

Andrew Lockwood

09/19/22

COMP 362L

LAB 04 *Algorithm*

My attempt a pseudo-code.

```
In monitor.c
argv [monitor temp] [number of nodes] [node 1 temp]...[node n temp]

config msg struct

for (number of nodes)
    create message queue
    create nodeData[i] array
        msgQid
        currentTemp
        stable(false)
    //clone node
    pid fork to execlp -> ./node
end
```

Each node gets a fork and node program

In node.c

```
nodeId = argv[1]
currTemp = argv[2]

stable_msg set to false//flag to end program

config msg struct//for msg

check msg_open for viable msg connect

print "Node" and "Temp"

report temp to monitor and reduce node temp
do
    //report the node's temp (mq_send())
    TO_MONITOR_Temp = currentTemp

    previousTemp = currentTemp

    if(received message from monitor)
        currentTemp = (previousTemperature * 3 + 2 * msg_rcvd.temperature) / 5

    print "NODE"

while(stable)
```

Andrew Lockwood

09/19/22

COMP 362L

LAB 04 *Algorithm*

Back in monitor, after forking nodes the nodes.

```
monitorTemp = argv[1] monitor temp

while(number_of_unchanged_nodes != numberOfNodes)

    //recieve all messages and sum temperatures
    //(wait for 1 less message for each stable node)
    for(all the nodes)
        sum node temps
    end

    previousTemp = newTemp
    new_temp = (2 * previous_temp + sum_of_client_temps) / 6

    determine if stability is reached
    for(all the nodes)
        if((fabsf(nodeData[i].currentTemperature - nodeData[i].previousTemperature) < 0.001)
        then node stability flag true

        //send new info to all nodes
        msg_send!
    end
end

//remove all outgoing queues
//remove incoming queue
freenode
exit
```