Tutorial creating a concurrent datagram server

- boot into GNU/Linux and using an editor create this UDP server

- **udpserver.py**

```
from socket import *
from os import fork
import sys
import time
myHost = ""
myPort = 2000

# create a UDP socket
s = socket(AF_INET, SOCK_DGRAM)
# bind it to the server port number
s.bind((myHost, myPort))
```

Tutorial creating a concurrent datagram server

udpserver.py

```
while True:
    data, address = s.recvfrom(1024)
    print "UDP server:", data, "from", address
    if data:
        start_time = time.time()
        print "processing request received"
        time.sleep (5)
        end_time = time.time()
        print "processing took: ", end_time-start_time, "seconds"
        s.sendto("echo -> " + data, address)
    else:
        break
```

Tutorial creating a concurrent datagram server

- notice that we are simulating some large amount of server processing with the call to `time.sleep(5)`

- now open up the editor again and write the following into the file `udpclient.py`

Tutorial creating a concurrent datagram server

udpclient.py

```
#!/usr/bin/python

import sys, time
from socket import *

serverHost = "localhost"
serverPort = 2000
```

Tutorial creating a concurrent datagram server

udpclient.py

```
# create a UDP socket
s = socket(AF_INET, SOCK_DGRAM)

s.connect((serverHost, serverPort))
start_time = time.time()
s.send("Hello world")
data = s.recv(1024)
end_time = time.time()
print data

print "time to send to server and get reply was", end_time - start_time, "seconds"
```

Tutorial creating a concurrent datagram server

- now open up four command line terminal windows
  - order then neatly on the screen so that all four are visible and not overlapping
- start the server in one window
-
```
$ python udpserver.py
```
- in the other three terminals type the following but do not press <enter>
-
```
$ python udpclient.py
```

Tutorial creating a concurrent datagram server

- now quickly press the enter key in all three client windows and observe what happens
  - write down what you are seeing?

- why are you seeing this?

Implementing a concurrent datagram server

- using the `udpserver.py` code from the above slides, see if you can convert it into a concurrent server

- a successful implementation will allow both clients to connect simultaneously but will only incur a 5 second delay for a transaction

Implementing a concurrent datagram server

- hint you should read the lecture notes for the pseudo code for the udp concurrent server
- hint in Python the `exit(0)` system call is `sys.exit(0)`
- hint in Python the `fork()` system call is also `fork()`
- both can be used in your `udpserver.py` example code