

Lesson Plan 8 | Form 2 | Group Problem Solving with Network Sockets

Objective

Students will practice their problem-solving skills by modifying their Python program from LP7 independently and/or with their classmates in order to automate a tedious manual process.

Warm-up

None.

Presentation

Emphasize the importance of being able to solve problems independently and with others, and preface by acknowledging that today's lesson will be challenging.

Guided Practice

None.

Independent Practice (requires local wireless network)

Prior to class, place the listener Python program (see below for code) on the central server and run it.

In the lab, have the students retrieve from the central file server their Python program from LP7. Ask them to run it, and this time have them note the differences in what the server tells/asks them. Ask for students to explain what is different, and make sure everyone is clear about what the server is asking (if needed, draw a set of the numbers 1 through 5 on the board, cross off each, and have them call out how many numbers are left to cross off).

Then explain that the server wants the students to do the same thing, but using a set of 100 numbers rather than 5. Ask the students to change their Python program so that it gives us exactly those 100 numbers, and sends those to the server, rather than asking the user to enter each number.

As this is independent practice, it's advisable only to give occasional hints as needed to further class/individual progress, but not explicitly tell students what to write.

Closing

Have students save their program to their central file directory for additions, changes, etc. in later lessons.

In Hindsight

02/01/2017: As expected, some students made progress more independently than others. No one immediately got the solution without guidance or a hint, but several arrived at the solution before

the end of class without me standing over them looking on, which to me is the best possible outcome of this lesson. Those who finished early were encouraged to go around and help their classmates, and by the end of the lesson, everyone had changed their program to send all the numbers to the server. One big hurdle many students had was removing the `raw_input()` call; they changed their `while` loop to a `for` loop, but still asked the user for input on each iteration. Maybe more emphasis is needed up-front on the fact that the numbers 1 through 100 are easily available to us in Python (i.e., via `range()`), and thus we don't need to ask the user for any additional information. Overall, I've seen enough to believe that the direction of this lesson and LP7 before it provides the right amount of challenge/rigor in a domain (networking) that keeps most of them interested.

02/03/2017: In the second stream, one student impressively found the solution to this within five minutes. As with the other stream, most people understood why the `while` loop needed to be replaced, but they got hung up on sending the variable instead of the user's input. I think many of the students are hesitant to delete code because they think everything I give them will always be necessary, which will take time to sway. Overall, students enjoyed the collaboration in this lesson just as in the other stream.

Server Program (differences from LP7 program are highlighted)

```
import socket, sys
from thread import start_new_thread
HOST = "
PORT = 4002
MAX_CONNS = 100
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.bind((HOST, PORT))
    s.listen(MAX_CONNS)
    print "Socket created successfully on port: %d, now listening..." % PORT
except Exception:
    print "Error creating socket."
    sys.exit(1)
def client_thread(conn):
    print "Handling connection from %s:%s" % (addr[0], addr[1])
    nums_remaining = set(range(1, 101))
    conn.sendall("Hello " + addr[0] + ", please send me all the integers from 1 to 100.")
    while True:
        data = conn.recv(128)
        if not data:
            break
        try:
            user_num = int(data.strip())
            if user_num < 1 or user_num > 100:
                conn.sendall("The number " + str(user_num) + " is not between 1 and 100, try again.")
```

```

elif user_num in nums_remaining:
    nums_remaining.remove(user_num)
    conn.sendall("You sent me " + str(user_num) + \
        "; that was the last number, well done. Bye." if len(nums_remaining) == 0 \
        else "You sent me " + str(user_num) + \
        "; you have " + str(len(nums_remaining)) + " numbers left to send me.")
    if len(nums_remaining) == 0:
        break
except:
    conn.sendall("You didn't send me an integer, try again.")
conn.close()
print "Closed connection from %s:%s" % (addr[0], addr[1])
try:
    while True:
        # blocking call
        conn, addr = s.accept()
        start_new_thread(client_thread, (conn,))
except KeyboardInterrupt:
    print "Interrupt received, closing server."
    s.close()

```

Client Program (differences from LP7 program are highlighted)

```

import socket

IP = '10.0.2.100'
PORT = 4002

s = socket.socket()
s.connect((IP, PORT))
print s.recv(256)

for i in range(1, 101):
    s.send(str(i))
    print s.recv(256)

s.close()

```