System Programming
4rd Laboratory (10th ..13th March 2020)

Signals
I

Implement a program that continuously prints increasing numbers (at the main), and every second resets the counter to zero.
Use the following functions:

- **`alarm`**
- **`sigaction`** or **signal**

Debugging Threads
II

Compile the **lab4-II-bad.c** program (with the name **prog1**) and run it.

- This program receives as an argument the number of threads to be created.
- Each thread should run for 10 seconds, but enters into an infinite loop.
- Kill the running program with Ctrl-C

DO NOT CORRECT THIS PROGRAM

To debug and modify the execution of each thread we will use the debugger (**ddd** or **gdb**).

- Recompile the program using the **-g** option:
  - **gcc prog1.c -pthread -g -o prog1**
- run gdb/ddd with the program as argument:
  - **ddd prog1**
- inside gdb console execute the program with a numerical argument:
  - **run 3**
- All the threads will enter into an infinite loop….
- Stop the program pressing **Crt-C**
- List the available thread
  - menu Status → Threads
  - or issue the command: **info threads**
- Observe this list and select one thread to debug
  - Double click the thread to debug
  - or use the command **thread** to change to one of the threads. For instance:
    - **thread 2**
- Observe where such thread is blocked:
- From now on, the thread 2 will be controlled by the debugger:
  - you can use the up, down, step, next, set commands
- Issue the **step** command until the thread gets out of the sleep
- When the PC is on the while (n<10) line, change the value of n:
  - **set var n=20**
- Issue the command **continue** to resume execution.
  - Now a thread will terminate
- do the same procedures until all threads are terminated

Simple Threads
III

Correct the program form the previous exercise

IV

Change the previous program so that all the created threads read characters from the keyboard, slep for one second and exit when the read character is **'k'**.
The main thread should execute the same code as all the other threads.
Each thread will count how many characters has read and will print this value before terminating.

What happens if the main thread reads a 'k' before all others?

Arguments into threads
V

Modify exercise IV of the first laboratory (**lab4-V-serial.c**) so that 4 threads are created and each thread is responsible for finding and printing the multiples of either 2, 3, 5 or 7:

- The first thread will print the multiples of 2
- The second thread will print the multiples of 3
- The third thread will print the multiples of 5
- The forth thread will print the multiples of 7

The main thread creates an array with the random number and the threads read those values.
The main thread will wait for the other threads using a getchar.

Waiting for threads to dye
VI

Change the exercise V so that the main thread does not read from the keyboard, but waits for the termination of all other thread, using the **pthread_join** system call.

Getting thread return values
VII

Change the exercise VI so that each thread, when terminating will returns the corresponding count to the main thread.
The main thread will receive this value and print it in the screen.
Experiment the following ways to get a value out of the thread:

- pthread_exit(count)
- pthread_exit(&count)
- pthread_exit(ptr)

How to compile programs using threads.

- **#include <pthread.h>**
- Compile and link with **-lpthread**

System calls

- pthread_create
- pthread_join
- pthread_exit

Documentation

- Compiling for debugging
  - ftp://ftp.gnu.org/pub/old-gnu/Manuals/gdb/html_node/gdb_16.html
- Your program's arguments
  - ftp://ftp.gnu.org/pub/old-gnu/Manuals/gdb/html_node/gdb_18.html
- Debugging programs with multiple threads
  - ftp://ftp.gnu.org/pub/old-gnu/Manuals/gdb/html_node/gdb_24.html
- Assignment to variables
  - ftp://ftp.gnu.org/pub/old-gnu/Manuals/gdb/html_node/gdb_111.html
- Chapter 2 Basic Threads Programming
  - docs.oracle.com/cd/E19120-01/open.solaris/816-5137/tlib-12926/index.html
- Chapter 8 Compiling and Debugging
  - docs.oracle.com/cd/E19120-01/open.solaris/816-5137/compile-74765/index.html
- man pages
  - man 2 alarm
  - man 2 signal
  - man 2 sigaction
  - man 2 pthread_create
  - man 2 pthread_join
  - man 2 pthread_exit