CS594

UNIX Network Programming

Tom Dunigan *dunigan@cs.utk.edu*

TA: John Randle *randle@cs.utk.edu*

Tue/Thurs 3:40-4:55

Eunuchs Networking

supportive system of sharing
information among male sopranos

Today

- overview
- class mechanics
- networking history

Objectives

- understand BSD socket calls
- understand IP protocols
- examine client/servers
  "how do they do that"
- develop client/servers
  design
  implement
  diagnose
- understand design tradeoffs
- develop portable tools

Concepts

- **client/servers**
  concurrent/iterative
  statefull/stateless
- **concurrency**
  multiple processes/threads
  asynchronous events (signal)
  asynchronous I/O
- **communications**
  stream/datagram (circuits/packets)
  reliable/unreliable
  performance

Syllabus

- BSD sockets, TCP/UDP client-servers
- IP protocols
- UNIX servers
- IP security
- multicast, RPC
- IPv6, hi-speed

tools: commands and library

Why are you taking this course?

- thought real professor was teaching it
- want to be rich and famous
- networking is *hot*
- don't want to be a road-kill on the information highway
- thought TCP was an hallucinogen
- need credit hours

Tom's Timeline

1876 telephone invented

1878 first report of teenagers kicked off phone system for pranks

1950's/60's single user then batch computing

'64 teletype/acoustic coupler

'67 DEC10 timesharing, IBM RJE

'69 ARPANET (email)

'70 DEC 11 / UNIX

'72 lab computer interconnects

'76 Ethernet

'77 Apple II / uucp/USENET DECnet

'79 VAX / BSD UNIX (free)

'80 MFEnet / SNA / CSnet / BITNET / MS DOS

'83 OSI begun, BSD sockets

'84 ORNL on Internet (ARPAnet/MILNET)
9.6, DNS

'85 Sun workstations (sniffers)

'86 NSFnet 56 Kbs

'88 Internet worm

'89 OSI required in govt bids, gigabit testbeds

'90 US backbone to 1.5 Mbs

'91 US backbone to 45 Mbs

'92 MBONE

'93 Mosaic/www, IPv6 specs

'95 vBNS, $50 domain name, 155 Mbs

'96 MCI backbone to 622 Mbs

'99 MCI backbone to 2.5 Gbs

IBM networking

- big iron servers, dumb remotes
- not peer-to-peer (originally)
- look like a reader/punch/3270
- SNA (7 layers)
- proprietary

DECnet

- scientific labs, PDP-11s
- point-to-point
- then Ethernet, wide-area (HEP-net)
- proprietary

others: XNS, Apollo, Prime

OSI

Open Systems Interconnection – '83

- designed network
- seven layers
- services/interfaces/protocols
- bigger addresses
- international standard (ISO) – not proprietary
- IBM, European phone companies
- in '89, all govt procurements required OSI

OSI failed

Q: *What do you get when you cross a mobster with an international standard?*

A: *Someone who makes you an offer you can't understand.*

- design by committee, 3' of paper
- flawed and complex
- initial implementation slow
- initially, connection-based
- "not invented here"
- TCP/IP was spreading (free)

TCP/IP

- ARPAnet '69
- packet-switched, survivable
- experimental, universities
- BSD 4.1 (free) included TCP/IP, sockets, '83
- non-proprietary
- loosely administered

others: Appletalk, Netware (XNS/IPX)

UNIX history

'65	Multics	
'69	unix pdp-7	
'71	unix pdp-11, 16-bit	
'73	UNIX in C	
'79	UNIX v7, 64k swap, cheap	
'78	UNIX 32V Interdata 8/32	
'79		3BSD,paging
'80		4.1BSD, prepaging clusters
'82	UNIX Sys III	Vax
'83		4.1c/4.2 BSD TCP/IP
'85	UNIX Sys V	
'86	UNIX V.3 (TLI)	4.3BSD
'88		4.3Tahoe TCP/IP+ (Posix)
'89	UNIX V.4	
'90		4.3Reno SLIP,OSI
'92	Solaris	4.4 OSI,386
'93	Linux	NetBSD FreeBSD
'96		OpenBSD

C

- positional command line args or *getopt()*
- comment functions and points of interest
- use **#define** for constant
- use **#include** for common info in multi-source file projects
- use **#ifdef** for multi-UNIX portability
- check return value from system calls
- other tools: *make, ar, ranlib, truss, gdb*
- solaris gcc *-lnsl -lsocket*

System calls

- program request of kernel services
- implemented as function calls
- returns < 0 if error

```
#include <errno.h>
...
if (read(fd,buff,lth) < 0) {
   perror("db read failed");
   exit(1);  /* give up */
}
...
if ((n=select(FD_SETSIZE, &readfds,
          NULL,NULL,NULL))<0) {
   if (errno == EINTR) continue;
   perror("select");
   exit(1);
}
```

- see *cs494/unpv12e/lib/error.c*

next time

network 101: topology, terminology

```
Readings:
   class 1 readings (see web page)
   class 2 readings (see web page)
```