Advance Operating System Lab (CSD-416)
National Institute of Technology, Hamirpur

ADVANCE OPERATING SYSTEM LAB (CSD-416) ASSIGNMENT 5

Huang's termination detection algorithm

AKSHAT RAJ VANSH (185520)

December 11, 2021

Computer Science Department
National Institute of Technology, Hamirpur

Contents

1 Huang's termination detection algorithm

1.1 Code - Server

```
import socket as socket
import _thread
import threading
import random
from time import sleep
import time
from client import Client


class Server:
    def __init__(self, port, n, host=""):
        self.host = host
        self.port = port
        self.connection = []
        self.clients = []
        self.weight = 1

        self.server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        # loop through n times
        for i in range(n):
            self.clients.append(Client('127.0.0.1', self.port, i+1))

    def configure(self):
        try:
            self.server.bind((self.host, self.port))
            print("Server binded to port", self.port)
            self.server.listen(5)
            print("Server is listening")
        except Exception as e:
            print(e)

    def connect_clients(self):
        time.sleep(3)
        for i in range(len(self.clients)):
            print("Connecting to client {}".format(i+1))
            _thread.start_new_thread(self.clients[i].start, ())

    def decode(self, value):
        return value.decode('ascii')

    def encode(self, value):
        return value.encode('ascii')

    def listen(self, client, i):
        while True:
            data = client.recv(1024)
            data = self.decode(data)
            message = data[:data.find('(')]
            if(message == "C"):
                self.weight += float(data[data.find('(')+1:data.find(')')])
                print("{} Weight released by process {}".format(
                    data[data.find('(')+1:data.find(')')], i),)
                print("Process {} terminated".format(i))

```

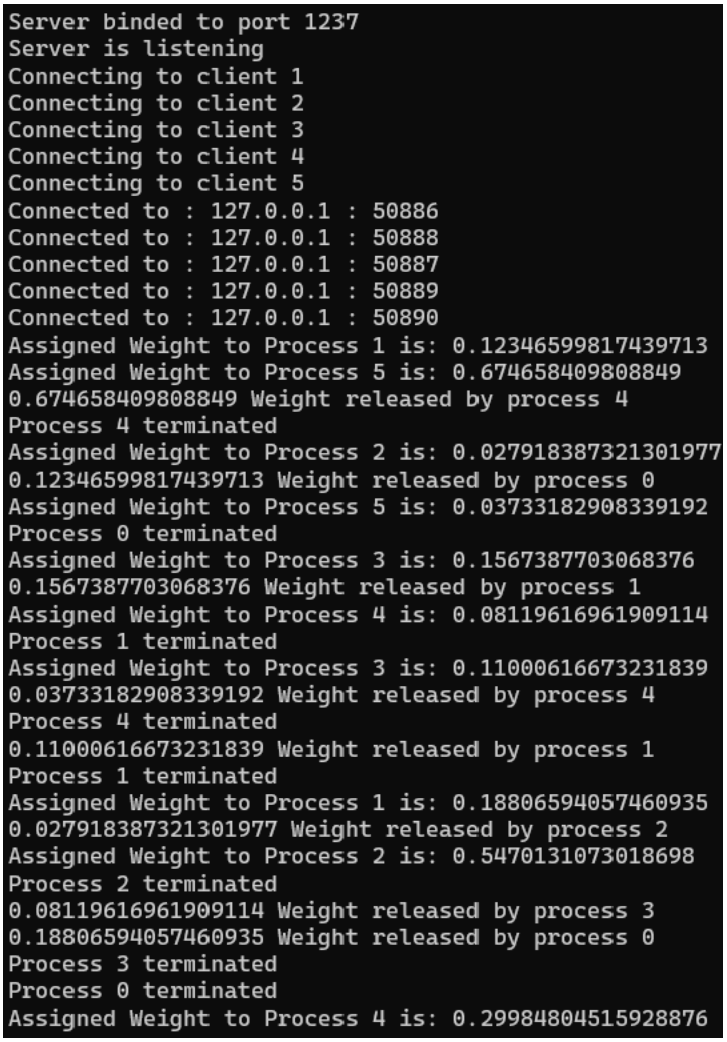
```
55      def threaded(self, client, client_addr):
56          while True:
57              if(self.weight > .2):
58                  i = random.randint(0, len(self.clients)-1)
59                  # print i and len of self.connection
60                  if(i < len(self.connection) and client == self.connection[i]):
61                      rweight = random.random() * (self.weight-.1)
62                      client.send(self.encode("B({})".format(str(rweight))))
63                      self.weight -= rweight
64                      _thread.start_new_thread(self.listen, (client, i))
65                  else:
66                      time.sleep(1)
67          client.close()
68
69      def start(self):
70          self.configure()
71          _thread.start_new_thread(self.connect_clients, ())
72          while True:
73              client, client_addr = self.server.accept()
74              self.connection.append(client)
75              print('Connected to :', client_addr[0], ':', client_addr[1])
76
77              _thread.start_new_thread(
78                  self.threaded, (client, client_addr))
79
80
81  if __name__ == '__main__':
82      server = Server(1237, 5)
83      server.start()
```

1.2 Code - Client

```
import socket
import json
import _thread
import time
import random
import threading


class Error:
    commandInputError = Exception("Please enter correct command")
    portInputError = Exception("Please enter correct port number")
    controllerError = Exception("Controller Error. Try After Sometime")
    createRoomError = Exception("Error in creating the room")


class Client:
    def __init__(self, host, port, id):
        self.id = id
        self.host = host
        self.port = port
        self.connections = []
        self.weight = ""

    def createSocket(self, port):
        client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        client.connect((self.host, port))
        return client

    def decode(self, value):
        return value.decode('ascii')

    def encode(self, value):
        return value.encode('ascii')

    def listen(self, client):
        while True:
            data = client.recv(1024)
            data = self.decode(data)
            message = data[:data.find('(')]
            if(message == "B"):
                self.weight = data[data.find('(')+1:data.find(')')]
                print("Assigned Weight to Process {} is: {}".format(
                    self.id, self.weight))
                time.sleep(random.randint(1, 20))
                client.send(self.encode("C({})".format(self.weight)))
        client.close()
        exit(0)

    def start(self):
        client = self.createSocket(self.port)
    # _thread.start_new_thread(self.send, (client,))
        _thread.start_new_thread(self.listen, (client,))
        while True:
            continue
```

1.3 Output

Huang's termination detection algorithm Output



```
Server binded to port 1237
Server is listening
Connecting to client 1
Connecting to client 2
Connecting to client 3
Connecting to client 4
Connecting to client 5
Connected to : 127.0.0.1 : 50886
Connected to : 127.0.0.1 : 50888
Connected to : 127.0.0.1 : 50887
Connected to : 127.0.0.1 : 50889
Connected to : 127.0.0.1 : 50890
Assigned Weight to Process 1 is: 0.12346599817439713
Assigned Weight to Process 5 is: 0.674658409808849
0.674658409808849 Weight released by process 4
Process 4 terminated
Assigned Weight to Process 2 is: 0.027918387321301977
0.12346599817439713 Weight released by process 0
Assigned Weight to Process 5 is: 0.03733182908339192
Process 0 terminated
Assigned Weight to Process 3 is: 0.1567387703068376
0.1567387703068376 Weight released by process 1
Assigned Weight to Process 4 is: 0.08119616961909114
Process 1 terminated
Assigned Weight to Process 3 is: 0.11000616673231839
0.03733182908339192 Weight released by process 4
Process 4 terminated
0.11000616673231839 Weight released by process 1
Process 1 terminated
Assigned Weight to Process 1 is: 0.18806594057460935
0.027918387321301977 Weight released by process 2
Assigned Weight to Process 2 is: 0.5470131073018698
Process 2 terminated
0.08119616961909114 Weight released by process 3
0.18806594057460935 Weight released by process 0
Process 3 terminated
Process 0 terminated
Assigned Weight to Process 4 is: 0.29984804515928876
```