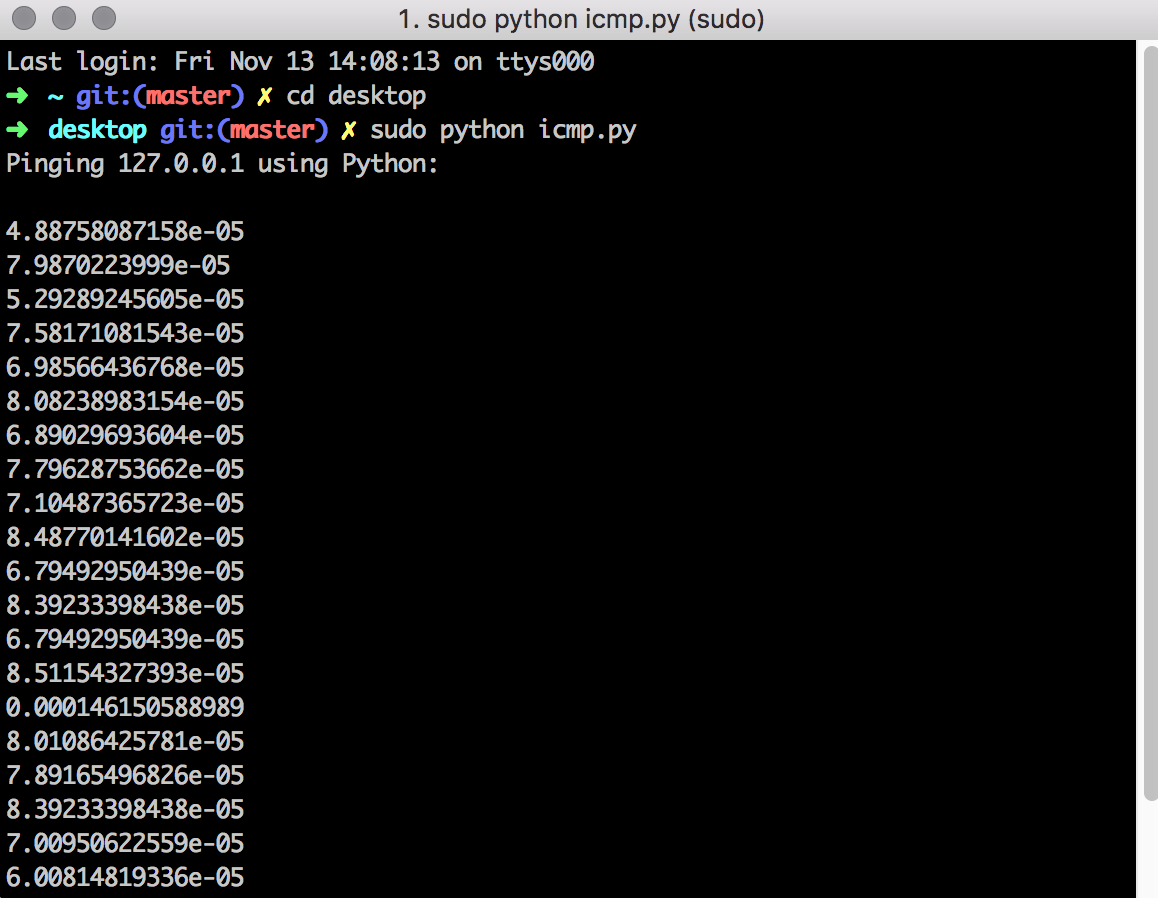
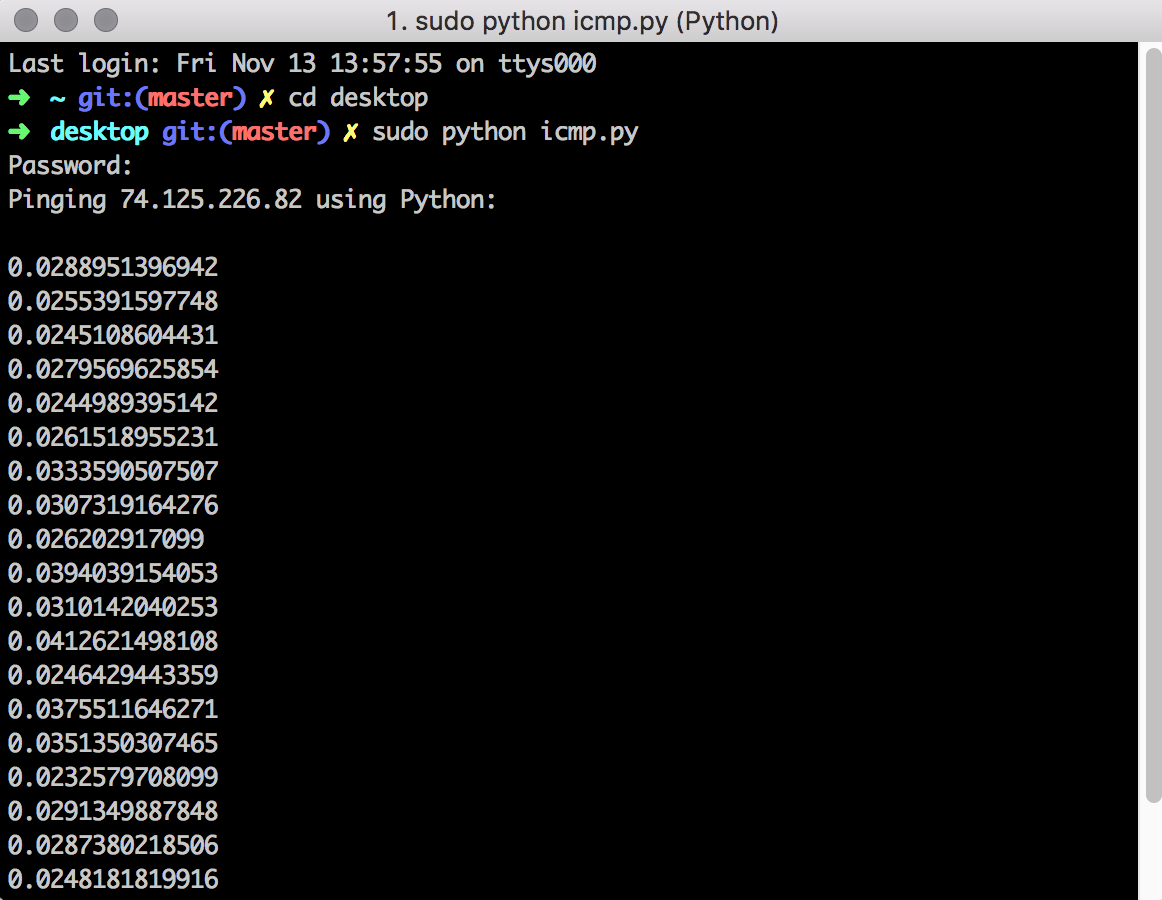
ICMP Program Lab

Ze He(zh700@nyu.edu)

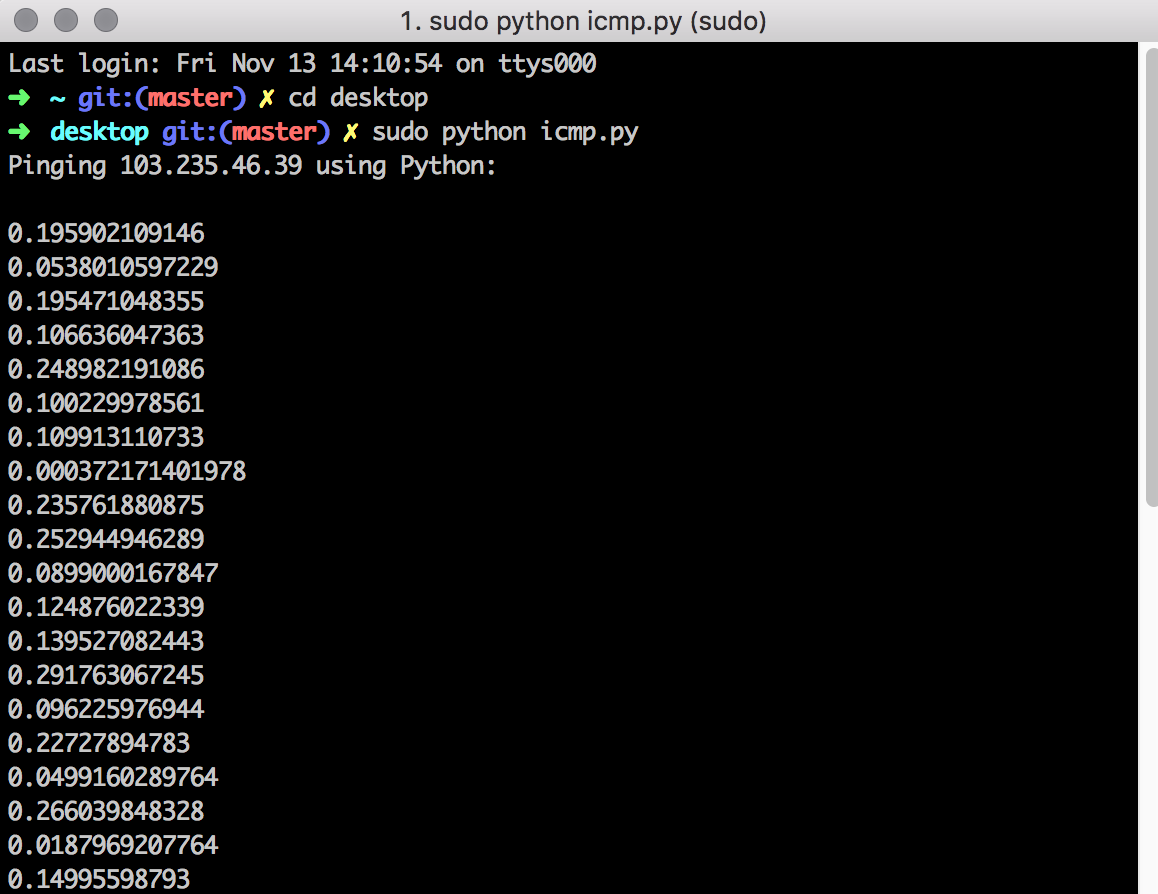
127.0.0.1(localhost)



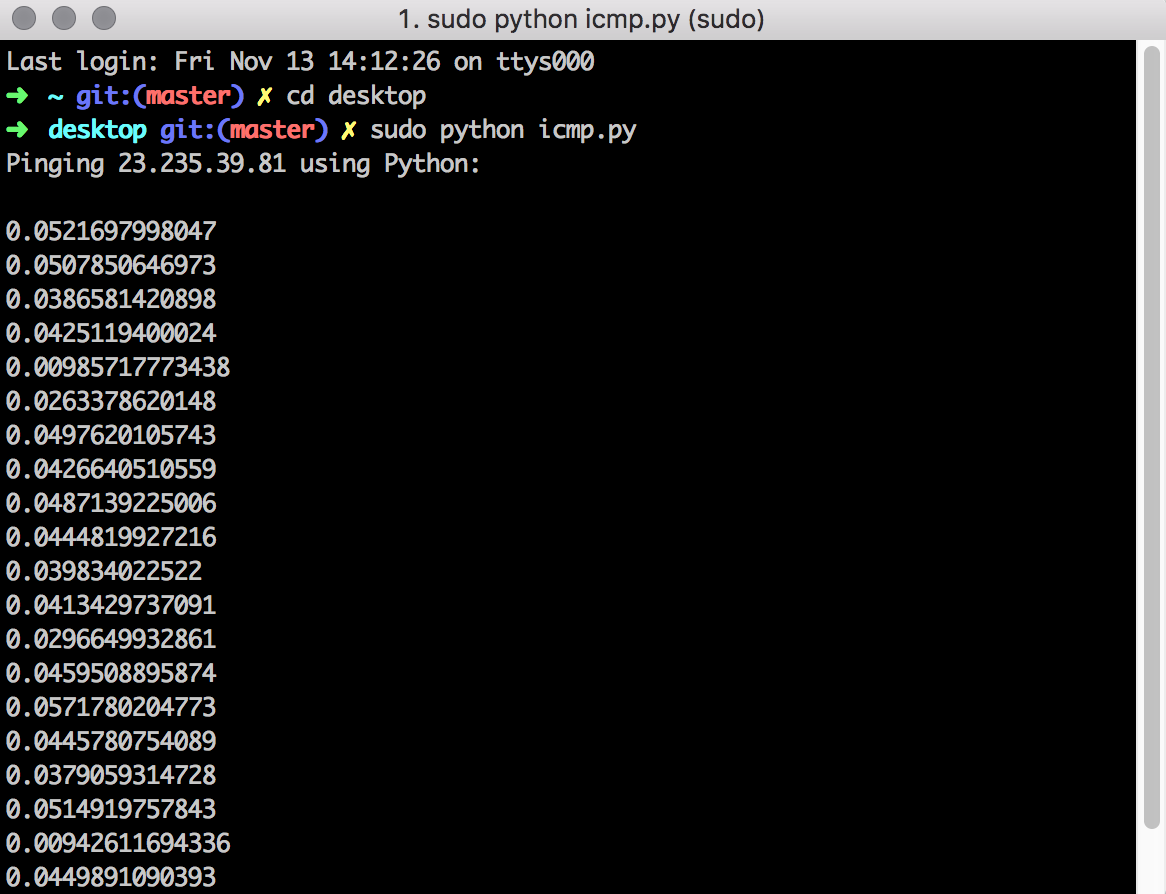
www.google.com(North America)



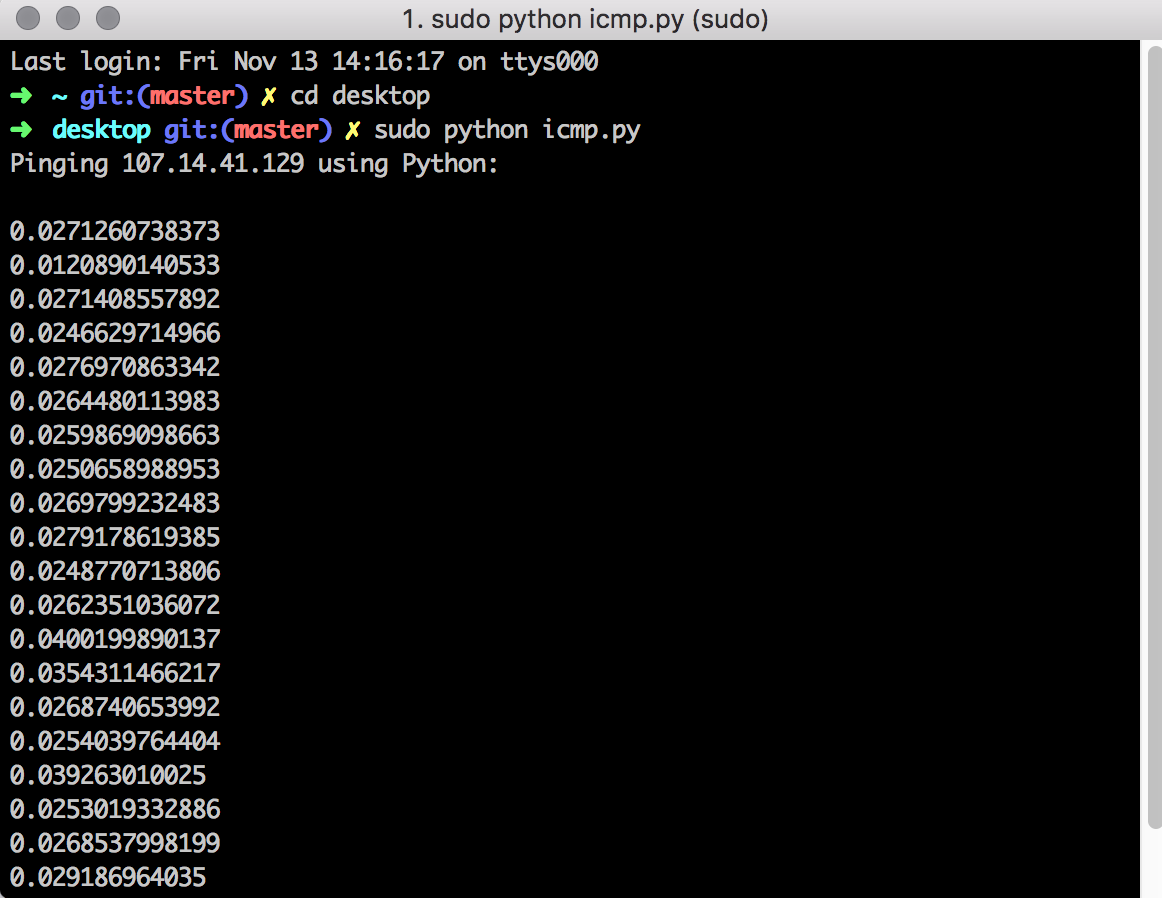
www.baidu.com(Asia)



www.bbc.com(Europe)



www. australia.com(Australia)



Code:

|  |
| --- |
| import socket  import os  import sys  import struct  import time  import select  import binascii  ICMP\_ECHO\_REQUEST = 8;  ERROR\_DESCR = {  1: ' - Note that ICMP messages can only be '  'sent from processes running as root.',  10013: ' - Note that ICMP messages can only be sent by'  ' users or processes with administrator rights.'  }  def checksum(str):  csum = 0  countTo = (len(str)/2)\*2    count =0  while count<countTo:  thisVal = ord(str[count+1])\*256 + ord(str[count])  csum = csum + thisVal  csum = csum & 0xffffffffL  count = count + 2    if countTo<len(str):  csum = csum + ord(str[len(str)-1])  csum = csum & 0xffffffffL    csum = (csum >> 16) + (csum & 0xffff)  csum = csum + (csum >> 16)  answer = ~csum  answer = answer & 0xffff  answer = answer >>8 | (answer <<8 & 0xff00)  return answer      def receiveOnePing(mySocket, ID, timeout, destAddr):  timeLeft = timeout    while 1:  startedSelect = time.time()  whatReady = select.select([mySocket],[],[],timeLeft)  howLongInSelect = (time.time()- startedSelect)  if whatReady[0]==[]:  return "Request timed out."  timeReceived = time.time()  recPacket,addr =mySocket.recvfrom(1024)  #Fill in start  icmp\_header = recPacket[20:28]  type, code, checksum, p\_id,sequence = struct.unpack('bbHHh',icmp\_header)  if p\_id == ID:  return timeReceived - startedSelect  #Fill in end  timeLeft = timeLeft - howLongInSelect  if timeLeft <=0:  return "Request timed out."  def sendOnePing(mySocket,destAddr,ID):  myChecksum = 0  header = struct.pack("bbHHh", ICMP\_ECHO\_REQUEST, 0, myChecksum, ID, 1)  data = struct.pack("d", time.time())  myChecksum = checksum(header + data)  if sys.platform == 'darwin':  myChecksum = socket.htons(myChecksum) & 0xffff  else:  myChecksum = socket.htons(myChecksum)  header = struct.pack("bbHHh", ICMP\_ECHO\_REQUEST, 0, myChecksum, ID, 1)  packet = header + data  mySocket.sendto(packet, (destAddr, 1))  def doOnePing(destAddr, timeout):  icmp = socket.getprotobyname("icmp")  #Fill in start  try:  mySocket = socket.socket(socket.AF\_INET,socket.SOCK\_RAW, icmp)  except socket.error as e:  if e.errno in ERROR\_DESCR:  raise socket.error(''.join((e.args[1],ERROR\_DESCR[e.errno])))  raise  try:  host= socket.gethostbyname(destAddr)  except socket.gaierror:  return  #Fill in end  myID = os.getpid() & 0xFFFF  sendOnePing(mySocket, destAddr, myID)  delay = receiveOnePing(mySocket, myID, timeout, destAddr)  mySocket.close()  return delay  def ping(host,timeout=1):  dest = socket.gethostbyname(host)  print "Pinging " + dest + " using Python:"  print ""  while 1 :  delay = doOnePing(dest, timeout)  print delay  time.sleep(1)  return delay  ping("127.0.0.1") |