

Distributed Operating System Principles

(Fall 2020)

Project 1

Venkata Vikramaditya Varma Kunaparaju	1537-8659
---------------------------------------	-----------

Prajan Tikayyolla	6690-9943
-------------------	-----------

How to run the Project:

First open the FirstIonideProject and in the terminal run the following command

```
dotnet fsi --langversion:preview Program.fsx 100 24
```

The last two integers are the inputs for the Program N & k.

Project Material:

1. **Size of the work unit that you determined results in best performance for your implementation and an explanation on how you determined it. Size of the work unit refers to the number of sub-problems that a worker gets in a single request from the boss.**

We worked on a 4-core system so, we divided the Work into 6,8,10 Units and then compared the results. The best performance occurred when we partitioned to 8 actors.

The following are the results when we ran the program for **100000000 50** when the actors were 6,8,10 respectively.

```
Real: 00:00:29.175, CPU: 00:02:18.234, GC gen0: 10875, gen1: 3, gen2: 0
```

```
Real: 00:00:25.685, CPU: 00:02:33.078, GC gen0: 10877, gen1: 3, gen2: 0
```

```
Real: 00:00:32.053, CPU: 00:02:37.421, GC gen0: 10876, gen1: 4, gen2: 0
```

As we can see the best performance occurred with 8 actors.

2. **The result of running your program for dotnet fsi proj1.fsx 1000000 4**

```
PS C:\Users\vikra\FirstIonideProject> dotnet fsi --langversion:preview Program.fsx 1000000 4
Real: 00:00:00.000, CPU: 00:00:00.000, GC gen0: 0, gen1: 0, gen2: 0
Real: 00:00:00.520, CPU: 00:00:01.828, GC gen0: 93, gen1: 1, gen2: 0
```

3. **The running time for the above as reported by time for the above, i.e. run time scala project1.scala 1000000 4 and report the time. The ratio of CPU time to REAL TIME tells you how many cores were effectively used in the computation. If your are close to 1 you have almost no parallelism (points will be subtracted).**

As we can see from the above screenshot the ratio of CPU time to Real Time is 3.515.

4. The largest problem you managed to solve.

Largest problem we have solved is : 1000000000 50

```
Real: 00:04:56.497, CPU: 00:27:37.890, GC gen0: 109893, gen1: 32, gen2: 2
```