

# Systemprogrammierung: fork und exec

## Klaus Kusche

Das Programm mit **system** und **alarm** soll so umgebaut werden, dass das **system** durch ein "händisches" **fork**, **exec** und **wait** ersetzt wird (**alarm** und das Signal Handling bleiben unverändert).

Der Einfachheit halber ändern wir den Aufruf des Programms:

Der auszuführende Befehl mit allen seinen Optionen und Argumenten wird auf der Befehlszeile nicht mehr als ein einziges Wort angegeben, sondern als ein oder mehrere getrennte Worte (das vereinfacht den **exec**-Aufruf).

Im Detail:

- Das Hauptprogramm ruft in der Schleife gleich zu Beginn **fork** auf. Danach unterscheidet es drei Fälle:
  - Den Fehlerfall (Fehlermeldung, Programmende)
  - Den Vater
  - Den Sohn
- Der Sohn macht sofort ein **exec** des angegebenen Programms (verwende **execvp**). Im Fehlerfall wird eine Fehlermeldung ausgegeben, dann beendet sich der Sohn. Es wäre allerdings nicht sinnvoll, dass der Sohn wegen eines **exec**-Fehlers endet, ohne dafür zu sorgen, dass der Vater ebenfalls endet: Die Schleife im Vater würde sofort wieder einen Sohn starten, dessen **exec** fast sicher wieder mit demselben Fehler fehlschlagen würde, usw..
  - Die "saubere" Möglichkeit wäre, den Fehler durch einen Exitcode (z.B. **127**) anzuzeigen, den Exitcode nach dem **wait** im Vater auszuwerten, und in diesem Fall die Schleife zu verlassen.
  - Wir implementieren zur Übung hingegen die "brutale" Variante: Unmittelbar bevor er sich beendet, killt der Sohn den Vater (du wirst **kill** und **getppid** brauchen, verwende das Signal **SIGTERM**).
- Der Vater wartet mit **wait** auf das Ende des Sohnes. Ein Fehler im **wait** selbst soll das Programm mit Fehlermeldung beenden, ansonsten wird jedes Ende des Sohnes ohne Ausgabe einer Meldung akzeptiert (egal, ob erfolgreich, erfolglos oder durch ein Signal verursacht): Die Schleife zählt eine Programmausführung und macht den nächsten Durchlauf.

Hinweise:

- Prüfe alle zentralen System- und Library-Aufrufe auf Fehler und gib "schöne" Fehlermeldungen aus (mit Programmname, **errno**-Text usw.).