Aufgabenblatt 4
Echtzeitsysteme

Institut: Berliner Hochschule für Technik
Dozent: Prof. Dr. Christian Forler
Url: `https://lms.bht-berlin.de/`
Email: `cforler(at)bht-berlin.de`
SoSe 2022

Aufgabe 1 (8 Punkte) Parallele Quersummenberechnung I
Erstellen Sie sich mit dem Kommando
`i=0; while [ $i -lt 1000 ]; do echo $(( ($RANDOM%10000)+1 )); i=$(($i+1)); done`
oder mittels `https://www.random.org/integers/` eine Liste von 1.000 Zufallszahlen zwischen 1 und 10.000 und speichern sie diese unter `number.txt` ab.

a) Schreiben Sie eine Funktion `int sum_of_digits(char *num)` welche die Quersumme einer Zahl `num` berechnet. Beispielsweise ist die Quersumme von `"1234"` gleich 10. Handelt es sich bei `num` nicht um eine Zahl gibt die Funktion `-1` zurück. Beispiel: `sum_of_digits("not a number") == -1`. Testen Sie diese Funktion ausgiebig.

b) Schreiben Sie eine Funktion `char *read_line(FILE *f, ssize_t *len)` welche eine Zeile von dem Datenstrom f liest und zurückgibt und in `len` dessen Länge schreibt. Diese Funktion soll die POSIX Funktion `getline()` verwenden. Testen Sie diese Funktion ausgiebig.

c) Schreiben Sie eine Funktion `char **read_file(const char *filename)` welche eine Textdatei liest und ein Array zurückliefert, welche die Zeilen der Datei enthält, d.h. jeder Eintrag in dem Array enthält genau eine Zeile. Verwenden Sie zu implementation die Funktion `read_line()` aus dem vorherigen Aufgabenteil. Testen Sie diese Funktion ausgiebig.

d) Schreiben Sie ein Programm welche die Datei `number.txt` als Kommandozeilenparameter einliest, von jeder Zahl die Quersumme berechnet und am Ende die Summe aller Quersummen ausgibt. Das Programm soll die Funktion `int sum_of_digits()` aus dem Aufgabenteil a) und `read_file()` aus dem Aufgabenteil c) verwenden. Testen Sie diese Funktion ausgiebig.

e) Schreiben Sie eine alternative Lösung für Aufgabenteil d) welche mittels `fork()` 5 Prozesse startet. Teilen Sie die Zahlen gerecht auf die Kinder auf. Die Kinder sollen parallel arbeiten. Dabei soll jedes Kind die Quersumme seiner Zahlen berechnen und sein Zwischenergebnis mit Hilfe einer Pipe an den Elternprozess übertragen. Der Elternprozess soll die Zwischenergebniss aufsummieren und dann das Endergebnis (Summe aller Quersummen) ausgeben. Gehen Sie dazu wie folgt vor

   1. Schreiben Sie eine Funktion `int child_sum(char **numbers, int offset, int count)` welche die Summe eine Kindes zurückgibt. Die

Summe errechnet sich aus den `count` Quersummen aller Einträge von `numbers`, beginnend bei dem Index `offset` (`numbers[offset]`). Testen Sie diese Funktion ausgiebig.

2. Schreiben Sie eine Funktion `void gen_workers(char **numbers, int pfd[], int n)`, welche `n` Kinder generiert die, jedes der Kinder soll mit der Funktion `child_sum()` aus dem vorherigen Aufgabenteil sein Zwischenergebnis berechnen und dieses mit Hilfe der Pipe `pfd` zu seinem Elternprozess senden. Testen Sie diese Funktion ausgiebig.

Aufgabe 2 (8 Punkte) Das FIFO Orakel
Bei dieser Aufgabe soll man ein Orakel implementieren welches über FIFOs mit Fragestellern kommuniziert. Das Orakel soll Anfragen der Form `PID_DES_FRAGESTELLERS:FRAGE` (Beispiel: `123:Are you awesome?`) verarbeiten können. Alle anderen Nachrichten sollen ignoriert werden. Gehen Sie wie folgt vor.

a) Bei dem Orakel `oracle.c` handelt es sich um eine Server-Programm welches wie folgt verhalten soll.

- Es legt einen FIFO `ask.me` an, öffnet diesen zum lesen und wartet auf eine eintreffende Nachricht.
- Bei dem Eintreffen einer Nachricht soll eine Funktion aufgerufen werden, welche diese behandelt.
- Die Antwort schreibt das Orakel in den FIFO `awnser.PID_DES_FRAGESTELLERS`.
- Nachrichten die nicht auf ein Fragezeichen enden sollen mit `Dies ist keine Frage.\n` beantwortet werden.
- Fragen die auf einen Vokal enden, sollen mit der Zeichenkette `Yes.\n` beantwortet werden, alle anderen Fragen mit der Zeichenkette `No.\n`.
- Nach dem beantworten der Frage soll das das Orakel die beiden FIFOs schließen und den FIFO `ask.me` anschließend ereneut öffnen und auf das eintreffen der nächsten Nachricht warten.

b) Schreiben Sei ein Fragesteller Programm (`client.c`), das als Kommandozeilenparameter eine Frage übergeben bekommt und dann wie folgt vorgeht.

1. Es legt den FIFO `awnser.PID_DES_FRAGESTELLERS` an.
2. Es öffnet den FIFO `ask.me` und schreibt die Nachricht der Form `PID_DES_FRAGESTELLERS:FRAGE` dort hinein.
3. Es öffnet den FIFO `awnser.PID_DES_FRAGESTELLERS` und liest daraus die Antwort des Orakels, welche es im Anschluss auf dem Terminal ausgibt.
4. Es schliet beide geöffneten FIFOs und löscht im Anschluss den FIFO `awnser.PID_DES_FRAGESTELLERS`

Unterteilen Sie ihre Programm in sinnvolle Funktionen welche alle weniger als 25 Anweisungen enthalten.