Signal and Event-Based Programming

Signal Handling

- Handling exceptional behaviors and asynchronous events
- Software-generated interrupts
- Indicating various events

Event-driven

- Relies on external events
- Logic flow of the program is controlled by events
- Events are triggers or actions
- Most programs are EDP

Types of IPC (Inter Process Communication)

- Signals
  - Sending, Handling, Blocking
- Pipes
- FIFOS
- Message Queues
- Semaphores
- Shared Memory

Common Signal Identifiers

- SIGHUP : Hangup detected terminal or death of process
- SIGINT : Interrupt from keyboard
- SIGQUIT : Quit from keyboard
- SIGILL : Illegal Instruction
- SIGABRT : Abort signal from abort(3)
- SIGFPE : Floating point exception
- SIGKILL : Kill signal

Common Signal Identifiers

- SIGSEGV : Invalid memory reference
- SIGALRM : Timer signal from alarm(2)
- SIGTERM : Termination signal
- SIGCHLD : Child stopped or terminated
- SIGCONT : Continue if stopped
- SIGSTOP : Stop process
- SIGTSTP : Stop typed at tty

Listing Processes

State Codes	Description
D	uninterruptible sleep (usually IO)
R	running or runnable (on run queue)
S	interruptible sleep (waiting for an event to complete)
T	stopped by job control signal
t	stopped by debugger during the tracing
W	paging (not valid since the 2.6.xx kernel)
X	dead (should never be seen)
Z	defunct ("zombie") process, terminated but not reaped by its parent

- Command `ps -u`

Sockets

- Used for client and server interaction
- Configuration places the server on one machine
- Clients on other machines

Typical Flow

- Server establishes (binds) an address, clients use
- Server waits for clients to request a service
- Client connects to the server through a socket
- Server performs request and sends the reply back

Type of Protocol

- TCP. Transmission Control Protocol
- IP. Internet Protocol
- UDP. User Datagram Protocol
- FTP. File Transfer Protocol
- SSH. Secure Shell
- SSL. Secure Sockets Layer
- SMTP. Simple Mail Transfer Protocol
- POP3. Post Office Protocol

Type of Protocol

- IMAP4. Internet Message Access Protocol
- DNS. Domain Name System
- ARP. Address Resolution Protocol
- DHCP. Dynamic Host Configuration Protocol
- HTTP. Hypertext TP
- HTTPS. Hypertext TP Secure
- Ethernet.
- SIP. Session Initiation Protocol

Raw Socket - depends on the interface provided

- Direct access to network layer
- Bypass protocol stack
- Special privileges, Not intended for general user

Sequenced Packet Sockets

- Reliable, Ordered Delivery, Connection-Oriented
- FLow Control and Congestion Control
- Easy for Application to Use

Stream Sockets - based on TCP

- Reliable, Ordered Delivery, Connection-Oriented
- FLow Control and Congestion Control
- Applications require reliable and continuous data

Datagram Sockets - use UDP

- Connectionless, Unreliable Delivery
- Application handles packets
- Efficiency over reliability

Little Endian

- low-order byte is stored on the starting address A and high-order byte is stored on the next address A + 1.

Big Endian

- high-order byte is stored on the starting address A and low-order byte is stored on the next address A + 1.

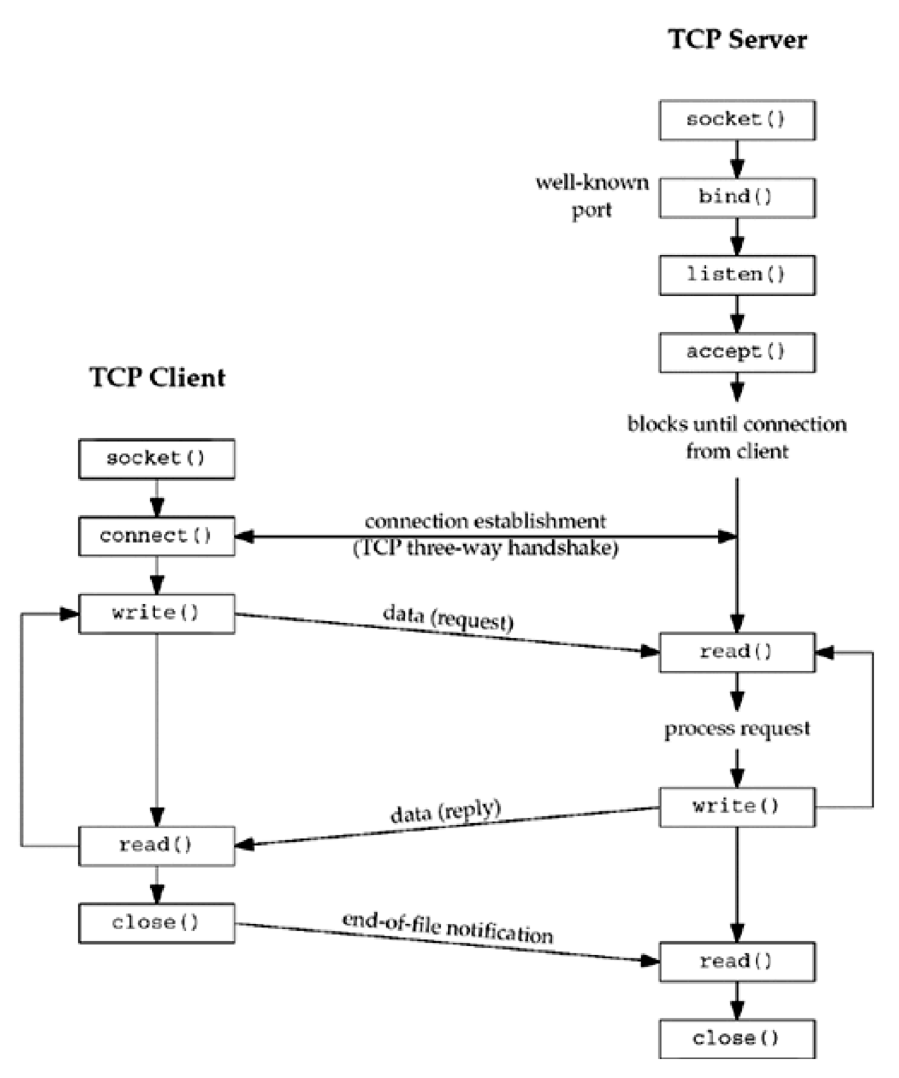
"Client" Workflow

- socket() => connect() => recv()

"Server" Workflow

- socket() => bind() => listen() => accept()

Socket Diagram



Questions?