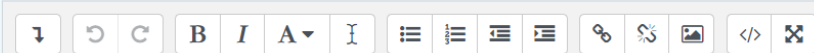
```
01 #include "myheader.h" /* alle notwendigen Includes */
02 #define BUF 1024
03 #define PORT 6543
04
05 int main (void) {
06     int cs, ns;
07     socklen_t addrlen;
08     char buffer[BUF];
09     struct sockaddr_in addr, cl;
10     pid_t pid;
11
12     cs = socket (AF_INET, SOCK_STREAM, 0);
13
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.

```
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 }
```

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?



IPC

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