CPSC 479
Project 1
Anthony Galustyan
Louis Zuckerman

README.md

# CPSC 479 Project 1

## Electing Leaders in a Ring Topology

**Problem Summary**

Our algorithm selects two leaders, which we can call them as the president and the vice-president. The president will be the largest odd value and the vice president will be the largest even value. The two elections can run concurrently (by sending/receiving two values in the same MPI send) or separately (sending one message for president-odd value and another message for vice-president-even value).

## Group members:

> Anthony Galustyan
>
> *agalustyan@csu.fullerton.edu*
>
> Louis Zuckerman
>
> *louiszman@csu.fullerton.edu*

```
root ➜/workspaces/project-1---electing-leaders-procrastinate-or-deadlock (main X) $ mpiexec -n 10 python3 proj1.py
The even leader is 1940 and the odd leader is 1629
```

Project 1 Pseudocode: Concurrent two leader election algorithm

Assign rank of the thread executing to rank and amount of threads (or ranks) to size. Open communication path with Comm

Comm = MPI.COMM_WORLD

Rank = comm.Get_rank()

Size = comm.Get_size()

Generate a random number between 10 and 100 for each thread in the program and assign that rank that number

Num = RandomInt

Odd = 0

Even = 0

Determine whether the starting rank(0) number is even or odd then send it to the next thread rank 1 with a tag determining its status as odd or even

If rank == 0:

CPSC 479
Project 1
Anthony Galustyan
Louis Zuckerman

If num is even:

Odd = 1

Even = number

Else:

Odd = number

Even = 0

Send even to rank 1 with tag 2

Send odd to rank 1 with tag 3

Compare received numbers to current rank number and send the greater number to the next rank

If rank > 0:

Odd = received number from previous rank with tag 3

Even = received number from previous rank with tag 2

If number is even:

If number > even:

Even = number

If number > odd:

Odd = number

If rank > 0:

Send even to (rank+1)%(amount of ranks) with tag 2

Send odd to (rank+1)%(amount of ranks) with tag 3

When rank 0 receives a number then the program will output its value and whether it is even or odd and terminate

If rank == 0:

Even = received number from previous rank with tag 2

odd = received number from previous rank with tag 3

print(f”The even leader is {even} and the odd leader is {odd}”)

CPSC 479
Project 1
Anthony Galustyan
Louis Zuckerman

**How to use**

**Run**

mpiexec -n 8 python3 proj1.py

or

mpirun -n 8 python3 proj1.py

-n can be set to the desired number of processes.

**Dependencies:**

**A Dockerfile with all required dependencies has been provided in the .devcontainer folder**

A working MPI implentation (tested with MPICH)

Python3 (+python3-dev & pip)

MPI for Python (mpi4py)

mpich:

apt-get install mpich

pip:

apt-get install python3-pip

python3-dev:

apt-get install python3-dev

mpi4py:

python3 -m pip install mpi4py

Code running with n = 10

```
root ➜/workspaces/project-1---electing-leaders-procrastinate-or-deadlock (main X) $ mpiexec -n 10 python3 proj1.py
The even leader is 1940 and the odd leader is 1629
```

Code running with n = 4

```
root ➜/workspaces/project-1---electing-leaders-procrastinate-or-deadlock (main X) $ mpiexec -n 4 python3 proj1.py
The even leader is 1972 and the odd leader is 1493
root ➜/workspaces/project-1---electing-leaders-procrastinate-or-deadlock (main X) $
```