14 - Lecture - Sockets and HTTP

Sockets API

Recommended reading:

Beej’s Guide to Network Programming

Chapter 2: What is a socket?
Chapter 3: IP Addresses, structs, and Data Munging
Chapter 5: System Calls or Bust

- available online: http://beej.us/guide/bgnet/
- excellent, up-to-date, and very detailed sockets tutorial

Lecture slides and code examples:

http://cs.baylor.edu/˜donahoo/practical/CSockets/

- This is a web site for a sockets programming book by Michael Donahoo and Ken Calvert. This book is not required for this course (although it’s a pretty good book).

  I’m just using their slides and the example code (TCPEchoClient.c, DieWithError.c, TCPEchoServer.c, HandleTCPClient.c) for this lecture.

- Note that the URL is for the 1st edition of the book. The 2nd edition of the book is available, but I am using the slides and sample code from the older 1st edition.

Network client/server example:

TCPEchoClient/Server code walk-through

Sending and receiving bytes through a socket connection:

```
send(int socket, const void *buffer, size_t length, int flags);
```

- normally, send() blocks until it sends all bytes requested
- returns num bytes sent or -1 for error

```
recv(int socket, void *buffer, size_t length, int flags);
```

- normally, recv() blocks until it has received at least 1 byte
- returns num bytes received, 0 if connection closed, -1 if error

HTTP

Recommended reading:

HTTP Made Really Easy: A Practical Guide to Writing Clients and Servers

- available online: http://www.jmarshall.com/easy/http/

HTTP protocol in action:

An example of HTTP exchange using netcat.

HTTP 1.0 v. HTTP 1.1

- persistent connection

Dynamic web page

- HTML form:

```
<FORM METHOD=GET ACTION="/mdb-lookup">
    Lookup: <INPUT TYPE="text" NAME="key">
    <p>
    <INPUT TYPE="submit" VALUE="Lookup">
</FORM>
```

when the text box is filled with "abc" and button clicked, the browser sends:

```
GET /mdb-lookup?key=abc HTTP/1.1
```

- How do dynamic web sites manage sessions?

Server "feeds" cookie at the start of session:

```
HTTP/1.1 200 OK
Content-type: text/html
Set-Cookie: name=value
```

Browser stores the cookies in a local file, indexed by web sites, and on subsequent visits, it brings the corresponding cookie along:

```
GET /index.html HTTP/1.1
Cookie: name=value
```