

Übungen zu Systemnahe Programmierung in C

Abschnitt 13.2: Hands-On (Stoppuhr)

20.07.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT



```
01 $ ./stoppuhr
02 Press Ctrl+C (SIGINT) to start and stop
03 ^CStarted...
04 1 sec
05 2 sec
06 3 sec
07 4 sec
08 ^CStopped.
09 Duration: 4 sec 132 msec
```

■ Ablauf:

- Stoppuhr startet durch SIGINT Signal
 - Gibt jede Sekunde die bisherige Dauer aus (Format: "3 sec")
- Stoppuhr stoppt bei weiterem SIGINT und gibt Dauer aus
 - Gibt Gesamtdauer inkl. Millisekunden aus (Format: "4 sec 132 msec")
 - Beendet sich anschließend

■ Verwendet intern SIGALRM und alarm(2)

■ Schutz kritischer Abschnitte beachten



1. Signalhandler installieren: `sigaction(2)`

```
01 struct sigaction act;  
02 act.sa_handler = SIG_DFL; // Handlersignatur: void f(int signum)  
03 act.sa_flags = SA_RESTART;  
04 sigemptyset(&act.sa_mask);  
05 sigaction(SIGINT, &act, NULL);
```

2. Signale blockieren/deblockieren: `sigprocmask(2)`

```
01 sigset_t set;  
02 sigemptyset(&set);  
03 sigaddset(&set, SIGUSR1);  
04 sigprocmask(SIG_BLOCK, &set, NULL); /* Blockiert SIGUSR1 */  
05 // kritischer Abschnitt  
06 sigprocmask(SIG_UNBLOCK, &set, NULL); /* Deblockiert SIGUSR1 */
```



3. Auf Signale warten: sigsuspend(2)

```
01 sigprocmask(SIG_BLOCK, &set, &old); /* Blockiert Signale */
02 while(event == 0){
03     sigsuspend(&old); /* Wartet auf Signale */
04 }
05 sigprocmask(SIG_SETMASK, &old, NULL); /* Deblockiert Signale */
```



- Timerkonfiguration mit `alarm(2)`

```
01 unsigned alarm(unsigned seconds);
```

Einmaliger Alarm nach definierter Wartezeit (in Sekunden)

seconds = 0 Alarm abbrechen

- Timerkonfiguration mit `ualarm(3)`

```
01 useconds_t ualarm(useconds_t useconds, useconds_t interval);
```

Erster Alarm nach `useconds` Mikrosekunden,
anschließend alle `interval` Mikrosekunden

useconds = 0 Alarm abbrechen

interval = 0 einmaliger Alarm

- SIGALRM: Timer ist abgelaufen bzw. Alarm eingetreten
 - Standardbehandlung: Programm beenden
 - Eigenen Signalhandler installieren