Faculty of Computers and Information
Cairo University
Spring-2019

Parallel Processing- 2019

Project Description

Deadline & Submission:

1. **The project is group of 3 Maximum.**
2. **At least one team member should submit the compressed group solution as zip file containing the program under Acadox –> tasks (name your assignment file "Project_ID1_ID2_ID3_G#_G#.zip"). e.g. Project _20168383_201638838_G1_G1.zip**
3. **The deadline for submitting the solution of the project is 25 April 11:59 PM.**
4. **Acadox link: http://www.acadox.com/class/56801**
5. **Code must be in C and MPI & you must run it before sending.**
6. **Cheating means zero for all teams who collaborated in the cheating process with no further discussions with the collaborating teams.**

Project Title:

Managing the election process to choose the new president

Project Short Description:

You know that the results of the elections may take weeks to be announced. But we want quickly to announce which candidate will win. So given the preferences lists, you need to write a parallel program to announce which candidate will win and in which round.

Project Statement:

Finally, it is time to vote for a new president and you are really excited about that. You know that the final results may take weeks to be announced, while you can't really wait to see the results.

Somehow you managed to get the preferences list for every voter. Each voter sorted out all candidates starting by his most preferred candidate and ending with his least preferred one. When voting, a voter votes for the candidate who comes first in his preferences list. For example, if there are 5 candidates (numbered 1 to 5), and the preferences list for one voter is [3, 2, 5, 1, 4] then voter will give the highest vote for candidate 3 and the lowest vote for candidate 4.

Here are the rules for the election process:

1. There are C candidates (numbered from 1 to C), and V voters.
2. The election process consists of up to 2 rounds. All candidates compete in the first round. If a candidate receives more than 50% of the votes, he wins, otherwise another round takes place, in which only the top 2 candidates compete for the presidency, the candidate who receives more votes than his opponent wins and becomes the new president.
3. The voters' preferences are the same in both rounds so if the preference list [1 2 3 4 5] in the first round and the second round become between candidate 1 and 2 so the preferences is the same [1 2].

Given the preferences lists, you need to write a program to announce which candidate will win and in which round.

For example: If the input

3 5 // number of candidates & number of voters
1 2 3 // voter 1 preference list
1 2 3 // voter 2 preference list
2 1 3 // voter 3 preference list
2 3 1 // voter 4 preference list
3 2 1 // voter 5 preference list

Then the output will be 2 2 // candidate 2 wins in round 2

Explanation: You should print the output something like this:

Candidate [1] got 2/5 which is 40%

Candidate [2] got 2/5 which is 40%

Candidate [3] got 1/5 which is 20%

So second round will take place between candidates 1 and 2 with same preferences

1 2 // voter 1 preference list
1 2 // voter 2 preference list
2 1 // voter 3 preference list
2 1 // voter 4 preference list
2 1 // voter 5 preference list

Candidate [1] got 2/5 which is 40%

Candidate [2] got 3/5 which is 60% so candidate 2 wins in round 2

And if input

2 3 // number of candidates & number of voters
2 1 // voter 1 preference list
1 2 // voter 2 preference list
2 1 // voter 3 preference list

Then the output will be 2 1 // candidate 2 wins in round 1

The code should generate **a data file** that contains the voters' preferences. The format of the file must be as follows: number of candidates in the first line, number of voters in the second line, and voters' preferences equal to number of voters. The format of the file is shown in the following figure.

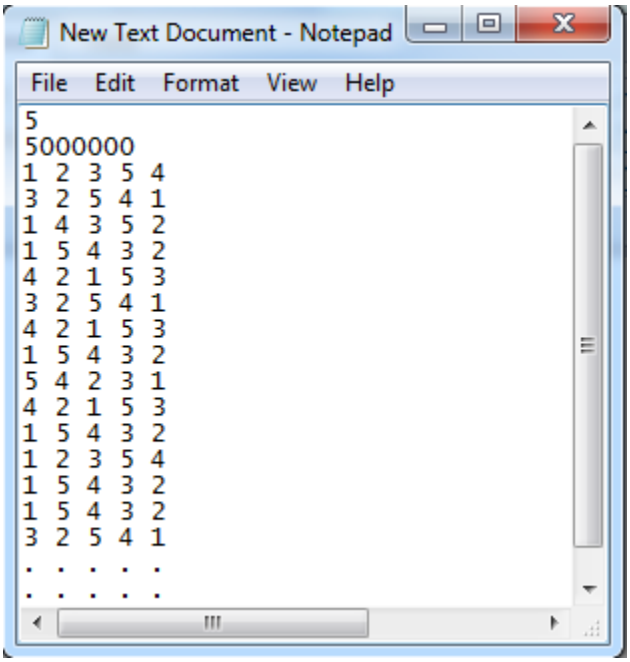

Figure 1

Run Steps that should be followed:

When the program run you should prompt the user to:

1- Generate a data file like the one showed in **Figure 1** to generate the voter's preferences
2- Calculate the result and print the winner and in which round & if the user chooses this option take the filename of file generate from step 1 as input.

Note: when running the program the file must not be loaded by one process but every running process should loads its part from the file by seeking into it, and the number of voters may not be divisible by number of processes so you should handle it.

The Expected Output:

Print to the console the steps happening in every process and print which candidate will win the elections and in which round. And if there are 2 rounds print this information and show the percentage of every candidate per each round.

Deliverables:

Submit the **source code** and a PDF document that contains your team members' names, ids and group number(s). The PDF document should describe the implemented parallel scenario and the steps followed in the source code.

Marking Criteria:

The project is out of 80 grades
Handling the file read = 10 grades
Handling the file generation = 10 grades
Submitting the PDF = 5 grades
Handling first round = 20 grades
Handling second round = 30 grades
Print the output = 5 grades