

California State University Fullerton

Department of Computer Science

Conner Cook

connerccook@csu.fullerton.edu

Spencer DeMera

spencer.demera@csu.fullerton.edu

Diane Le

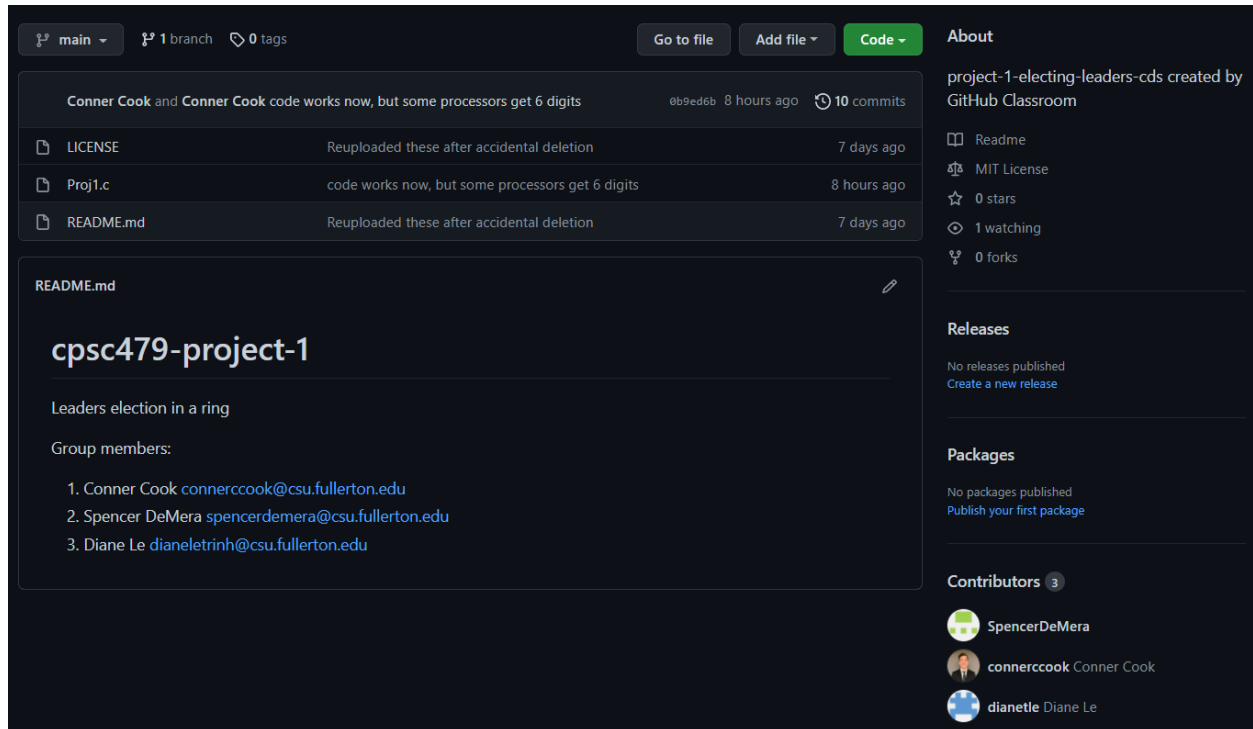
dianeletrinh@csu.fullerton.edu

CPSC 479 : Introduction to High Performance Computing

Project One Report

18 March 2022

README Screenshot:



Pseudo Code:

main():

```

    Int Arr[4] = {-1,-1,-1,-1}
    Int id
    MPI_Init();
    MPI_Comm_rank()
    MPI_Comm_size()

    srand(time(NULL) + rank)

    If rank not equal to 0
        Id = createID(rank)
        MPI_Recv(&arr, 4, MPI_INT, rank-1, 0, MPI_COMM_WORLD,
MPI_STATUS_IGNORE);

        If id is even
            Arr[0] = id
  
```

```

        Arr[1] = rank
    Else
        Arr[2] = id
        Arr[3] = rank
Else
    Id = createID(rank)
    If id is even
        Arr[0] = id
        Arr[1] = rank
        Arr[2] = 1
        Arr[3] = -1
    Else
        Arr[0] = 99998
        Arr[1] = -1
        Arr[2] = id
        Arr[3] = rank
    MPI_Send(&arr, 4, MPI_INT, (rank + 1) % size, 0, MPI_COMM_WORLD)

    If rank == 0
        Id = createID(rank)
        MPI_Recv(&arr, 4, MPI_INT, size-1, 0, MPI_COMM_WORLD,
MPI_STATUS_IGNORE)
        If id is even
            Arr[0] = id
            Arr[1] = rank
        Else
            Arr[2] = id
            Arr[3] = rank
        print("president is processor arr[1] with id arr[0]")
        print("vice president is processor arr[3] with id arr[2]")

```

Code Compilation & Execution Description

Code is written and compiled in bash / Linux:

- Compile Command : mpicc Proj1.c -o Proj1.out
- Run Command : mpirun -oversubscribe -np N ./Proj1.out
- N represents the intended number of processors to be used
- Use of -oversubscribe is necessary for machines with less than N number of processor cores

== Warning ==

- When printing IDs of each processor from the createID() function, an extra ID for processor N + 1 is printed and is ambiguous / can be ignored.
- Not sure why this is occurring

Output on N Processors (in VS Code Terminal)

Output Running on N Processors when N = 15:

```
Rank 1 has ID : [84010]
Rank 4 has ID : [19041]
Rank 12 has ID : [97121]
Rank 2 has ID : [59021]
Rank 3 has ID : [34030]
Rank 6 has ID : [92060]
Rank 7 has ID : [57071]
Rank 9 has ID : [79091]
Rank 10 has ID : [10100]
Rank 11 has ID : [71111]
Rank 5 has ID : [10050]
Rank 13 has ID : [34130]
Rank 0 has ID : [67001]
Rank 0 has ID : [58000]
-> Rank 1 has [99998].[67001] from Rank 0
-> Rank 2 has [84010].[67001] from Rank 1
-> Rank 3 has [84010].[67001] from Rank 2
-> Rank 4 has [34030].[67001] from Rank 3
Rank 14 has ID : [88140]
Rank 8 has ID : [98080]
-> Rank 5 has [34030].[67001] from Rank 4
-> Rank 6 has [10050].[67001] from Rank 5
-> Rank 7 has [10050].[67001] from Rank 6
-> Rank 8 has [10050].[67001] from Rank 7
-> Rank 9 has [10050].[67001] from Rank 8
-> Rank 13 has [10050].[97121] from Rank 12
-> Rank 12 has [10050].[79091] from Rank 11
-> Rank 11 has [10050].[79091] from Rank 10
-> Rank 10 has [10050].[79091] from Rank 9
-> Rank 14 has [10050].[97121] from Rank 13
-> Rank 0 has [10050].[97121] from Rank 14

The President is processor 5 with the ID [10050]
The Vice President is processor 12 with the ID [97121]
```

Output Running on N Processors when N = 10:

```
Rank 1 has ID : [52010]
Rank 2 has ID : [75021]
Rank 6 has ID : [88060]
Rank 7 has ID : [62070]
Rank 8 has ID : [65081]
Rank 4 has ID : [88040]
Rank 3 has ID : [32030]
Rank 9 has ID : [86090]
Rank 5 has ID : [77051]
Rank 0 has ID : [21001]
Rank 0 has ID : [77001]
-> Rank 1 has [99998].[21001] from Rank 0
-> Rank 2 has [52010].[21001] from Rank 1
-> Rank 3 has [52010].[75021] from Rank 2
-> Rank 5 has [32030].[75021] from Rank 4
-> Rank 4 has [32030].[75021] from Rank 3
-> Rank 8 has [32030].[77051] from Rank 7
-> Rank 6 has [32030].[77051] from Rank 5
-> Rank 7 has [32030].[77051] from Rank 6
-> Rank 0 has [32030].[77051] from Rank 9
-> Rank 9 has [32030].[77051] from Rank 8
```

```
The President is processor 3 with the ID [32030]
The Vice President is processor 5 with the ID [77051]
```