Prof. Dr. Franz Schweiggert   Abteilung Angewandte Informationsverarbeitung   10. Mai 2005
Ralph Guderlei   Blatt 5

Systemnahe Software / Systemnahe Software II (SS 2005)

Abgabetermin: 24. Mai 2005

1 Kill, aber mit Menü! (10 Punkte)

Finden Sie es nicht auch etwas mühsam, bevor Sie mit `kill` einen Prozeß beenden können erst dessen PID mittels `ps` herausfinden zu müssen? Um in Zukunft leichter Prozesse beenden zu können sollen Sie ein Programm schreiben, dass

- eine Liste bestehend aus einer laufenden Nummer und den ersten 50 Zeichen des Programmnames ausgibt,
- nach Eingabe einer Nummer den dazugehörigen Prozess mit `SIGTERM` beendet,
- und mit dem Kommandozeilenargument "-u" gestartet alle Prozesse des Benutzers anzeigt.

Beispiel-Menü:

```
Nummer  Programmname
1       bash
2       /usr/local/bin/../Adobe/Acrobat7.0/Reader/intellin
3       scite hb-master2.c hb-slave.c
4       ./a.out
Nummer bzw. "0" zum beenden:
```

Folgende Vereinfachungen können als gegeben angenommen werden:

- Die Anzahl der laufenden Prozesse ist kleiner als 100
- Der Programmname (Feld COMMAND bei `ps`) ist kürzer als 2048 Zeichen.

2 Lebt er noch? (10 Punkte)

Manchmal ist es interessant zu wissen, ob ein Prozess noch "lebt" oder nicht. Diese Überprüfung soll mittels Signalen implementiert werden. Die Implementierung soll aus zwei separaten Programmen bestehen, eines dient als Ziel der Untersuchung ("slave") und eines führt die Überprüfung durch ("master"). Die Überprüfung läuft folgendermaßen ab: Der `master` sendet an den `slave` das Signal `SIGUSR1`, und der `slave` antwortet mit `SIGUSR2`.

Der `master` erhält die PID des `slave`-Prozesses beim Programmstart, der `slave` muß die PID des `master`-Prozesses bei der Signalverarbeitung den vom Betriebssystem bereitgestellten Daten entnehmen.

Wichtige `man`-Pages:

- ps(1)
- kill(2)
- sigaction(2)

Tipp: Der struct `siginfo_t` enthält zusätzliche Informationen über das empfangene Signal.

Viel Erfolg!