

## Praktikum zu Betriebssysteme

---

### Vorabhinweis zu allen Aufgaben:

Bitte fangen Sie im vernünftigen Umfang mögliche Fehler ab und dokumentieren Sie den Quelltext ausreichend. Insbesondere sollte ein Kopf vorhanden sein, der folgendes enthält:

- Name des Autors, Erstellungsdatum, ggf. Versionsnummer,
  - in Stichworten, worum es geht,
  - welchen Funktionsumfang Sie realisiert haben, welche Einschränkungen gelten,
  - Welche Systemfunktionen Sie verwendet haben und wo nähere Information zu diesen zu finden ist,
  - Wie Ihr Programm zu handhaben ist.
- 

Thema: Threads

### **Aufgabe 4-1:**

Schreiben Sie bitte einen thread `main`, der zwei threads erzeugt und auf deren Beendigung wartet. In dem einen thread soll ein Zeichen 'x'  $n_1$  mal, in dem anderen ein Zeichen 'y'  $n_2$  mal nach `stdout` ausgegeben werden. Das Zeichen und die Anzahl soll den threads bei ihrer Erzeugung in einer Struktur `char_print_parms` übergeben werden. Für die Ausgabe des übergebenen Zeichens soll in beiden threads eine Funktion `void* char_print(void* parameters)` verwendet werden, die bei der Erzeugung der threads übergeben wird.

Hinweis: Verwenden Sie die Funktionen `pthread_create` und `pthread_join`.

### **Aufgabe 4-2:**

Schreiben Sie bitte einen thread `main`, der zwei threads erzeugt und auf deren Beendigung wartet. Der eine thread soll sich alle 3 sec melden, der andere alle 5 sec.

Variieren Sie bitte das Thema, indem Sie ein einfaches multi-threaded Serverprogramm schreiben. So könnten mittels eines Zufallsgenerators Ereignisse erzeugt werden, die Client-Anforderungen bedeuten. Das Server-Programm erzeugt für jede 'Anforderung' einen Thread. Dieser könnte sich melden, daß er da ist, die erhaltene Aufgabe bearbeitet hat und danach terminieren.

Details der Ausgestaltung sind Ihrer Kreativität überlassen.

---

**Aufgabe 4-3:**

Schreiben Sie ein Programm, in dem der main thread einen zweiten thread erzeugt. Der main thread wartet auf Eingabe von der Tastatur. Wenn der Benutzer die Enter-Taste drückt, soll er den zweiten thread beenden und eine entsprechende Meldung ausgeben. Der zweite thread befindet sich einer Endlosschleife, in der er 5 sec schläft und dann eine Meldung ausgibt. Kann der zweite thread periodisch seine Meldung ausgeben während der main thread auf Eingabe von der Tastatur wartet?

Kann der main thread seine Aufgabe erfüllen, Enter von der Tastatur entgegennehmen, den zweiten thread terminieren und eine Meldung ausgeben während der zweite thread sich gerade am Anfang eines 5 sec Schlafes befindet?