COP 5615 – PROJECT 1 (README FILE)

Group Members: Deepak Nair (UFID:13361231)
Abhishek Potnis (UFID:57164619)

1. The search space for mining bitcoins is a long string of random alphanumeric characters. This string is divided into smaller chunks of 5 characters prefixed with the gatorid – 'dnair'. The number of workers is determined by the number of processors in the machine (3/2 times number of processors) and they are assigned work in round robin fashion. The data is divided into smaller chunks rather than splitting it equally amongst all available actors (for proper load balancing)
   (Running time: A Loop goes from 1 to 100000 – this counter can be increased/decreased to control the duration for which the project runs.)

2. The result of running program for k-the number of leading zeros required as **4** is :

   ```
   Bitcoin : dnairCvSkH
   000095535b9747f772d38ead273cd6ccf2058d8c5e5322b7c4c157ebe1a1c94a

    Bitcoins from Client

   Bitcoin : dnairxDzrG
   0000ca92ecb3793955d85e532ef51886d431f4def364cb1d76f0f2b4a17440bd
   ```

3. We observed that multiple actors run at the same time and so multiple cores were utilized.

4. Coin with maximum number of zeros (5 leading zeros):

   ```
   Bitcoin : dnairAsv16
   00000ddb00a154d70af72561736e6a6a9fcd7a1a9fac632e9f460834492fca88
   ```

5. Largest number of working machines: We were able to run on 2 machines.

<u>Steps to execute:</u>

This project submission includes a folder **PROJECT1_UFIDs_13361231_57164619** which has subfolders in it.
In order to run the program (using SBT), open 2 command prompt terminals.

In terminal 1, reach: **PROJECT1_UFIDs_13361231_57164619\eclipseWorkspace\HelloLocal**
Type sbt run and hit enter.
In terminal 2, reach: **PROJECT1_UFIDs_13361231_57164619\eclipseWorkspace\HelloRemote**
Type sbt run and hit enter.
All bitcoins mined from both - Server and Client will be displayed in 'Local.scala', the server. (Server has the capability to mine by itself even if client is not available).

- Update IP address of application.config present in HelloRemote folder to client machine's IP and update IP address in Local.scala file to client's IP address.
- Update IP address of application.config present in HelloLocal folder to server machine's IP and update IP address of HelloRemote.scala to server's IP address.

Source files:

1. Server : **Local.Scala**
   Path:
   PROJECT1_UFIDs_13361231_57164619\eclipseWorkspace\HelloLocal\src\main\scala\local\Local.scala

2. Client : **HelloRemote.scala**
   Path:
   PROJECT1_UFIDs_13361231_57164619\eclipseWorkspace\HelloRemote\src\main\scala\remote\HelloRemote.scala