

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI,
K. K. BIRLA GOA CAMPUS
I SEMESTER 2010-2011**

Operating Systems (CS C372 & IS C362)

Assignment 4

Due date 04/10/2010 (8.00 A.M)

Marks 15

Instructions:

- (1) Please upload the assignment using <http://cc.bits-go.a.ac.in/file>
(The file name should be <your id number>.tar.gz Example: **2006A7PS123G.tar.gz**)
You are allowed to upload only once.
- (2) This is an **individual** assignment. Please see section 4b of handout for Malpractice regulations.
- (3) The programming assignments will be graded according to the following criteria
 - Completeness; does your program implement the whole assignment?
 - Correctness; does your program provide the right output?
 - Efficiency; have you chosen appropriate algorithms and data structures for the problem?
 - Programming style (including documentation and program organization); is the program well designed and easy to understand?
 - Viva conducted by me.

DO NOT FORGET to include a README file (text only) in your tar.gz file with the following contents.

General README instructions

In the directory you turn in (please upload the assignment as a tar.gz file), you must have a text-only file called README, in which you will cover AT LEAST the following:

1. Your name. If you interacted significantly with others indicate this as well.
2. A list of all files in the directory and a short description of each.
3. HOW TO COMPILE your program.
4. HOW TO USE (execute) your program.
5. A description of the structure of your program.
6. In case you have not completed the assignment, you should mention in significant detail:
 - o What you have and have not done
 - o Why you did not manage to complete your assignment (greatest difficulties)This will allow us to give you partial credit for the things you have completed.
7. Document any bugs of your program that you know of. Run-time errors will cost you fewer points if you document them and you show that you know their cause. Also describe what you would have done to correct them, if you had more time to work on your project.

NB: We will remove the link exactly by 8 A.M. Those who are taking the lifeline can mail the code (only one lifeline is allowed for the entire set of assignments) to biju@bits-go.a.ac.in within 24 hours of the deadline.

Please refer section 4b of the handout to know more about Malpractice regulations of the course.

Question 1: Merging 2 arrays using threads

This program is to test your understanding in pthread functions

Suppose we have two sorted arrays $A[]$ and $B[]$ of N and M elements respectively. For simplicity, assume that all of these $N + M$ elements are distinct. Merge these two arrays into a sorted one by using threads.

Take an element from array A , say $A[i]$, which is larger than $i-1$ elements of array A . To know the exact location of $A[i]$ in merged array, one should find how many elements of array B is smaller than $A[i]$. With a slightly modified binary search, one can easily determine the location of $A[i]$ in merged array. There are only three possibilities:

1. $A[i]$ is less than $B[0]$: In this case, $A[i]$ is larger than i elements in A and smaller than all elements in B . Therefore, $A[i]$ should be in position i of the merged array.
2. $A[i]$ is larger than $B[m-1]$: In this case, $A[i]$ is larger than i elements in A and m elements in B . Therefore, $A[i]$ should be in position $i+m$ of the merged array.
3. $A[i]$ is between $B[k-1]$ and $B[k]$. In this case, $A[i]$ is larger than i elements in A and k elements in B . Therefore, $A[i]$ should be in position $i+k$ of the sorted array.

The main thread creates a new thread to read N and M . Based on N and M values, the new thread allocates space for array A , array B and array **Result**. Then the new thread takes elements of array A and elements of array B from the user (You are allowed to use global variables in this program). After the successful execution of new thread, it joins the main thread. Then the main thread creates $N + M$ concurrently executing threads, each of which handles an element in A or in B . Each of these threads determines its position in the merged array and writes the values into the corresponding location. After the completion of all the thread executions, the array **Result** will have a merged result. Thus, we use $2N$ (if $N = M$) threads, each of which takes $O(\log_2(N))$ comparisons to get the job done. The main thread should create a new thread to print the merged sorted resultant array **Result**.

Question 2: Multiplication of two N digit Numbers

This program is to test your understanding in pthread functions

Write a program in C to demonstrate multiplication of two N digit numbers. The computation for each digit in the result must be performed by one thread. The final generation of the result (along with the carry forwarding) will be taken care by the main thread.

Example:

```

      1   2   3   X
      4   3   1
=====
      1 | 2 | 3
    3 | 6 | 9
  4 | 8 | 12
=====
    4 | 11 | 19 | 11 | 3      (returned by threads t1 to t5)
=====
    5 | 3 | 0 | 1 | 3      (computed by main)
```