CS 360 Internet Programming

Ruby

Ruby Network Programming

Daniel Zappala
Computer Science
Brigham Young University

1. High-Level Modules
   - NET::HTTP
   - NET::FTP
   - NET::SMTP

2. Socket Library
   - Socket Class
   - TCPSocket Class
   - TCPServer Class

NET::HTTP

- fetch headers and web page contents using HTTP

```
require 'net/http'

Net::HTTP.start('www.pragmaticprogrammer.com') do |http|
  response = http.get('/index.html')
  puts "Code = #{response.code}"
  puts "Message = #{response.message}"
  response.each {|key, val| printf "%-14s = %-40.40s\n", key, val }
  puts response.body[0,100]
end
```

NET::FTP

- use ftp, including anonymous and password-authenticated sessions

```
require 'net/ftp'

ftp = Net::FTP.new('ruby-lang.org')
ftp.login
ftp.chdir('pub')
files = ftp.list()
puts files
ftp.getbinaryfile('ruby-core.tar.bz2', 'ruby-core.tar.bz2', 1024)
ftp.close
```

NET::SMTP

- send e-mail

```
require 'net/smtp'

Net::SMTP::start('smtp.comcast.net',25) do |smtp|
  smtp.open_message_stream('zappala@cs.byu.edu', # from
                             [ 'zappala@cs.byu.edu' ]          # to
                             ) do |stream|
    stream.puts "From: Daniel Zappala <zappala@cs.byu.edu>"
    stream.puts "To: Daniel Zappala <zappala@cs.byu.edu>"
    stream.puts "Subject: test message"
    stream.puts "Date: Sat, 23 Feb 2010 12:26:43 +0700"
    stream.puts
    stream.puts "This is a test of the Future Email System"
  end
end
```

Socket Library Hierarchy

BasicSocket: base class

- IPSocket
    - TCPSocket
        - SOCKSSocket
        - TCPServer
    - UDPSocket
- Socket
- UNIXSocket
    - UNIXServer

Socket Class

- direct access to BSD socket API
- need to use `pack_sockaddr_in` to create string representation of socket address structure

```
require 'socket'

addr = Socket.pack_sockaddr_in(80,'ilab.cs.byu.edu')
sock = Socket.new(Socket::AF_INET,Socket::SOCK_STREAM,0)
sock.connect(addr)
sock.send("GET / HTTP/1.1\r\nHost: ilab.cs.byu.edu\r\n\r\n",0)
puts sock.recv(1000)
sock.close()
```

Socket Class

- use open() instead of new() to pass in a block

```
require 'socket'

addr = Socket.pack_sockaddr_in(80,'ilab.cs.byu.edu')
Socket.open(Socket::AF_INET,Socket::SOCK_STREAM,0) do |sock|
  sock.connect(addr)
  sock.send("GET / HTTP/1.1\r\nHost: ilab.cs.byu.edu\r\n\r\n",0)
  puts sock.recv(1000)
end
```

Socket Server

```
require 'socket'

server = Socket.new(Socket::AF_INET,Socket::SOCK_STREAM,0)
server.setsockopt(Socket::SOL_SOCKET,Socket::SO_REUSEADDR,true)
address = Socket.pack_sockaddr_in(8000,'localhost')
server.bind(address)
server.listen(5)
client,address = server.accept
data = client.recv(1000)
client.send(data,0)
```

TCPSocket

- creates socket, initializes address, connects to server

```
client = TCPSocket.new('localhost',8000)
message = "hello"
client.send(message,0)
response = client.recv(1000)
puts response
```

TCPServer

- creates socket, initializes address, binds, listens

```
server = TCPServer.new('localhost',8000)
session = server.accept
message = session.gets
response = "goodbye"
session.puts response
```

Code Examples

- `http://ilab.cs.byu.edu/cs360/code/ruby-net.tgz`