

**Lab 2 Assignment (Max 40 points)** (Due: Monday by 11:59 pm, November 11, 2020)

- Goal - We will learn the basic of distance vector algorithms, using TCP socket applications.

•Project Description: **DV (Distance Vector Algorithm)**

- In this programming assignment, you are to build a simple Distance Vector program that will communicate with N partners.
  - DV program
    - You will run N instances on your machine
    - Each instance will run on a different port (instance 1 will run on port 18181, instance 2 on port 18182, instance 3 on port 18183, etc),
    - The program will start by reading in the appropriate neighbors file and vector file.
    - The program will ONLY read vectors where the fromNode is equal to that node. For instance, node 1 will only read in vectors where node1 is the fromNode.
    - The program will read in the should support the following requests from clients:
- Show files at the server's current directory: **'print'**
  - This will print the current vector table on a node.
  - ONLY print valid vectors (don't print uninitialized ones)
- Download files: **'refresh'**
  - This will force the current node to send it's vector table to all it's neighbors.
- Upload files: **'update'**
  - This will be of the form: Update fromNode toNode cost
  - Update checks that the toNode is the current node
    - If it is not it is ignored
  - Update then checks if the toNode-fromNode exists in vector table. If it does not it is added and neighbors are notified
  - If the pair is in the table, the if the new cost is less than the old cost, that cost is stored and the neighbors are notified.
- Programming environment
  - All programs should be written C or C++ and run on UNIX like platform.
  - The starter code works on Mac and should work on linux
  - All connections between a server and clients should be TCP/IP socket.
  - I have included a makefile and all the required templates

- **Required Skills**
- Everyone is expected to know following skills and knowledge in order to complete this programming assignment.
  - TCP/IP Socket programming
  - Understanding UNIX like Operating System
  - Creating/invoking processes in UNIX like environment
  - Makefile
  - C or C++

#### •Deliverables

- The deliverables for this assignment include the following files:
  - Written C or C++ Code for program
  - Makefile
  - Readme: A short description of your programs includes: the names of created executable images which will be created after run your makefile, how to run your programs.

#### •Submission

- Please do the followings when you submit your programming assignment.
  - Create a zip file that contains your written source code, makefile and readme. DO NOT INCLUDE EXECUTABLES AND COMPILED OBJECT FILES.
  - **Upload it to the class Canvas by deadline.**

#### •Grading

- The maximum point for the assignment is 40. This programming assignment will be graded by following criteria.
  - Completeness: 40 points
    - Connection :
      - 10 points – if your server can connect to all nodes
    - Function :
      - 10 : if your server shares vectors with all neighbors
      - 0: Otherwise
    - Commands
      - 3.33 points for each command
      - “download” and “upload” service is 4 points each
    - Code quality 10points
      - Comments/structure/etc.
  - EXTRA credit 10pts
    - Enhance the program so that a user can type in ROUTE <node>
    - The program will respond with the current lowest cost to <node> and what the first/next hop is