

IS F462 Network Programming
I semester 2015-16
Lab6 Exercise

Write a program *merge.c* for the following requirements.

In merge sort, in every step the input is divided into two halves and the same process is applied to the two halves. This is done until the data becomes trivial to sort. We will use multiple processes to run the merge sort. Every time the data is split into two halves, each half is taken care by one process. This way there is parallel processing to some degree. The merge step happens in the process where the data is partitioned into two halves as shown in the pseudo code below. Finally the process which has started the execution of merge sort should print the sorted list. **Use shared memory and necessary semaphores wherever required.**

Algorithm mergeSort (S, C)

Input sequence S with n elements, comparator C

Output sequence S sorted according to C

```
if S.size() > 1
    (S1, S2) <-- partition(S, n/2)
    mergeSort(S1, C) //should happen in child1
    mergeSort(S2, C) //should happen in child2
    S <-- merge(S1, S2)
```

Write a program which reads input (N, and list of N integers) from a file *input.txt*. It implements the above algorithm as described above. During computation, it should print the pid of the process which is calling merge() and output of the merge() function.

Files Expected: A tar file **<idno>_lab6.tar** containing merge.c and makefile to compile your program.

Please upload on <http://nalanda> latest by 8th Nov.