NWEN241 2014 Assignment 1

This assignment is worth 2.5 marks overall.

Your answer should be in the form of a report that provides clear evidence of testing. Your code should be well commented – submitting code without comments will result in low marks.

Submit your report via the Submission system as a PDF

Background

Computers attached to the Internet communicate using a model known as **the client-server model**. Wikipedia **http://en.wikipedia.org/wiki/Client%E2%80%93server_model** gives this description:

The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system.

Examples of computer applications that use the client–server model are email, network printing, and the world wide web.

In this assignment we will build a pair of programs that communicate with each other via a network to perform the task of retrieving a text file from a remote system and displaying its contents.

The templates used in this assignment (Appendix 1) are derived from examples provided as part of the Python documentation found at **http://docs.python.org/3/library/socket.html#example**. You should not need to make major changes to this code.

IP addresses of server

As part of this exercise you will need to know the IP address(es) of the machine your server code is running on. You can find this out on the ECS workstations using the command:

```
ifconfig eth0
```

You should see output like:

```
[asjl@reds /]$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 130.195.4.152  netmask 255.255.255.128  broadcast 130.195.4.255
        inet6 2404:2000:2000:4:fab1:56ff:fea2:d845  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::fab1:56ff:fea2:d845  prefixlen 64  scopeid 0x20<link>
        ether f8:b1:56:a2:d8:45  txqueuelen 1000  (Ethernet)
        RX packets 21700498  bytes 31117908048 (28.9 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4573683  bytes 1042849784 (994.5 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf7c00000-f7c20000
```

Look for the lines that begin with **inet** and **inet6**. The IPv4 address of the host in this example is 130.195.4.152 and the IPv6 address is 2404:2000:2000:4:fab1:56ff:fea2:d845. (You can ignore the fe80::fab1:56ff:fea2:d845 address.)

Your tasks

1. Create a directory for your project using the **mkdir** command. e.g.

   mkdir NWEN241

2. Copy the file **Assignment1.tgz** to that directory and unpack it using:

   tar xvzf Assignment1.tgz

3. Your directory should contain a file with this text and two directories, src and data, containing copies of the

client and server programs and some sample data files.

4. Using the templates supplied and the material discussed in class, build a pair of programs that request one of the sample data files from the server and display the contents.

5. Test your program to make sure that you cannot retrieve files outside the **data** directory. Pay particular attention to use of the Unix **../** construct.

Submitting your work

Your answer should be in the form of a report that provides **clear evidence of testing**. Your code should be well commented – **submitting code without comments** will result in low marks.

Submit your report via the Submission system as a single PDF file containing your report and copies of your code.

Appendix 1

server.py

```
#!/usr/bin/env python3

# Sample server program
import os
import sys
import socket
import signal

#
# get the name of our program to use in error messages
#
progname = os.path.basename(sys.argv[0])

############################################################################
# WRITE CODE: get the IP address of the server from the command line.
# If no command line parameter is given, prompt the user for the information
#
host = ????
############################################################################
#
# Arbitrary non-privileged port
#
port = 50007

############################################################################
# This section opens a socket connection to the remote machine
# You probably don't need to make changes here
#
s = None
for res in socket.getaddrinfo(host, host, socket.AF_UNSPEC,
                              socket.SOCK_STREAM, 0, socket.AI_PASSIVE):
    af, socktype, proto, canonname, sa = res
    print ('Listening on', sa)
    try:
        s = socket.socket(af, socktype, proto)
        s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    except socket.error as msg:
        s = None
        continue
    try:
        s.bind(sa)
        s.listen(1)
    except socket.error as msg:
        s.close()
        s = None
        continue
    break
```

```
if s is None:
    print ('could not open socket')
    sys.exit(1)
#
# We're going to listen for socket connections in an infinite loop
# We need a signal handler to catch Ctrl-C
#
def handler(signum, frame):
    s.close()
    sys.exit(1)

signal.signal(signal.SIGINT, handler)
############################################################################

while True:
    print('Waiting for connection...')
    conn, addr = s.accept()
    print ('Connection from', addr)
    while True:
        data = conn.recv(1024)
        if not data:
            break
        print (data.decode('utf-8'))
        ####################################################################
        #
        # WRITE CODE: process incoming request here
        #
        ????
        # conn.send(data.encode('utf-8'))
        ####################################################################
    conn.close()

s.close()
sys.exit()
```

client.py

```
#!/usr/bin/env python3

# Sample client program
import os
import sys
import socket

#
# get the name of our program to use in error messages
#
progname = os.path.basename(sys.argv[0])

############################################################################
# WRITE CODE: get the hostname or IP address of the server from the
# command line. If no command line parameter is given, prompt the user
# for the information
#
host = ????
#
# WRITE CODE: get the filename requested from the server from the
# command line. If no command line parameter is given, prompt the user
# for the information
#
filename = ????
#
# End of input parameters
############################################################################
#
# The same port as used by the server
#
```

```
port = 50007

#############################################################################
# This section opens a socket connection to the remote machine
# You probably don't need to make changes here
#
s = None
for res in socket.getaddrinfo(host, port, socket.AF_UNSPEC, socket.SOCK_STREAM):
    af, socktype, proto, canonname, sa = res
    print("Connecting to", sa)
    try:
        s = socket.socket(af, socktype, proto)
    except socket.error as msg:
        s = None
        continue
    try:
        s.connect(sa)
    except socket.error as msg:
        s.close()
        s = None
        continue
    break
if s is None:
    print('could not open socket')
    sys.exit(1)
#############################################################################


#############################################################################
#
# Send an 'utf-8' encoded version of the 'filename' to the remote machine
#
s.sendall(filename.encode('utf-8'))
#
# WRITE CODE: get response from remote machine and display the result
#
????
#############################################################################
#
# Close the socket now that we're finished with it
#
s.close()
```

Appendix 2

sample1.txt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

sample2.txt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer nec odio. Praesent libero. Sed cursus ante dapibus diam. Sed nisi. Nulla quis sem at nibh elementum imperdiet. Duis sagittis ipsum. Praesent mauris. Fusce nec tellus sed augue semper porta. Mauris massa. Vestibulum lacinia arcu eget nulla. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

Curabitur sodales ligula in libero. Sed dignissim lacinia nunc. Curabitur tortor. Pellentesque nibh. Aenean quam. In scelerisque sem at dolor. Maecenas mattis. Sed convallis tristique sem. Proin ut ligula vel nunc egestas porttitor. Morbi lectus risus, iaculis vel, suscipit quis, luctus non, massa. Fusce ac turpis quis ligula lacinia aliquet.

Mauris ipsum. Nulla metus metus, ullamcorper vel, tincidunt sed, euismod in, nibh. Quisque volutpat condimentum velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nam nec ante. Sed lacinia, urna non tincidunt mattis, tortor neque adipiscing diam, a cursus ipsum ante quis turpis. Nulla facilisi. Ut fringilla. Suspendisse potenti. Nunc feugiat mi a tellus consequat imperdiet. Vestibulum sapien. Proin quam. Etiam ultrices. Suspendisse in justo eu magna luctus suscipit.

Sed lectus. Integer euismod lacus luctus magna. Quisque cursus, metus vitae pharetra auctor, sem massa mattis sem, at interdum magna augue eget diam. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi lacinia molestie dui. Praesent blandit dolor. Sed non quam. In vel mi sit amet augue congue elementum. Morbi in ipsum sit amet pede facilisis laoreet. Donec lacus nunc, viverra nec, blandit vel, egestas et, augue. Vestibulum tincidunt malesuada tellus. Ut ultrices ultrices enim. Curabitur sit amet mauris.

Morbi in dui quis est pulvinar ullamcorper. Nulla facilisi. Integer lacinia sollicitudin massa. Cras metus. Sed aliquet risus a tortor. Integer id quam. Morbi mi. Quisque nisl felis, venenatis tristique, dignissim in, ultrices sit amet, augue. Proin sodales libero eget ante. Nulla quam. Aenean laoreet. Vestibulum nisi lectus, commodo ac, facilisis ac, ultricies eu, pede. Ut orci risus, accumsan porttitor, cursus quis, aliquet eget, justo. Sed pretium blandit orci.