```
#2018-08-17  Nam Vu  <npv14@cs16079ho>
from socket import *
from datetime import *
import sys
import select

def checkPacket(data):
    """This function is using for check the packet request as the require"""
    valid = True
    if len(data) != 6:
        valid = False
    if data[0:2].hex() != '497e':
        valid = False
    if data[2:4].hex() != '0001':
        valid = False
    if data[4:6].hex() != '0001' and data[4:6].hex() != '0002' :
        valid = False
    return valid

def dtResponse(dateOrTime, language, year, month, day, hour, minute):
    """This function is using for pack the packet response as the require"""
    pack = bytearray();
    pack += int(0x497e).to_bytes(2, byteorder='big')
    pack += int(0x0002).to_bytes(2, byteorder='big')
    if language == 1:
        pack += int(0x0001).to_bytes(2, byteorder='big')
    elif language == 2:
        pack += int(0x0002).to_bytes(2, byteorder='big')
    else:
        pack += int(0x0003).to_bytes(2, byteorder='big')
    pack += int(year).to_bytes(2, byteorder='big')
    pack += int(month).to_bytes(1, byteorder='big')
    pack += int(day).to_bytes(1, byteorder='big')
    pack += int(hour).to_bytes(1, byteorder='big')
    pack += int(minute).to_bytes(1, byteorder='big')
    #create dictionary to store the month's name of English, Maori and German
    dictMonthE = {1:"January", 2:"February", 3:"March", 4:"April", 5:"May" ,6:"June"
, 7:"July", 8:"August", 9:"September", 10:"October", 11:"November" ,12:"December" }
    dictMonthM = {1:"Kohit¯atea", 2:"Hui-tanguru", 3:"Pout¯u-te-rangi", 4:"Paenga-wh
¯awh¯", 5:"Haratua" ,6:"Pipiri", 7:"H¯ongongoi", 8:"Here-turi-k¯ok¯a", 9:"Mahuru", 1
0:"Whiringa-¯a-nuku", 11:"Whiringa-¯a-rangi" ,12:"Hakihea" }
    dictMonthG = {1:"Januar", 2:"Februar", 3:"M¨arz", 4:"April", 5:"Mai" ,6:"Juni",
7:"Juli", 8:"August", 9:"September", 10:"Oktober", 11:"November" ,12:"Dezember" }
    if language == 1:
        monthText = dictMonthE[month];
    elif language == 2:
        monthText = dictMonthM[month];
    elif language == 3:
        monthText = dictMonthG[month];
    #create the return text field with an appropriate language
    if dateOrTime == '0001':
        if language == 1:
            text = "Today¿s date is {} {}, {}".format(monthText, day, year)
        elif language == 2:
            text = "Ko te ra o tenei ra ko {} {}, {}".format(monthText, day, year)
        else:
            text = "Heute ist der {}. {} {}".format(day, monthText, year)
    else:
        if language == 1:
            text = "The current time is {}:{}".format(hour, minute)
        elif language == 2:
            text = "Ko te wa o tenei wa {}:{}".format(hour, minute)
        else:
            text = "Die Uhrzeit ist {}:{}".format(hour, minute)
    #Encode the text in to bytes Using String.encode('utf8')
    encodeText = text.encode('utf8')
    #Check and count the length of the Text field
    length = len(encodeText)
    if length <=  225:
        pack += int(length).to_bytes(1, byteorder='big')
        pack += encodeText
        print("Successul pack Data Response")
    else:
        print("The text exceeds 255 bytes")
    return pack

def main(englishPortNo, maoriPortNo, germanPortNo):
    #create Server UDP socket and bind with appropriate Port number
    try:
        serverSocketE = socket(AF_INET, SOCK_DGRAM)
        serverSocketE.bind(("127.0.1.1", englishPortNo))
        serverSocketM = socket(AF_INET, SOCK_DGRAM)
        serverSocketM.bind(("127.0.1.1", maoriPortNo))
        serverSocketG = socket(AF_INET, SOCK_DGRAM)
        serverSocketG.bind(("127.0.1.1", germanPortNo))
    except:
        print("There is something wrong with the UDP socket")
    while True:
        #Waiting for the packet request
        reader,_, _ = select.select([serverSocketE, serverSocketM, serverSocketG], [
], [])
        for i in reader:
            if i is serverSocketE:
                #found another socket connect to English Socket
                data = i.recvfrom(1024)
                ipSender = data[1]
                print("Recived request from server Socket English at IP:", ipSender)
                packetByteArray = bytearray(data[0])
                #Check the request packet and prepare the Packet response and send b
ack to the client
                if(checkPacket(packetByteArray)):
                    timeCurrent = str(datetime.now().time()).split(':')
                    dateToday = str(datetime.now().date()).split('-')
                    packet = dtResponse(packetByteArray[4:6].hex(), 1, dateToday[0],
 int(dateToday[1]), dateToday[2], timeCurrent[0], timeCurrent[1])
                    print(packet)
                    serverSocketE.sendto(packet,ipSender)
                    print("Sent Response Packet")
                else:
                    print("Invalid Packet")
                print("-----------------------------------------------------------------
-------------------")
            if i is serverSocketM:
                #found another socket connect to Maori Socket
                data = i.recvfrom(1024)
                ipSender = data[1]
                print("Recived request from server Socket Maori at IP:", ipSender)
                packetByteArray = bytearray(data[0])
                #Check the request packet and prepare the Packet response and send b
ack to the client
                if(checkPacket(packetByteArray)):
                    timeCurrent = str(datetime.now().time()).split(':')
                    dateToday = str(datetime.now().date()).split('-')
                    packet = dtResponse(packetByteArray[4:6].hex(), 2, dateToday[0],
 int(dateToday[1]), dateToday[2], timeCurrent[0], timeCurrent[1])
                    print(packet)
                    serverSocketE.sendto(packet,ipSender)
                    print("Sent Response Packet")
                else:
                    print("Invalid Packet")
                print("-----------------------------------------------------------------
-------------------")
            if i is serverSocketG:
                #found another socket connect to German Socket
                data = i.recvfrom(1024)
                ipSender = data[1]
                print("Recived request from server Socket German at IP:", ipSender)
                packetByteArray = bytearray(data[0])
                #Check the request packet and prepare the Packet response and send b
ack to the client
                if(checkPacket(packetByteArray)):
                    timeCurrent = str(datetime.now().time()).split(':')
                    dateToday = str(datetime.now().date()).split('-')
                    packet = dtResponse(packetByteArray[4:6].hex(), 3, dateToday[0],
 int(dateToday[1]), dateToday[2], timeCurrent[0], timeCurrent[1])
                    print(packet)
                    serverSocketE.sendto(packet,ipSender)
                    print("Sent Response Packet")
                else:
                    print("Invalid Packet")
                print("-----------------------------------------------------------------
-------------------")

def start():
    """This function is using for start the program"""
    englishPortNo = int(sys.argv[1])
    maoriPortNo = int(sys.argv[2])
    germanPortNo = int(sys.argv[3])
    if englishPortNo != maoriPortNo and maoriPortNo != germanPortNo and englishPortN
o != germanPortNo and 1024 < englishPortNo < 64000 and 1024 < maoriPortNo < 64000 an
d 1024 < germanPortNo < 64000:
        main(englishPortNo, maoriPortNo, germanPortNo)
    else:
        print("Input Error the Server is closed")
start()

"""
Command codes for testing the program:
    python3 Server.py 63999 63998 63997
"""
```