Universität Ulm - Abteilung Angewandte Informationsverarbeitung
Prof. Schweiggert und Dr. J. Mayer mit H. Braxmeier und C. Ehrhardt, Sommersemester 2004

4. Übungsblatt zur Vorlesung Systemnahe Software II

Abgabetermin: Montag, 24.05.2004

Signale und Signalhandler *(10 Punkte)*

Morse Code

Beim Morse Code werden Zahlen und Buchstaben in eine Folge von langen und kurzen Signalen (notiert als "`.`" bzw. "`-`") übersetzt, so entspricht etwa die Signalfolge "kurz lang" bzw. "`.-`" dem Buchstaben "A" oder die Signalfolge "kurz kurz lang" bzw. "`..-`" dem Buchstaben "U". Bis zu 6 einzelne Signale können notwendig sein, um einen Buchstaben zu kodieren. Das Ende eines Buchstabens ist an einer kurzen Übertragungspause (notiert als Leerzeichen) zu erkennen. "`... -- ...`" steht dann etwa für "SMS".

Hilfsfunktionen

Die Datei `morse.h`, die auf der Homepage der Vorlesung zur Verfügung steht, definiert bereits zwei Funktionen:

- `char * encode (unsigned char c);`
  Diese Funktion übersetzt den Buchstaben c in den zugehörgen Morse Code. Der Rückgabewert ist ein String, der aus den Zeichen "`.`" und "`-`" wie oben beschrieben besteht. Wenn für das angegebene Zeichen kein Morse Code definiert ist, wird ein `NULL`-Zeiger zurückgegeben.

- `unsigned char decode (char *p);`
  Diese Funktion leistet das Umgekehrte: Der String p, der nur aus den Zeichen "`.`" und "`-`" bestehen darf, wird als Morse Code interpretiert. Der Rückgabewert ist der zu diesem Morse Code gehörende Buchstabe. Falls der Morse Code ungültig war, wird 0 zurückgeliefert.

Das kurze Beispielprogramm `morse_test.c` zeigt, wie die beiden Funktionen verwendet werden und gibt nebenbei auch noch eine Liste aller Zeichen mit ihrem Morse Code aus.

Aufgabe 1 - Morse Code dekodieren (5 Punkte)

In diesem Blatt soll nun ein Programm geschrieben werden, das Morse Code in Form von Unix-Signalen empfängt und in Buchstaben übersetzt. Dabei soll das Unix-Signal SIGUSR1 einem kurzen Morse-Signal und das Unix-Signal SIGUSR2 einem langen Morse-Signal entsprechen. Statt der Übertragungspause wird das Signal SIGTERM verwendet.
Testet Euer Programm zunächst, indem Ihr ihm von Hand (mit dem Kommando kill) einige Signale schickt. Zu beachten ist, daß ggf. `fflush` verwendet werden muß, damit ein ausgegebener Buchstabe sofort auf dem Bildschirm erscheint.

Aufgabe 2 - Morse Code kodieren (5 Punkte)

Schreibt jetzt ein Programm, das von der Kommandozeile eine Prozeß-ID und einen Text entgegen nimmt. Der Text soll dann als Morse Code kodiert wie oben beschrieben an den angegebenen Prozeß gesendet werden.
Ein erster Versuch wird zeigen, daß der Text nicht richtig ankommt. Warum? Paßt Euer Programm entsprechend an, so daß die Übertragung zumindest meistens funktioniert.

Hinweise

- Durch die Verwendung der Datei `morse.h` können ggf. Warnungen über nicht verwendet Funktionen auftreten. Diese Warnungen können ignoriert werden.
- Kurze Pausen im Programmablauf (einige Millisekunden) können durch eine Endlosschleife realisiert werden, für längere Pausen (ab 1 Sekunde)sollte dagegen `sleep` verwendet werden.
- Der Morse Code definiert weder Leerzeichen noch Newlines. `encode` und `decode` wie oben beschrieben verwenden dagegen einen leeren String (nur ein SIGTERM) für das Leerzeichen.

Viel Erfolg!