Sockets and IPC

- Sockets provide endpoints for communication.
- Socket Creation:

```
sockid = socket( domain, type, protocol);
```

- Before it can be used, it needs an address.

```
status = bind( sockid, address, addrlen);
```

- Exactly one address per socket
- Exactly one socket per address
- Right address format for the domain.

Socket Domains

- Specifies the communication domain or address family

– PF_UNIX, PF_LOCAL	Local communication
– PF_INET	IPv4 Internet protocols
– PF_INET6	IPv6 Internet protocols

PF_UNIX Domain

- Addressed by a path name.
- Results in an inode being allocated which needs to be unlinked later.
- Only provides a name for the socket – the regular file system is not involved.

PF_INET

- In this domain, addresses take the form of an IP address and a port.

```
struct sockaddr_in {
   short in_family;
   u_short  sin_port;
   struct  in_addr sin_addr;
   char sin_zero[8];
}
```

Internet Domain Addresses

- The IP number specifies the machine:
  192.168.97.103
  (4 bytes)
- The port number is like a mail box.
- There are standard port numbers such as 79 for finger, 513 for remote login, (see /etc/services)
- 1 – 1024 reserved for use by the kernel
- If a port number of 0 is specified, the system will assign an unused number.

Socket Types

- The type specifies the communication semantics.
  - SOCK_STREAM sequenced, reliable, two-way, connection-based byte streams.
  - SOCK_DGRAM datagrams (connectionless, unreliable messages of a fixed maximum length).
  - SOCK_SEQPACKET Provides a sequenced, reliable, two-way connection-based on data-grams

Protocols

- Protocols are conventions governing the exchange of information.
- Two common protocols in use:
  - UDP = user datagram protocol
  - TCP = transmission control protocol

- Actual implementations of the socket types.
- Implemented in the kernel

Establishing Connections (Streams)

-
- connect()

  status = connect(sock, address, namelen)
- Use an address for the right domain ...