

2. Prozesse und Threads

2.1 Prozesse

Aufgabe 1:

Schauen Sie sich in den man-Pages folgende Kommandos an und beschreiben Sie diese jeweils mit einem Satz:

- fork
- execl
- waitpid
- clone
- system

Aufgabe 2:

- Schreiben Sie ein Python-Programm,
 - dem Argument übergeben wird,
 - das in einer Endlosschleife das Argument ausgibt und sich dann 10 Sekunden schlafen legt (sleep).
- Starten Sie das Programm zweimal als Shell-Hintergrundprozess.
- Senden Sie einem der Programme ein STOP-Signal, danach ein CONT-Signal.

Fragen:

- was beobachten Sie ?
- welchen Status haben die Prozesse, während sie schlafen (ps).
- Senden Sie einem Prozess ein kill-Signal (TERM).

Frage: was beobachten Sie ?

- Senden Sie einem Prozess, der Ihnen nicht gehört, ein 'kill -9'.

Frage: was beobachten Sie ?

- Schreiben Sie ein Programm, das mittels fork 3 Childs erzeugt:
 - der Vaterprozess soll 200mal die PIDs seiner Childs ausgeben: PID1, PID2, PID3
 - die Kind-Prozesse sollen jeweils ihre eigene PID 200 mal ausgeben: „Kind mit PID=XXX“

Frage: Was beobachten Sie bezüglich der Reihenfolge der Prozessausführung?

Aufgabe 3:

Warum gibt es die Zustandsübergänge

- blockiert -> laufend und
- bereit -> blockiert

nicht ?

2.2 Threads

Aufgabe 4:

Suchen und beschreiben Sie in 1-2 Sätzen die Python-Entsprechungen zu pthread_create() und pthread_join().

Aufgabe 5:

Schreiben Sie ein Python-Programm, das

1. eine globale Variable zu 0 initialisiert,
2. 10 Threads erzeugt, die jeweils 1000 mal eine Schleife durchlaufen,
 - 5 davon addieren jeweils eine 1,
 - 5 davon subtrahieren jeweils eine 1
3. der Hauptthread sammelt mit einem Join alle Threads wieder ein und gibt dies aus:
„Thread Nr. X joined“
4. Zuletzt wird der Inhalt der globalen Variablen ausgegeben

Fragen:

- was beobachten Sie ?
- warum ?