|  |  |  |
| --- | --- | --- |
| Christian Client:  import datetime  import socket  from timeit import default\_timer as timer  from dateutil import parser  def client\_clk():  s = socket.socket()  port = 12345  s.connect(('127.0.0.1',port))  req\_tim = timer()  server\_t = parser.parse(s.recv(1024).decode())  res\_tim = timer()  latency = res\_tim - req\_tim  act\_time = datetime.datetime.now()  sync\_time = server\_t + datetime.timedelta(latency/2)  print("sync\_time is :",sync\_time)  print("act\_time is :",act\_time)  print("Difference is :",sync\_time-act\_time)  if \_name\_ == "\_main\_":  client\_clk() | Christian server  import socket  import datetime  def server():  s = socket.socket()  port = 12345  s.bind(('',port))  s.listen(5)  while True:  print("Waiting")  conn,addr = s.accept()  conn.send(str(datetime.datetime.now()).encode())  conn.close()  if \_name\_ == '\_main\_':  server() | |
| Berkeley’s Client:  from timeit import default\_timer as timer  from dateutil import parser  import threading  import datetime  import socket  import time  def startSendingTime(slave\_client):    while True:  slave\_client.send(str(  datetime.datetime.now()).encode())    print("Recent time sent successfully",  end = "\n\n")  time.sleep(5)    def startReceivingTime(slave\_client):    while True:  Synchronized\_time = parser.parse(  slave\_client.recv(1024).decode())    print("Synchronized time at the client is: " + \  str(Synchronized\_time),  end = "\n\n")    def initiateSlaveClient(port = 8080):    slave\_client = socket.socket()  slave\_client.connect(('127.0.0.1', port))  print("Starting to receive time from server\n")  send\_time\_thread = threading.Thread(  target = startSendingTime,  args = (slave\_client, ))  send\_time\_thread.start()    print("Starting to receiving " + \  "synchronized time from server\n")  receive\_time\_thread = threading.Thread(  target = startReceivingTime,  args = (slave\_client, ))  receive\_time\_thread.start()      if \_name\_ == '\_main\_':  initiateSlaveClient(port = 8080) | |
|  | |