1 SYSTEMS PROGRAMMING LABORATORY VI - Threads II & IPC

Examples&Exercises:

- Complete the following codes if necessary, then compile and run the code.
- Analyze the code and output.

1. Semaphores for Threads; job-queue3.c
   - Complete the code.
   - The queue is controlled with a semaphore. Do we have any synchronization problem any more?
2. Shared Memory; shm.c
   - Study and analyze the code.
   - During/after execution, give the following commands

     ```
     $ ipcs -m
     $ ipcrm shm SHMID
     ```

     for the cases
     - do that after execution;
     - put a **sleep** inside the code;
     - do not deallocate shared memory segment.
3. Mapped Memory; mmap-write.c, mmap-read.c
   - Study and analyze the code.
   - Apply the following procedure

     ```
     $ ./mmap-write /yourdirectory/integer-file
     $ cat /yourdirectory/integer-file
     42
     $ ./mmap-read /yourdirectory/integer-file
     value: 42
     $ cat /yourdirectory/integer-file
     84
     ```

4. Pipes;
    - Communication between parent and child processes; pipe.c,
    - Redirecting the standard input, output, and error streams; dup2.c.
    - Study and analyze the codes and outputs.
5. FIFOs;
    - Creating a named pipe,

      ```
      $ mkfifo /yourdirectory/fifo
      $ ls -l /yourdirectory/fifo
      ```

    - In one window, read from the FIFO by invoking the following:

      ```
      $ cat < /yourdirectory/fifo
      ```

    - In a second window, write to the FIFO by invoking this:

      ```
      $ cat > /yourdirectory/fifo
      ```

    - Then type in some lines of text. Each time you press *Enter*, the line of text is sent through the FIFO and appears in the first window. Close the FIFO by pressing Ctrl+D in the second window.
    - Remove the FIFO with this line:

      ```
      $ rm /yourdirectory/fifo
      ```