12 - Lecture - Introduction to TCP/IP networking

pipe

simplest form of one-way interprocess communication

- created by pipe() system call (see "man 2 pipe")
- usually called by a process before it forks in order to setup a communication channel between child processes
- example:

```
echo | ~jae/cs3157-pub/lab4/mdb-lookup-cs3157 | grep '{}'
```

TCP/IP networking

sockets are similar to pipe, but:

- two way
- connects remote processes

5 protocol layers of TCP/IP

- protocols of the Internet
- sockets API sits between layer 4 & 5
- we'll take layer 1 - 4 as blackbox (learn them in W4119)

IP address

- 32-bit integer
- usually written in dotted-quad notation: "128.59.0.5"
- DNS translates hostnames, such as "tokyo.clic.cs.columbia.edu", into an IP address

port number

- need to distinguish between many network apps in a host
- unsigned 2-byte int: 1 - 65535 (0 is reserved)
- fixed # for well-known apps (ex. 80 for web servers)

client-server model

- server "listens" on a known port
- client "connects" to it
- from that point on, 2-way communication

netcat: TCP/IP swiss army knife

- see "man nc"

- connects stdin/stdout with a socket

- server mode: nc -l <port>
  client mode: nc <hostname or IP addr> <port>

- can we turn our mdb-lookup-cs3157 into a network server
  using nc and some pipes?

  - yes, but we'll need a "named pipe" (see "man mkfifo")