

PROJECT 01

RING LEADER ELECTION WITH MPI (Message Passing Interface)

DAVID NGUYEN

D_NGUYEN@CSU.FULLERTON.EDU

CALIFORNIA STATE UNIVERSITY, FULLERTON

CPSC 479

HIGH PERFORMANCE COMPUTING

DOINA BEIN

Table of Contents

Abstract	1
Pseudocode	2
Main	2
randomInt(int rank)	3
int concat(int gen, int rank)	3
How to run program	4
Screenshots	5
Credit Proof.....	5
Example Outputs.....	6

Abstract

Project 1 of California State University, Fullerton's, CPSC-479 High Performance Computing, named *Introduction to HPC – Variant of Leader Election on a Ring Topology*, nicknamed by David Nguyen, *Parallel Ring Leader Election*.

Project goal is to implement an algorithm to select a President and a Vice-President on a ring topology based on processes' exclusive randomly generated computed value, smallest odd and smallest even respectively. Using MPI (Message Passing Interface), results and comparisons can be checked and passed among processes.

This project showcases one of many tasks that can be achieved using Message Passing Interface to speed up minor tasks and remove overheads that would normally appear in sequential processing.

Pseudocode

Main

```
int results[4], rank, size, temp;  
// results integer array of size 4 to hold results of election  
// rank and size for MPI values  
// temp is to hold each process's randomly generated value
```

<Initiate MPI>

```
if size is less than 6 or greater than 20:  
    exit program  
else if rank is not 0  
    MPI Receive results array from previous process  
    Generate random integer and store into temp  
    if temp is even:  
        if temp is less than even in results:  
            replace even in results with temp  
            replace process number in results  
    else if temp is less than odd in results:  
        replace odd in results with temp  
        replace process number in results  
else  
    Generate random integer and store into temp  
    If temp is even:  
        set results array to {19999, 0, temp, 0}  
    else:  
        set results array to {temp, 0, 19990, 0}
```

MPI Send results to next process, else if last process, send back to process 0

```
if rank is 0:  
    MPI receive finalized results from final process  
    Print results
```

<End MPI block with MPI_Finalize()>

randomInt(int rank)

```
int randTmp = generate random integer between 0 - 89, then add  
10
```

```
if randTmp is negative:
```

```
    multiply randTmp by -1 to make positive
```

```
return concat(randTmp, rank);
```

int concat(int gen, int rank)

```
char s1[5] = "1" // size of 5 because largest concatenation is  
1XXYY
```

```
char s2[2]; // size of 2 because largest input is double digit
```

```
char s3[2];
```

```
convert gen into string and store into s2
```

```
convert rank into string and store into s3
```

```
concatenate s1, which contains "1", with s2
```

```
concatenate s1, which now contains "1" + contents of s2, with s3
```

```
return s1 converted back to integer
```

How to run program

To run program:

1. In terminal, change to directory with file **ring_election.c**.
2. Run ``mpicc ring_election.c`` to build the program.
3. Run ``mpirun -n <# of processes wished> ./a.out`` to run program
 - a. # of processes must be between [6, 20]
 - b. a.out is the default compiled name. If you named the compiled name differently, replace a.out with that name.
4. Success!

Refer to follow pages for screenshots of example outputs.

Screenshots

Credit Proof

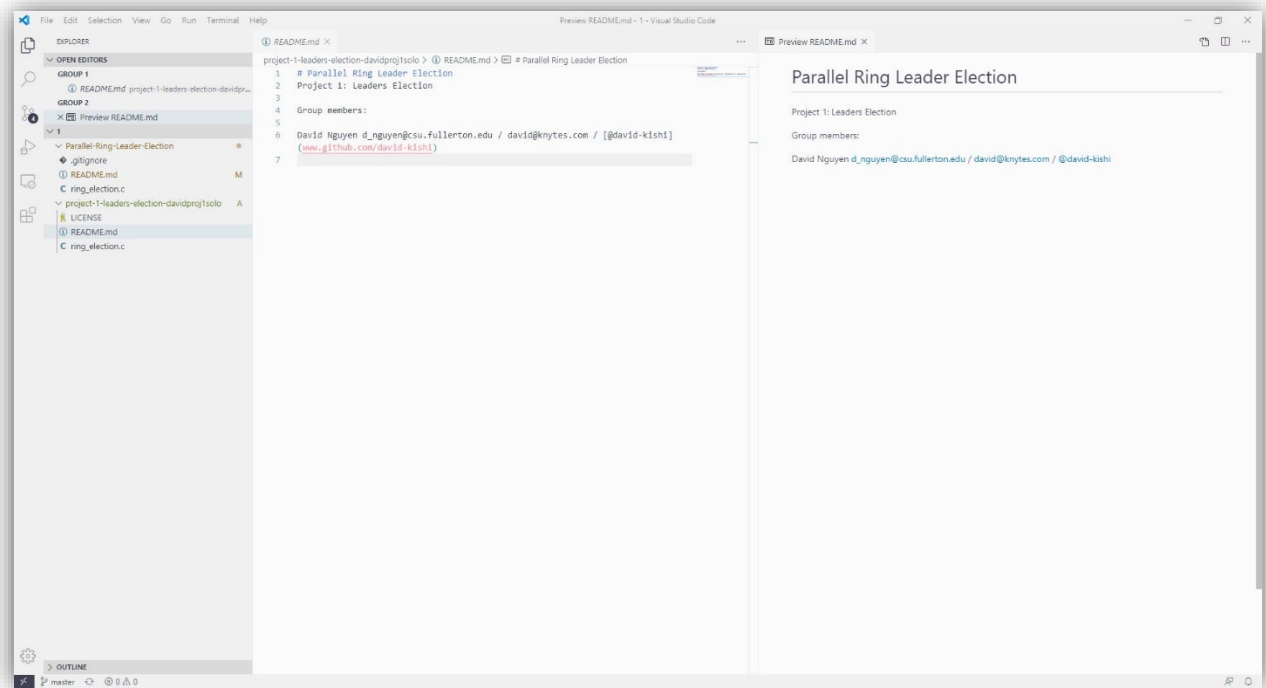
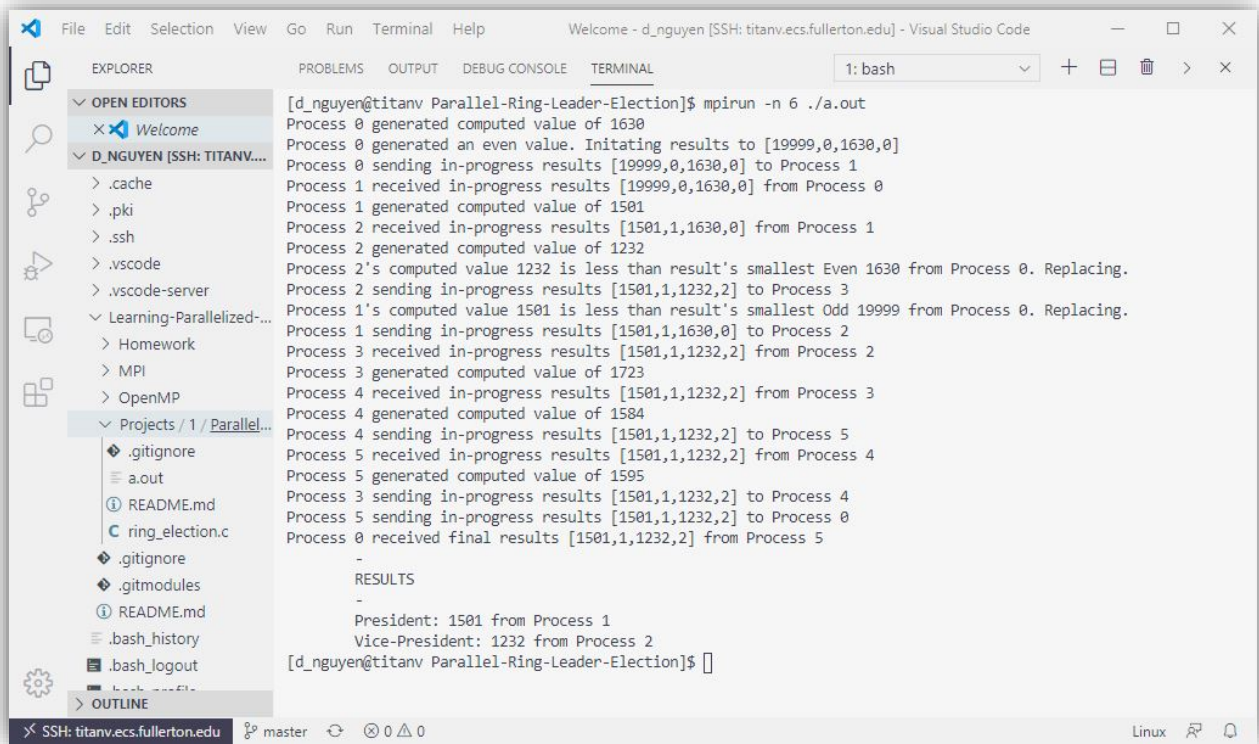


Figure 1. Credits

Example Outputs



The screenshot shows a Visual Studio Code window with a terminal open. The terminal is running a program called 'Parallel-Ring-Leader-Election' with 6 processes. The output shows the following sequence of events:

```
[d_nguyen@titanv Parallel-Ring-Leader-Election]$ mpirun -n 6 ./a.out
Process 0 generated computed value of 1630
Process 0 generated an even value. Initiating results to [19999,0,1630,0]
Process 0 sending in-progress results [19999,0,1630,0] to Process 1
Process 1 received in-progress results [19999,0,1630,0] from Process 0
Process 1 generated computed value of 1501
Process 2 received in-progress results [1501,1,1630,0] from Process 1
Process 2 generated computed value of 1232
Process 2's computed value 1232 is less than result's smallest Even 1630 from Process 0. Replacing.
Process 2 sending in-progress results [1501,1,1232,2] to Process 3
Process 1's computed value 1501 is less than result's smallest Odd 19999 from Process 0. Replacing.
Process 1 sending in-progress results [1501,1,1630,0] to Process 2
Process 3 received in-progress results [1501,1,1232,2] from Process 2
Process 3 generated computed value of 1723
Process 4 received in-progress results [1501,1,1232,2] from Process 3
Process 4 generated computed value of 1584
Process 4 sending in-progress results [1501,1,1232,2] to Process 5
Process 5 received in-progress results [1501,1,1232,2] from Process 4
Process 5 generated computed value of 1595
Process 3 sending in-progress results [1501,1,1232,2] to Process 4
Process 5 sending in-progress results [1501,1,1232,2] to Process 0
Process 0 received final results [1501,1,1232,2] from Process 5
-
RESULTS
-
President: 1501 from Process 1
Vice-President: 1232 from Process 2
[d_nguyen@titanv Parallel-Ring-Leader-Election]$
```

Figure 2. Example output with 6 processes


```

Welcome - d_nguyen [SSH: titanv.ecs.fullerton.edu] - Visual Studio Code
1: bash

Process 1's computed value 1731 is less than result's smallest Odd 19999 from Process 0. Replacing.
Process 1 sending in-progress results [1731,1,1730,0] to Process 2
Process 2 received in-progress results [1731,1,1730,0] from Process 1
Process 2 generated computed value of 1632
Process 2's computed value 1632 is less than result's smallest Even 1730 from Process 0. Replacing.
Process 2 sending in-progress results [1731,1,1632,2] to Process 3
Process 3 received in-progress results [1731,1,1632,2] from Process 2
Process 3 generated computed value of 1853
Process 3 sending in-progress results [1731,1,1632,2] to Process 4
Process 4 received in-progress results [1731,1,1632,2] from Process 3
Process 4 generated computed value of 1684
Process 4 sending in-progress results [1731,1,1632,2] to Process 5
Process 5 received in-progress results [1731,1,1632,2] from Process 4
Process 5 generated computed value of 1705
Process 5's computed value 1705 is less than result's smallest Odd 1731 from Process 1. Replacing.
Process 5 sending in-progress results [1705,5,1632,2] to Process 6
Process 6 received in-progress results [1705,5,1632,2] from Process 5
Process 6 generated computed value of 1586
Process 6's computed value 1586 is less than result's smallest Even 1632 from Process 2. Replacing.
Process 6 sending in-progress results [1705,5,1586,6] to Process 7
Process 7 received in-progress results [1705,5,1586,6] from Process 6
Process 7 generated computed value of 1387
Process 7's computed value 1387 is less than result's smallest Odd 1705 from Process 5. Replacing.
Process 7 sending in-progress results [1387,7,1586,6] to Process 8
Process 8 received in-progress results [1387,7,1586,6] from Process 7
Process 8 generated computed value of 1748
Process 8 sending in-progress results [1387,7,1586,6] to Process 9
Process 9 received in-progress results [1387,7,1586,6] from Process 8
Process 9 generated computed value of 1569
Process 9 sending in-progress results [1387,7,1586,6] to Process 10
Process 10 received in-progress results [1387,7,1586,6] from Process 9
Process 10 generated computed value of 12810
Process 10 sending in-progress results [1387,7,1586,6] to Process 11
Process 11 received in-progress results [1387,7,1586,6] from Process 10
Process 11 generated computed value of 17911
Process 11 sending in-progress results [1387,7,1586,6] to Process 12
Process 0 received final results [1387,7,1586,6] from Process 15
-
RESULTS
-
President: 1387 from Process 7
Vice-President: 1586 from Process 6
Process 12 received in-progress results [1387,7,1586,6] from Process 11
Process 12 generated computed value of 14912
Process 12 sending in-progress results [1387,7,1586,6] to Process 13
Process 13 received in-progress results [1387,7,1586,6] from Process 12
Process 13 generated computed value of 11613
Process 13 sending in-progress results [1387,7,1586,6] to Process 14
Process 14 received in-progress results [1387,7,1586,6] from Process 13
Process 14 generated computed value of 13514
Process 14 sending in-progress results [1387,7,1586,6] to Process 15
Process 15 received in-progress results [1387,7,1586,6] from Process 14
Process 15 generated computed value of 12115
Process 15 sending in-progress results [1387,7,1586,6] to Process 0
[d_nguyen@titanv Parallel-Ring-Leader-Election]$

```

Figure 3. Example output with 16 processes