

Aufgabe 3: Prozesse und Kommunikation über Message Queues

* Ziele dieser Aufgabe sind:

- Arbeiten mit Prozessen: generieren, kontrolliert beenden
- Kommunikation über Message Queues
- Verwendung von Semaphoren
- Programmtechnische Umsetzung nach üblichen Regeln (.h-Datei verwenden, Funktionsprototypen verwenden, Abfangen von Fehlern bei Systemcalls usw.)

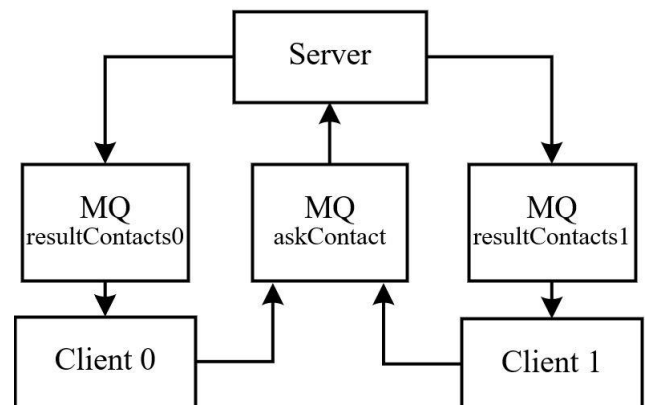
* Client-Server-Anwendung „Kontaktdaten-Auskunft“

Clients fragen über eine MQ einen Server nach Kontaktdaten. Der Server übergibt an die anfragenden Clients über MQs die Auskunft.

Das Vorbild zu dieser Aufgabe besteht auf der „Client-Server-Anwendung Zeitauskunft“ der Vorlesung. Zur Erleichterung der Ausarbeitung wird eine Rumpfversion BS-A2-2021-2022-Rumpf.c bereitgestellt, die ergänzt und ggf. modifiziert werden muss.

* Überblick

- Es gibt einen Server-Prozess und eine Anzahl von MAXCLIENTS Client-Prozesse.
- Die MQ askContact wird für alle Anfragen verwendet
- Pro Client eine gibt es eine MQ um die Antwort entgegen zu nehmen: resultContacts[MAXCLIENTS]



* Der Elternprozess hat folgende Aufgaben:

- Kontaktdaten anlegen: die mitgelieferte Funktion fillContacts ist anzuwenden. Es wird eine Anzahl CONTACTS_N von Kontaktdatensätzen zufällig aus einem Vorrat von Daten gefüllt. Die Kontaktdatensätze haben dieses Format:

```

typedef struct {
    int    counter;           // Zahl der Kontakte
    char   Nachname [CONTACT_N][NAMELENGTH]; // Speicher für Nachname
    char   Vorname  [CONTACT_N][NAMELENGTH]; // Speicher für Vorname
    char   email    [CONTACT_N][3*NAMELENGTH]; // Speicher für email
    int    stop;             // Stop-Flag
} contactsStruct;
  
```

- Semaphore semPrintf einrichten. Diese Semaphore ist zu verwenden, wenn Konsolenausgaben gemacht werden.
- MQs installieren s.o.: askContact und resultContacts[MAXCLIENTS]
- Einen Handler für SIGCHLD einrichten
- MAXCLIENTS Kindprozesse erzeugen und mit Funktion clientCode (nr) losschicken, wobei int nr die Nummer des jeweiligen Clients ist.
- Der Elternprozess führt serverCode aus.

- Wenn alle Kindprozesse beendet sind, beendet sich auch der Elternprozess im SIGCHLD-Handler.

*** Die Funktion `clientCode` hat diese Aufgaben:**

- Jeder Kindprozesse führt die Funktion `clientCode` aus (s.o.).
- Jeder Client-Prozess soll eine ANZAHL Frage-Nachrichten über `askContact` an den Server senden. Die Frage-Nachricht besteht aus einer zufälligen Ganzzahl zwischen 0 und `CONTACT_N`, die als Index zur Auswahl eines Datensatzes dient.
- Über die passende `resultContacts`-MQ wird die Antwort-Nachricht ausgelesen und auf der Konsole ausgegeben, Es sollen *alle* Daten eines Datensatzes ausgegeben werden (Vorname, Nachname, E-Mail-Adresse). Die Konsolenausgabe ist über eine Semaphore zu steuern.
- Nachdem alle ANZAHL -Anfragen und -Antworten verarbeitet wurden, beendet sich der jeweilige Kindprozess.

*** Die Funktion `serverCode` hat diese Aufgaben:**

- In einer Endlos-Schleife werden Frage-Nachrichten aus der MQ `askContact` ausgelesen. Die Frage-Nachricht besteht aus einer zufälligen Ganzzahl zwischen 0 und `CONTACT_N`, die als Index zur Auswahl eines Datensatzes dient.
- Die Antwort ist mittels der passenden `resultContacts`-MQ zu versenden. Es sollen *alle* Daten eines Datensatzes versendet werden (Vorname, Nachname, E-Mail-Adresse).

*** Dies ist die Funktionalität des SIGCHLD-Handlers:**

- Ein `wait` pro Aufruf beseitigt einen Zombie-Prozess
- Wenn keine Kindprozesse mehr vorhanden ist
 - ◆ Alle MQs löschen
 - ◆ Semaphore löschen
 - ◆ Elternprozess beenden

*** Ausgabe einer beispielhaften Programmausführung**

Je nachdem, welche Debugging-Ausgaben gemacht werden und je nach Auslastung der CPU können Umfang und Reihenfolge der Ausgabe variieren.

```
Client 3: nach Fork
Client 3: Beginn Client3
Server: Frage von Client 3 nach Datensatz 8
Client 3: Ergebnis von Server: 8 Hugo Schwarz Hugo.Schwarz@gmail.com
Client 2: nach Fork
Client 2: Beginn Client2
Server: Frage von Client 2 nach Datensatz 8
Client 2: Ergebnis von Server: 8 Hugo Schwarz Hugo.Schwarz@gmail.com
Client 1: nach Fork
Client 1: Beginn Client1
Server: Frage von Client 1 nach Datensatz 5
Client 1: Ergebnis von Server: 5 Mia Schwarz Mia.Schwarz@web.de
Server: Frage von Client 3 nach Datensatz 6
Client 3: Ergebnis von Server: 6 Mia Weiss Mia.Weiss@posteo.de
Server: Frage von Client 2 nach Datensatz 8
Client 2: Ergebnis von Server: 8 Hugo Schwarz Hugo.Schwarz@gmail.com
Server: Frage von Client 1 nach Datensatz 8
Client 1: Ergebnis von Server: 8 Hugo Schwarz Hugo.Schwarz@gmail.com
Server: Frage von Client 3 nach Datensatz 5
Client 3: Ergebnis von Server: 5 Mia Schwarz Mia.Schwarz@web.de
Server: Frage von Client 2 nach Datensatz 4
Client 2: Ergebnis von Server: 4 Mia Maier Mia.Maier@web.de
Server: Frage von Client 1 nach Datensatz 4
Client 1: Ergebnis von Server: 4 Mia Maier Mia.Maier@web.de
---in sigChildHandler--nclients=3
---Zombie geloescht---
---in sigChildHandler--nclients=2
---Zombie geloescht---
---in sigChildHandler--nclients=1
---Zombie geloescht---
```

*** Fragen**

- Welchen Speicherbedarf hat ein Objekt vom Typ `contactsStruct`?
- Wie lange dauert die komplette Ausführung des Programms vom Start bis zum Ende aller Prozesse?

* Bedingungen zur Lösung

- Systemcalls sollen i.d.R. eine Fehlerbehandlung mittels der Wrapperfunktionen verwenden.
- Es ist ein Sequenzdiagramm zum Ablauf des Programms (Anfang bis Ende) anzufertigen nach einem Muster in der Anlage. Dabei gelten die Parameter ANZAHL=1 und MAXCLIENTS=2

* Abgabedokument

- Die Ausarbeitungen der Aufgabe muss in elektronischer Form in die ILIAS-Lernplattform eingereicht werden.
- Die Abgabe besteht aus dem Abgabedokument und aus den Programmen als gezippte Datei A3.zip.
- Das Abgabedokument enthält
 - Ein Inhaltsverzeichnis
 - Eine verbale Beschreibung der Vorgehensweise zur Lösung der Aufgaben
 - Die eingebundenen und kommentierten Programmcodes.
 - Die Dokumentation der Programmausführung (ist mit Screenshots zu belegen).
 - Das Sequenzdiagramm
- Dabei ist drauf zu achten, dass die Authentizität der Ausführung (Datum, Nutzerkonto) erkennbar ist. Eine Möglichkeit diesbezüglich besteht darin, den Prompt vor Ausführen des eigenen Programms entsprechend zu setzen:
- Das Abgabedokument soll im PDF-Format sein.

Anhang: Muster-Sequenzdiagramm

