Part A: Write a program to connect to port 80 on nigelward.com; request the index.html file, and report the value of bytes 1012 through 1028.

#!/usr/bin/env python3

from socket import \*

#part A

s = socket(AF\_INET, SOCK\_STREAM)

s.connect(("nigelward.com", 80))

s.sendall(("GET /index.html HTTP/1.1\r\n" +

"Host: nigelward.com\r\n" +

"Accept: text/html\r\n" +

"Connection: close\r\n\r\n").encode())

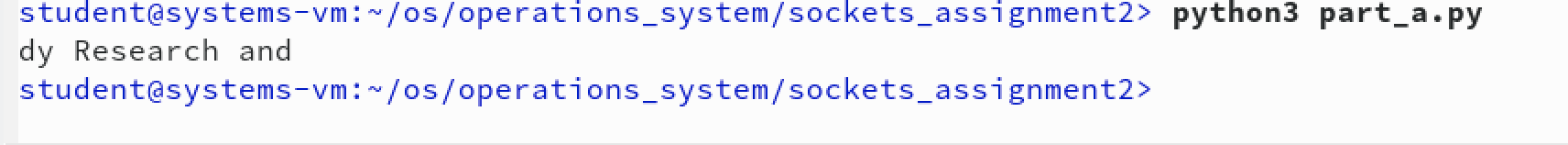
data = s.recv(1012).decode()

data = s.recv(16).decode()

print(data)

s.close()

OUTPUT:



Part B: Python is only willing to send bytes via sockets, so if I want

to send the number 256, for example, I have to do something on the

server side, like str(256) + '\n' and then on the client side

something like int(whatWasReceived). Some people prefer to do

(256).to\_bytes(2, byteorder=big) on the server side, and on the

client side do int.from\_bytes(received, byteorder=big).

1. Modify server-demo.py and your client to see if you can really send numbers this way.

Client Server

#!/usr/bin/env python3

from socket import \*

s = socket(AF\_INET, SOCK\_STREAM)

s.connect(("127.0.0.1", 7069))

while 1:

data = s.recv(10000)

data = int.from\_bytes(data, byteorder="big")

if data == 0:

break

print(data)

s.close()

#!/usr/bin/env python3

from socket import \*

import time

s = socket(AF\_INET, SOCK\_STREAM)

s.bind(("127.0.0.1", 7069))

s.listen(5)

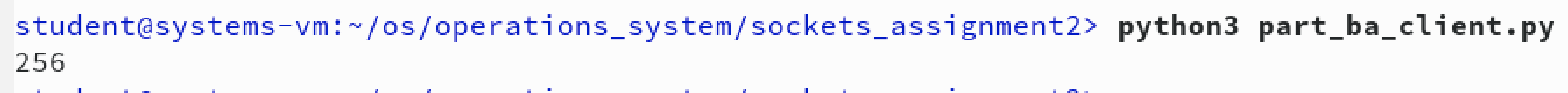
while True:

c,a = s.accept()

data = c.send((256).to\_bytes(2, byteorder = 'big'))

c.close()

Output:



1. Show what happens if one side instead uses byteorder=small.

Client

#!/usr/bin/env python3

from socket import \*

import time

s = socket(AF\_INET, SOCK\_STREAM)

s.connect(("127.0.0.1", 7069))

while 1:

data = s.recv(10000)

data = int.from\_bytes(data, byteorder="big")

#data = int.from\_bytes(data, byteorder="big")

if data == 0:

break

print(data)

s.close()

#!/usr/bin/env python3

from socket import \*

import time

s = socket(AF\_INET, SOCK\_STREAM)

s.bind(("127.0.0.1", 7069))

s.listen(5)

#data = s.send((256).to\_bytes(2, byteorder = 'big'))

while True:

c,a = s.accept()

data = c.send((256).to\_bytes(2, byteorder = 'little'))

#data = c.send((256).to\_bytes(2, byteorder = 'small'))

c.close()

Server

Output:

