8. Signals and Timers

1. Descriptions

In this assignment, you will learn how to set up real-time timers and implement the signal handlers. The example program accepts a command-line argument in the form of `[delay time]:[interval time]`. As shown in the following figure, when we run the example program with the command-line argument “2:1”, the program will create a timer that waits for 2 seconds and begins generating a SIGALRM (No. 14) signal every second.

```
csc@csc-VirtualBox:~/os_2023/7.Signals_and_Timers$ ./timer 2:1
Timer ID: 0 (2:1)
[16:25:05] Got signal 14
[16:25:06] Got signal 14
[16:25:07] Got signal 14
[16:25:08] Got signal 14
^C
```

In this project, your task is to alter the program so it utilizes a signal number distinct from SIGALRM. For example, as depicted in the subsequent figure, the altered program's timer will produce Signal No. 64. Upon receiving this signal, the signal handler will output both the signal number and the corresponding timestamp.

```
csc@csc-VirtualBox:~/os_2023/7.Signals_and_Timers$ ./timer 2:1
Timer ID: 0 (2:1)
[16:25:50] Got signal 64
[16:25:51] Got signal 64
[16:25:52] Got signal 64
[16:25:53] Got signal 64
[16:25:54] Got signal 64
^C
```

> **Signal No. should not be 14**

2. Exercise

You need to complete the following tasks:

- Modify the sigaction() function parameter at line 31, so that it can handle the signal you choose (you need to choose a signal different from SIGALRM). You can refer to the [sigaction()] manual page to learn how to use this function.
- We create a timer by calling timer_create() (line 38). By default, a timer created by timer_create will generate the SIGALRM signal. We can ask it to generate a different signal by setting the sigevent argument in the call of timer_create. Please implement the code in the ToDo section (lines 33-36) to properly set up the sigevent structure for your signal. You can check the [timer_create()] manual page to learn how to use this structure.
- Write a short report explaining what modifications you have done and what phenomena you have observed.
- Briefly describe the difference between a function call and the invocation of a signal handler.

```c
#include <signal.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include "curr_time.h"                  /* Declares currTime() */
#include "itimerspec_from_str.h"        /* Declares itimerspecFromStr() */

static void handler(int sig, siginfo_t *si, void *uc){
    printf("[%s] Got signal %d\n", currTime("%T"), sig);
}

int main(int argc, char *argv[]){
```

```
    struct itimerspec ts;
    struct sigaction  sa;
    timer_t *tidlist;

    if (argc != 2){
        fprintf(stderr, "%s [initial time up]:[interval time up]\n", argv[0]);
        exit(1);
    }
    tidlist = calloc(argc - 1, sizeof(timer_t));
    if (tidlist == NULL)
        perror("malloc");

    /* Establish handler for notification signal */

    sa.sa_flags = SA_SIGINFO;
    sa.sa_sigaction = handler;
    sigemptyset(&sa.sa_mask);
    if (sigaction(SIGALRM, &sa, NULL) == -1)
        perror("sigaction");
/* ToDo:
    struct sigevent sev;
    ...
*/
    itimerspecFromStr(argv[1], &ts);
    if (timer_create(CLOCK_REALTIME, NULL, &tidlist[0]) == -1)
        perror("timer_create");
    printf("Timer ID: %ld (%s)\n", (long) tidlist[0], argv[1]);

    if (timer_settime(tidlist[0], 0, &ts, NULL) == -1)
        perror("timer_settime");

    for (;;)                            /* Wait for incoming timer signals */
        pause();
}
```

3. Submission

Please submit a **zip** file to E3 that contains your program source code and report.

For the program source code part (80%):

- The program must be implemented using C.
- Make sure your code can be compiled on **Ubuntu 22.04**.
- Make sure your output is correct.

For the report part, there are two parts you need to write(20%):

- Part 1: what you modify in the code
- Part 2: describe the difference between function call and signal handler invocation

The name of the zip file should be <student_id>.zip, and the structure of the file should be as the following:

<stduent_id>.zip

|- <student_id>/

|- timer.c

|- hw7.pdf

> **You only need to modify the code in timer.c**

4. Reference

sigaction(2) - Linux manual page (man7.org)

timer_create(2) - Linux manual page (man7.org)

For questions, please contact TA Mr. Chuang <cscneko.cs06@nycu.edu.tw>