

OS LAB

Assignment 4

Hands-on experience of using shared-memory

Report

Group 30

Akash Das (20CS10006)
Prakhar Singh (20CS10045)
Rohit Kumar Prajapati (20CS30041)
Saras Umakant Pantulwar (20CS30046)

Heuristic Used to optimize the process of re-computing shortest paths

In the diagram shown below, nodes 0-7 and other nodes were present initially. Then the producer process adds node 8 to the graph and connects it to nodes 5 and 6.

Case 1: If there is an edge present between the source node 0 and newly added node:

- If the cost of the path from 0 to 6 without using node 8 is greater than the cost of the path $0 \rightarrow 8 \rightarrow 6$, then we replace previous path $0 \rightarrow \dots \rightarrow 6$ by this path.
- Else, keep the previous path as it is.
- Repeat for all such nodes.

Case 2: If there is no such edge:

- Now, we need to add a path from the source node to the newly added node.
- For all the nodes connected to newly added node (8), find the one which is closest to source node, i.e., having lowest cost from $0 \rightarrow \dots \rightarrow n$ (where n is a neighboring node of node 8).
- Add a path $0 \rightarrow \dots \rightarrow n \rightarrow 8$ in the path list (this would be the shortest path from source to newly added node).

In this way, we don't need to use dijkstra's algorithm everytime, a new node is added.

