SENG 462 Tutorial #9

Chris Pearson
pearson@csc.uvic.ca

University of Victoria

Need Help?

- Schedule time with me for tomorrow

B203 Servers

- Why are people using all of them?
- Do we need to assign certain machines to certain groups?
- Do we a scheduling mechanism for who is using the servers when?

Phone & Server Communication

- Why is everybody using HTTP?
- Sockets are simple…

```
import socket
# Create the socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 80))             # Bind the socket to a port
s.listen(1)                  # Allow 1 waiting connection
conn, addr = s.accept() # Wait for a connection
# Could spawn a thread for the connection here,
# then return to listening for new connections
while 1:                     # Echo the data back to the sender
    data = conn.recv(1024)
    if not data: break
    conn.send(data)
conn.close()                 # Close the connection
```

Send A Command

- Use Python objects to send a command
  1. Set up a socket listen thread on a server
  2. The phone creates an object that encapsulates a command
  3. The phone uses Pickle to serialize the object
     - `a_string = pickle.dumps(my_object)`
  4. The phone sends the string to the server
     - `a_socket.write(a_string)`
  5. The server unpickles the object
     - `my_object = pickle.load(a_string)`
  6. The server does its processing magic

University of Victoria

Send Back The Results

- Use Python objects to send back results
  1. Server creates an object the encapsulates the results of the command
  2. The server uses Pickle to serialize the results object
     - `a_string = pickle.dumps(results_object)`
  3. The server sends the string to the phone
     - `a_socket.write(a_string)`
  4. The phone unpickles the object
     - `results_object = pickle.load(a_string)`
  5. The phone uses the results directly as with any other Python object

University of Victoria

Database Connection Pooling

- Problem:
  - Need to keep database connections open
  - Web servers are stateless
- What do you do?

HI! I'M A SERVER! WHO ARE YOU?
I'M A BROWSER. I'D LIKE TO SEE THIS ARTICLE.
OH BOY! I CAN HELP! LET ME GET IT FOR— ...WHOA! YOU'RE A SMARTPHONE BROWSER?
YEAH.
COOOOL! HEY, I'VE GOT THIS NEW MOBILE VERSION OF MY SITE! CHECK IT OUT!
ISN'T IT PRETTY?
SURE. BUT THIS IS JUST YOUR MOBILE SITE'S MAIN PAGE. WHERE'S THE ARTICLE I WANTED?
WHAT ARTICLE?
THE ONE I—
WHO ARE YOU?
I—
HI! I'M A SERVER!

Source: xkcd.com/869

Database Connection Pooling

- The idea:
  - Run an application as your transaction server.
  - Make several connections to the database
  - Hand those connections to needing threads
  - Keep track of whether connections are in use or not
- This can mean a 10x performance increase

University of Victoria

Simple Data Storage

- `e32dbm`
- Stores data in (key, value) pairs, similar to Python dictionaries
- All data is stored as Unicode strings
- See Section 5.4, *S60 Module Reference 2.0*

University of Victoria

e32dbm – Setup

```
import e32dbm

DBPATHNAME = u"c:\\Data\\Others\\AccelGrapher"
DBFILENAME = DBPATHNAME + u"\\agraph2.db"

# create the directory for the database, if it
  doesn't already exist
if not os.path.exists(DBPATHNAME):
  os.makedirs(DBPATHNAME)
```

e32dbm – Load Values

```
# try to load values from the database
try:
  db = e32dbm.open(DBFILENAME, "r")
  currentTab = int(db[u"currentTab"])
  thicknessLines = int(db[u"thicknessLines"])
  thicknessDots = int(db[u"thicknessDots"])
  db.close()
# use default values if the database or one of its values
  isn't available
except:
  currentTab = 0
  thicknessLines = 5
  thicknessDots = 5
```

University of Victoria

e32dbm - Save Values

```
# Write the state variables out to the database
# c - opens the database for reading and writing (and
  creates a new database if the file does not exist)
# f - the database is not updated on disk until you close
  it or force it to be written to disk by using a special
  function.
try:
  db = e32dbm.open(DBFILENAME, "nf")
  db[u"currentTab"] = str(currentTab)
  db[u"thicknessLines"] = str(thicknessLines)
  db[u"thicknessDots"] = str(thicknessDots)
  db.close()
except:
  pass
```

University of Victoria