



16. Top und Bottom Half in Signal-Handlern (4 Punkte)

Linux unterteilt Interrupt-Handler in eine Top- und eine Bottom-Half, um als unmittelbare Reaktion auf den Interrupt nur die wichtigsten Dinge zu erledigen und weitere Aufgabe aufzuschieben.

a) In dieser Übung übertragen Sie dieses Konzept auf die Signalbehandlung:

- Für die Behandlung des Signals `SIGUSR1` melden Sie eine Funktion namens `top_sig()` als Signal-Handler an.
- Diese soll mit `pthread_create()` einen neuen Thread erzeugen, der die Funktion `bottom_sig()` ausführt. Der Bottom-Handler soll zu Beginn zwei Sekunden schlafen, also mit großer Verzögerung starten.
- Top- und Bottom-Signal-Handler sollen jeweils ausgeben, dass sie arbeiten:

```
printf("\n-- Top: SIGUSR1 erhalten! --\n");  
fflush(stdout);
```


(bzw. „Bottom: ...“ statt „Top: ...“).
- Eine globale Variable soll die Anzahl der erzeugten Threads erzeugen – vor dem Ende des Hauptprogramms sollen alle „Bottom-Half“-Threads wieder eingesammelt werden (vgl. Aufgabe 13).

Nehmen Sie als Ausgangspunkt das Programm aus Aufgabe 15 (letzte Übung) und spalten Sie den Signal-Handler wie beschrieben in zwei Teile. Achten Sie beim Programmieren des Bottom-Handlers darauf, wie Sie die Funktion deklarieren müssen (vgl. Aufgabe 15).

- b) Schreiben Sie eine neue Funktion `tbsignal()` (top-bottom-signal), die `signal()` ersetzen kann und als Argument den Top- und den Bottom-Signal-Handler sowie das Signal akzeptiert.
- c) Ist diese Art der Implementierung eines Signal-Handlers 1:1 mit den Top- und Bottom-Halves bei der Interrupt-Behandlung vergleichbar? Wenn nein, welchen Unterschied gibt es? (Die triviale Antwort, dass ein Interrupt kein Signal ist und entsprechend ein Interrupt-Handler kein Signal-Handler, ist nicht ausreichend ; -) .)
- d) Denken Sie sich ein Anwendungsbeispiel aus, in dem es sinnvoll sein kann, die Signal-Behandlung auf diese Weise in zwei Teile aufzuteilen, und erläutern Sie dieses Szenario.

17. sysenter und sysexit (2 Punkte)

Um einen System Call (Syscall) zu aktivieren, bieten moderne Intel-Prozessoren neben dem Einsprung über `int 0x80` mit `sysenter` eine Alternative. Recherchieren Sie im Internet den Unterschied zwischen der klassischen (und in der Vorlesung vorgestellten) Variante über `int 0x80` und der `sysenter`-Variante und erklären Sie den Vorteil der neuen Variante.

Die Lösungen schicken Sie mir bitte per E-Mail bis zum 12.05.2011 mit dem Betreff „BS-2011, Übung 6, NAMEN“.

Hinweis zum Stand der Korrekturen:

Ich habe jetzt alle Übungen 1 und 2 korrigiert und Feedback-Mails verschickt und fange in den nächsten Tagen damit an, Übung 3 zu korrigieren.