



# COSC 3360 - 24967 - Fundamentals of Operating Systems


[Dashboard](#) / [My courses](#) / [COSC3360SP2023-01](#) / [EXAM 2](#) / [Programming Question 1](#)

 Description

 [Submission view](#)

 **Available from:** Thursday, 30 March 2023, 2:30 PM

 **Due date:** Thursday, 30 March 2023, 3:50 PM

 **Requested files:** main.cpp ( [Download](#))

**Type of work:**  Individual work

**This question closes at 3:50 PM. You must save your work before this time, as Moodle will not save the work for you.**

Write a C++ program that creates multiple POSIX threads to print the value of a positive integer value (from STDIN) in reverse order. The program executes the following tasks:

1. Receives as input a string representing a positive integer value number.
2. Creates "nthreads" child threads (where "nthreads" is the number of digits of the positive integer value from the input).
3. Assigns a thread number to each child thread based on the order they are created. The value for the thread number must start from zero and be increased by one before assigning it to the next child thread. You must guarantee that the first child thread has a thread number equal to zero, the second child thread has a thread number equal to one, and the nth child thread has a thread number equal to n - 1.
4. Assigns a digit from the input value to each child thread. The child threads receive the digit from the input based on their thread number (from left to right); Given the input value "4723", the first child thread receives the digit "4", the second child thread receives the digit "7", the third child thread receives the digit "2", and the fourth child thread receives the digit "3".
5. Each child thread prints the received digit from the main thread. The print statement executed by the child thread cannot be inside a critical section.
6. Your program must guarantee that the digits are printed from the least significant to the most significant (in reverse order).

For a positive integer value = 4723, the expected output is:

3274

## NOTES

1. **You must create n child threads (where n is the number of digits from the input value).**
2. **You can only use POSIX semaphores, pthreads mutex semaphores, or pthreads condition variables to achieve synchronization (A penalty of 100% will be applied to solutions using mechanisms other than the synchronization mechanisms listed before).**
3. **A penalty of 100% will be applied to solutions that do not assign the "threadNumber" based on the specifications of the question.**
4. **Programs not compiling will receive a penalty of 100%.**
5. **To simplify your solution, the template file defines several global variables that you are allowed to use in your program for this programming question. However, remember that you should avoid using global variables when writing programs.**

## Requested files

main.cpp

```

1  #include <pthread.h>
2  #include <iostream>
3  #include <string.h>
4  #include <stdlib.h>
5  #include <fcntl.h>
6  #include <cmath>
7
8  static pthread_mutex_t bsem;    // Mutex semaphore
9  static pthread_cond_t waitTurn = PTHREAD_COND_INITIALIZER; // Condition variable to control the turn
10 static int turn; // Index to control access to the turn array
11
12 struct infoFromMain
13 {
14     int digit;
15     int threadNumber;
16 };
17
18 void *printInReverse(void *void_ptr_argv)
19 {
20     // std::cout << /*variable name*/;
21     return nullptr;
22 }
23
24 int main()
25 {
26
27     std::string number;
28     std::cin >> number;    // input from STDIN (you can safely assume that the input is always valid)
29
30
31     int nThreads = number.size();
32
33     pthread_mutex_init(&bsem, NULL); // Initialize bsem to 1
34
35     pthread_t *tid = new pthread_t[nThreads];
36     infoFromMain *arg = new infoFromMain[nThreads];
37
38     turn = // initialize the turn here
39
40     for(int i=0;i<nThreads;i++)
41     {
42
43         arg[i].threadNumber = // initialize the thread number here (remember to follow the rules from the specifications of the assignment)
44         arg[i].digit = // initialize the digit here (remember to follow the rules from the specifications of the assignment)
45         if(pthread_create(&tid[i],nullptr,printInReverse,&arg[i]))
46         {
47             std::cerr << "Error creating thread" << std::endl;
48             return 1;
49         }
50     }
51     for(int i=0;i<nThreads;i++)
52         pthread_join(tid[i],nullptr);
53
54     delete [] arg;
55     delete [] tid;
56     return 0;
57 }
58
59

```

[VPL](#)

You are logged in as Dosbol Aliev (Log out)

COSC3360SP2023-01

[Data retention summary](#)[Get the mobile app](#)