

CS342 Operating Systems - Spring 2015

Project 1: Cooperating Processes and IPC

Assigned: Feb 17, 2015, Tuesday

Due date: Feb 27, 2015, Friday, 11:55pm

Write a C program in Linux OS that will use multiple processes to sort a sequence of integers taken from an input file. The program will be called `msort` and it will take 3 parameters: an input filename, an output filename, and the number of child processes (N) to create and to use for sorting. The program (main process) will fork N children (child processes 0, 1, ..., $N-1$) to help in sorting. The main process will also create pipes to communicate with the children (see the man page of `pipe()`). You can create two pipes per child (one for each direction of communication between parent and child). The main process will read the input file one integer at a time and will feed the received integers to the child processes through the pipes with the following rule: an integer x will be given to a child process $x \% N$ ($x \bmod N$). For example, if there are 10 children, and integer 97 will be given to child 7. Each child will receive (read) its sequence of integers from its pipe, will sort them (using a linked list), and will pass them back to the parent through another pipe. The parent will receive the sorted sequence of integers from its children through the respective pipes, merge the sorted sequences while receiving them, and print out the sorted (merged) integers to the output file. At the end the output file will include the integers in the input file in sorted order. Then the program (main process and all children) will terminate. An example invocation of your program can be as follows:

```
msort filename.txt outfilename.txt 5
```

This will create 5 child processes to help in sorting the integers in the specified input file to the output file.

A filename can be at most 127 characters long. The maximum value of N can be 20. The number of integers in the input file can be anything. Use linked lists to store them in the child processes.

Do also some experiments. For example, time the execution of your program for several cases. For example, you can time the execution of your program for different values of N and for different input sizes (number of integers in input file). Put the results of your experiments into tables or charts which are to be put into a report. That means write a report about your experiments and results. Convert the report file to PDF before submission.

You will put your program C file in a directory. In that directory you will also include a Makefile to compile your program, and a report.pdf file. Also include a README file with your id in it. Pack (using tar command) and zip (using gzip command) your project folder containing the files mentioned and upload it as a single project1.tar.gz file in Moodle. Both your report and program (source code, execution, results) will be graded.