

Systemmanagement und Sicherheit

5. Übung

Aufgabe 1 (Signal-Handling `signal()`)

Schreiben Sie ein C-Programm `sigtest`, das einen einzigen Signalhandler für alle möglichen Signale besitzt. Der Signalhandler soll mittels `signal(3)` aktiviert werden.

Mittels `sleep(3)` soll `sigtest` für eine Minute existieren und innerhalb von main folgenden Return-Code zurückgeben:

- 0, falls kein Signal innerhalb dieser Minute empfangen
- die Signalnummer und seine Beschreibung (siehe `psignal()`), falls durch den Handler ein Signal abgefangen wurde.

Die Kommunikation zwischen Signal-Handler und `main()`, um die Signalnummer mitzuteilen, soll über eine globale `int` Variable `signo` realisiert werden.

Aufgabe 2 (Signal-Handling `sigaction()`)

Schreiben Sie analog zur vorherigen Aufgabe ein Programm `sigtest2`, das den gleichen Mechanismus mit Hilfe von `sigaction(2)` implementiert.

Aufgabe 3 (Signal-Handling (Signale künstlich erzeugen))

Benutzen Sie Ihr Programm `sigtest2` als Grundlage für ein Programm `sigtest3`, das mit einem erhöhten Parameter für `sleep()` arbeitet, um möglichst viele der in der Vorlesung angegebenen Signale zu simulieren. Hierbei bedeutet *Simulieren* nicht, dass mit Hilfe von `kill` oder `kill()` die jeweilige Signalnummer erzeugt wird. Vielmehr sollen beim Ablauf des Programms `sigtest` die Bedingungen erzeugt werden, die zum Senden des Signals führen, etwa bei `SIGABRT` der Aufruf der Funktion `abort()` oder bei `SIGSEGV` der tatsächliche illegale Zugriff auf eine Speicherstelle usw.