Aufgabe 08

Tim Zolleis

1. Normaler Fork

```
Terminal Console Scale Console Console Scale Console Scale Console Scale Console Scale Console Console Scale Console Scale Console Scale Console Scale Console Console Scale Console Scale Console Scale Console Scale Console Console Scale Console Scale Console Scale Console Scale Console Console Scale Console Scale Console Scale Console Scale Console Console Scale Console Scale Console Scale Console Console Scale Console Scale Console Console Console Scale Console Cons
```

2. Fork auf dem Target

```
<terminated> signals - remote [C/C++ Remote Application] /home/tim/eclipse-worksp
Last login: Tue Jun 4 16:04:54 2024 from 192.168.1.24
/tmp/remotetest/signals;exit
[?2004htim@target068:~$ /tmp/remotetest/signals;exit
[?2004l
--- Im Elternprozess ---
global var = 1, Speicheradresse : 0x491008
lokal var = 1, Speicheradresse : 0x7e9ec42c
--- Im Kindprozess ---
global var = 1, Speicheradresse : 0x491008
lokal var = 1, Speicheradresse : 0x7e9ec42c
--- Im Kindprozess ---
global var = 2, Speicheradresse : 0x491008
lokal var = 2, Speicheradresse : 0x7e9ec42c
--- Im Elternprozess ---
global var = 1, Speicheradresse : 0x491008
lokal var = 1, Speicheradresse : 0x7e9ec42c
logout
```

3. For mit unterschiedlichen Laufvariablen

Dies kann durch Initialisieren der Variable (i = 0;) behoben werden - je nach Betriebssystem und Speicherplatz kann es zu unterschiedlichen Werten kommen.

```
<terminated> (exit value: -1) signals - local [C/C++ Application] /home/tim,
        Child: i = 10
Parent: i= -1178121709
        Child: i= 11
Parent: i= -1178121707
        Child: i= 12
Parent: i= -1178121705
        Child: i= 13
        Child: i= 14
Parent: i= -1178121703
                               Ī
Parent: i= -1178121701
        Child: i= 15
        Child: i= 16
Parent: i= -1178121699
        Child: i= 17
Parent: i= -1178121697
        Child: i= 18
Parent: i= -1178121695
        Child: i= 19
Parent: i= -1178121693
Parent: i= -1178121691
        Child: i = 20
Parent: i= -1178121689
        Child: i= 21
        Child: i= 22
Parent: i= -1178121687
        Child: i= 23
Parent: i= -1178121685
Parent: i= -1178121683
```

Child: i= 24

/tmp/remotetest/signals;exit

[?2004htim@target068:~\$ /tmp/remotetest/signals;exit

[?2004l

Child: i= 0

Parent: i= 1995784192

Child: i= 1

Parent: i= 1995784194

Child: i= 2

Parent: i= 1995784196

Child: i= 3

Child: i= 4

Parent: i= 1995784198

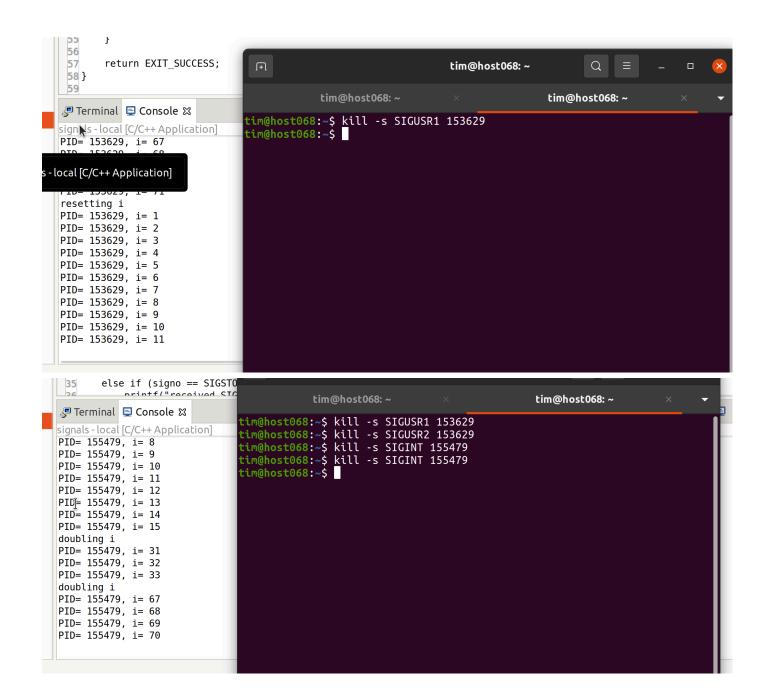
Child: i= 5

Parent: i= 1995784200

Child: i= 6

Parent: i= 1995784202

4. Signale an Prozesse

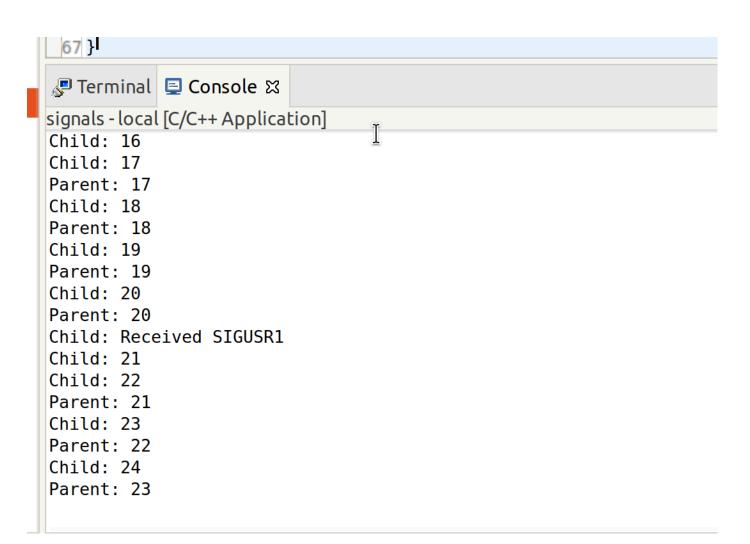


```
printf("\ncan't catch SIGINT\n");
 48
          if (signal(SIGKILL, sig_handler) == SIG_ERR)
 49
               printf("\ncan't catch SIGKILL\n")
 50
 51
          if (signal(SIGSTOP, sig_handler) ==
                                                                                                           tim@host068: ~
               printf("\ncan't catch SIGSTOP\n")
 52
 53
          for (i=0; i*i < 10000; i+=direction){
 54
                                                                             tim@host068: ~
                                                                                                                                  tim@host068: ~
               printf("PID= %d, i= %d\n",pid,i);
 55
                                                            tim@host068:~$ kill -s SIGUSR1 153629
tim@host068:~$ kill -s SIGUSR2 153629
tim@host068:~$ kill -s SIGINT 155479
tim@host068:~$ kill -s SIGINT 155479
tim@host068:~$ kill -s SIGUSR2 155479
tim@host068:~$ kill -s USR2 158146
tim@host068:~$ kill -s SIGUSR2 160215
tim@host068:~$
 56
               sleep (1);
 58
          return EXIT SUCCESS;
 59
 60 }
 61
্জু Terminal 📮 Console 🛭
signals - local [C/C++ Application]
PID= 160215, i= 6
PID= 160215, i= 7
PID= 160215, i= 8
PID= 160215, i= 9
PID= 160215, i= 10
PID= 160215, i= 11
PID= 160215, i= 12
PID= 160215, i= 13
                                            I
switching direction of i
PID= 160215, i= 12
PID= 160215, i= 11
PID= 160215, i= 10
PID= 160215, i= 9
PID= 160215, i= 8
PID= 160215, i= 7
PID= 160215, i= 6
PID= 160215, i= 5
```

5. Signale zwischen Parent und Child

```
#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
void child_sig_handler(int signo){
    if (signo == SIGUSR1){
        printf("Child received SIGUSR1\n");
    else if (signo == SIGUSR2){
        printf("Child received SIGUSR2\n");
    else if (signo == SIGSTOP ) {
        printf("Child received SIGSTOP\n");
    else if (signo == SIGCONT){
        printf("Child received SIGCONT\n");
    else if (signo == SIGHUP){
        printf("Parent wants me to hang up...\n");
        exit(0);
}
void runChild(){
```

```
int count = 0;
    //Handlers
    signal(SIGUSR1, child_sig_handler);
    signal(SIGUSR2, child_sig_handler);
    signal(SIGSTOP, child_sig_handler);
    signal(SIGCONT, child_sig_handler);
    signal(SIGHUP, child_sig_handler);
    while(1){
        printf("Child: %d\n", count);
        count++;
        sleep(1);
    }
}
pid_t global_child_pid;
void parent_sig_handler(int signo){
    if(signo == SIGHUP){
        printf("Parent received SIGHUP\n");
        kill(global_child_pid, SIGSTOP);
    }
    else if(signo == SIGCONT){
        printf("Parent received SIGCONT\n");
        kill(global_child_pid, SIGCONT);
}
void runParent(pid_t child_pid){
    global_child_pid = child_pid;
    //Handlers
    signal(SIGHUP, parent_sig_handler);
    signal(SIGCONT, parent_sig_handler);
    int count = 0;
    while(1){
        count+=10;
        printf("Parent: %d\n", count);
        if(count == 20){
            kill(child_pid, SIGUSR2);
        }
        if(count == 30){
            kill(child_pid, SIGUSR1);
        }
        if(count == 70){
            kill(child_pid, SIGHUP);
            break;
        }
        sleep(1);
    }
}
int main(void) {
        pid_t child_pid;
        switch (child_pid = fork()) {
        case -1:
                printf("Could not fork\n");
                break;
        case 0:
                runChild();
```



```
}
        return 0;
 66
 67 }

▼ Terminal 
□ Console 
□
signals - local [C/C++ Application]
Parent: 27
Child: 28
Parent: 28
Child: 29
Parent: 29
                          I
Child: 30
Parent: 30
Child: Received SIGUSR2
Child: 31
Child: 32
Parent: 31
Parent: 32
Child: 33
Child: 34
Parent: 33
Parent: 34
Child: 35
```

98 ্ৰু Terminal 📮 Console 🛭 <terminated> (exit value: 0) signals - local [C/C++ Application] /ho Child: 2 Parent: 30 Child received SIGUSR1 Child: 3 Parent: 40 Child: 4 Parent: 50 Child: 5 Child: 6 Parent: 60 Child: 7 Parent: 70 Parent wants me to hang up...