

Arbeitsblatt „Prozesse in UNIX“

Aufgabe 1:

- Programm schreiben, das folgendes leistet:
 - Vaterprozess erzeugt Sohnprozess.
 - Sohn schläft für 10 Sekunden, gibt dann einen Text aus, schläft dann weitere 5 Sekunden und terminiert schließlich.
 - Vater wartet, bis Sohn terminiert, gibt entsprechende Meldung aus und terminiert dann selbst.
 - benötigte Funktionen: `fork()`, `sleep()`, `wait()`, `exit()`
- während der Sohn schläft: Informationen über die Prozesse anschauen.
 - mit Kommando `ps -f`
 - von Interesse:
 - PID (= Process Identifier = Nummer des ausgeführten Prozesses)
 - PPID (= Parent Process Identifier = Nummer des Vaterprozesses)
 - ausgeführtes Programm
 - Tipp: Das Programm als Hintergrundprozess starten.

Aufgabe 2:

- Programm so ändern, dass der Sohn in einer Endlosschleife die natürlichen Zahlen beginnend mit 1 ausgibt und der Vater ihn nach fünf Sekunden löscht.
- weitere benötigte Funktion: `kill()`
- Programm mehrfach ausführen. Kommt der Sohn mit den Ausgaben immer gleich weit?

Aufgabe 3:

- Programm so ändern, dass der Code des Sohns in einer eigenen Datei steht.
- weitere benötigte Funktion: `execv()`

Aufgabe 4:

- Prozess schreiben, der versucht, den `sleep`-Prozess des Benutzers `vogt` (Prozessnummer durch `ps -ef` feststellen) sowie den Prozess Nr. 60000 zu killen.
 - Bei Auftreten eines Fehlers: Fehlernummer und Fehlermeldung im Klartext ausgeben.
- Zur Vorgehensweise siehe „UNIX/Linux-C-Schnittstelle“, Beispielprogramm auf S. 1.