Übungen zu Prozesse und Interprozesskommunikation

1. IPC: Anonyme Pipes

Schreibe folgendes Programm:

- Der laufende Prozess soll mit der Bibliotheksfunktion für den Systemaufruf „fork“ einen Kindprozess erzeugen. Die Prozesse sollen mit Hilfe von anonymen Pipes miteinander kommunizieren können.
- Der Elternprozess liest zwei Zahlen ein und gibt die so entstandene Bruchzahl am Bildschirm aus. Schreibe für die Ausgabe eine Funktion „ausgabe_bruch“ mit zwei Übergabeparametern.
- Den Zähler und den Nenner schickt er dem Kindprozess.
- Der Kindprozess berechnet den größten gemeinsamen Teiler mit Hilfe des Euklidischen Algorithmus und sendet ihn zum Vaterprozess zurück. Schreibe für den Euklidischen Algorithmus eine eigene Funktion mit zwei Übergabe- und einem Rückgabeparameter.
- Abschließend gibt der Vaterprozess die bestmöglich gekürzte Bruchzahl aus. Eine Bruchzahl ist bestmöglich gekürzt, wenn sowohl der Zähler als auch der Nenner durch den größten gemeinsamen Teiler dividiert werden.

```
Die Pipe 'pipe1' wurde angelegt.
Die Pipe 'pipe2' wurde angelegt.
Kindprozess mit PID: 6437
Elternprozerss mit PID: 6436
[Vaterprozess] Geben Sie einen Zähler ein: 32
[Vaterprozess] Geben Sie einen Nenner ein: 12
Die eingegebene Bruchzahl ist:
32
-----
12
Empfangene Daten: 32.
Empfangene Daten: 12.
Der vom Kind berechnete ggT: 4
Die gekürzte Bruchzahl ist:
8
-----
3
```

Erklärung des Euklidischen Algorithmus zum Berechnen des $ggTs$:
Den größten gemeinsamen Teiler zweier natürlichen Zahlen kann man mit Hilfe des Euklidischen Algorithmus schnell berechnen. Dazu ersetzt man so lange die größere der beiden Zahlen durch die Differenz der größeren und der kleineren Zahl, bis die Zahlen gleich sind. Der ggT ist dann diese Zahl.

Zum Beispiel: Berechne den ggT der Zahlen $a = 12$ und $b = 32$.

a	b
12	32
12	20
12	8
4	8
4	4

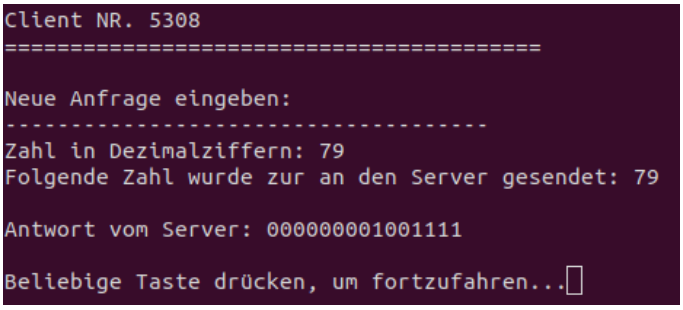
Der ggT der Zahlen 12 und 32 ist 4.

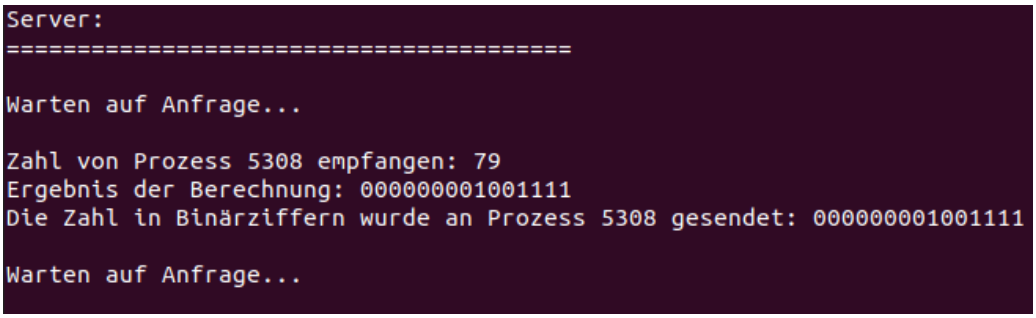
2. IPC: Nachrichtenwarteschlangen

Simuliere eine Client-Server-Kommunikation mittels Nachrichtenwarteschlangen. Beliebig viele Clients können einem Server eine Anfrage zur Umrechnung einer natürlichen Zahl in Dezimalziffern in eine Zahl in Binärziffern schicken. Der Server soll die Zahl umwandeln und diese Zahl dem entsprechenden Client zurückschicken. Der Client gibt die Antwort vom Server am Bildschirm aus. Die Identifikation der Clients soll mittels ihrer Prozess-ID geschehen.

Hinweis: Mit der Funktion „atoi“ kann ein String in einen Integer umgewandelt werden.

Das Programm könnte zum Beispiel so aussehen:


Client NR. 5308
Neue Anfrage eingeben:
Zahl in Dezimalziffern: 79
Folgende Zahl wurde zur an den Server gesendet: 79
Antwort vom Server: 000000001001111
Beliebige Taste drücken, um fortzufahren...


Server:
Warten auf Anfrage...
Zahl von Prozess 5308 empfangen: 79
Ergebnis der Berechnung: 000000001001111
Die Zahl in Binärziffern wurde an Prozess 5308 gesendet: 000000001001111
Warten auf Anfrage...