# POSIX Thread (pthread)

1. Download the source code from the assignment, createpthread.c, and answer the following questions (refer to <https://randu.org/tutorials/threads/>):
   1. What is the purpose of pthread\_create()? What is the meaning of parameters?
   2. What is the return value of pthread\_create() if it is successful?
   3. How many threads were created in the source file createpthread.c?
   4. Compile the createpthread.c. Remember use pthread lib when you compile the file (i.e. -lpthread).
   5. Let child threads sleep 60 seconds by adding sleep() to child thread’s function. Compile it and run it again. Open another terminal within 60 seconds and identify how many threads are running using ps -T. Observe the PID and SPID change. Please show the screenshot. Hint: You may use the parameter -u yourusername or -aT to collect your threads
   6. What is the purpose of pthread\_join()?
2. Read the source code, pthreadcounter0.c, and answer the following questions:
   1. Modify 0xFF into 0xA in pthreadcounter0.c. Compile the pthreadcounter0.c. Make sure that you link the appropriate library. Run the program and paste the screenshot to show what you got.
   2. In the lab, ‘Signal: Software Interrupt’, we have learned to send a software interrupt to processes. Please copy pthreadcounter0.c to pthreadcounter1.c and refer to the source code, ipc\_signal.c. In pthreadcounter1.c, you need to (1) register a signal handler to each created thread (children thread), (2) the signal handler function, named hello(), prints out “hello world! Bye Bye world! \n” and then exit the thread using pthread\_exit(NULL), (3) the main thread needs to sleep 5 seconds and then send out the signal SIGUSR1 to each thread using pthread\_kill (refer to the link above). Show your code and paste the screenshot.