Distributed Operating Systems- Project-3
Implementation of Tapestry Algorithm

Group info:

Name: Namratha Reddy Tippireddy **UFID: 9151-9979**
Name: Megha Nagarmunoli **UFID: 6768-1778**

Aim:
The aim of this project is to implement a second-generation peer-to-peer overlay network called Tapestry Algorithm

Implementation:

In this algorithm, we have a collection of nodes and objects, each node has to make a specific number of requests and the algorithm terminates when each node makes the required number of requests. Each node creates an object hash and finds it, at its root node, the total number of hops taken to reach the root node of the object are calculated and the maximum of all is returned as the output.

For dynamic node Nid addition, we are finding the surrogate(root of the hash value for Nid) node for the newly added node. Once the surrogate is reached, it checks the longest prefix ‘p’ matched with the Nid and sends a multicast message to all nodes in level p to add Nid to their routing tables. These nodes multicast this message to add the node to the p+1 level nodes and so on till all the levels are done.
While reaching to the surrogate, the intermediate nodes in each level are copied to a new node list (neighbor list) which the new node Nid would use to generate its own routing table.

All the nodes in the network are implemented as GenServers.

Execution:

- Extract Tippireddy_Nagarmunoli.zip
- Enter Tapestry_Algorithm
- Commands:
    - mix run project3.exs <number_of_nodes> <number_of_requests>
- Example:
    - mix run project3.exs 500 5

Maximum working nodes run for Tapestry Algorithm:
(number of hex digits = 6)

```
C:\Users\Mainsoft-HR\Music\Tapestry_Algorithm>  mix run project3.exs 3000 10
“Max count value is 5”
```