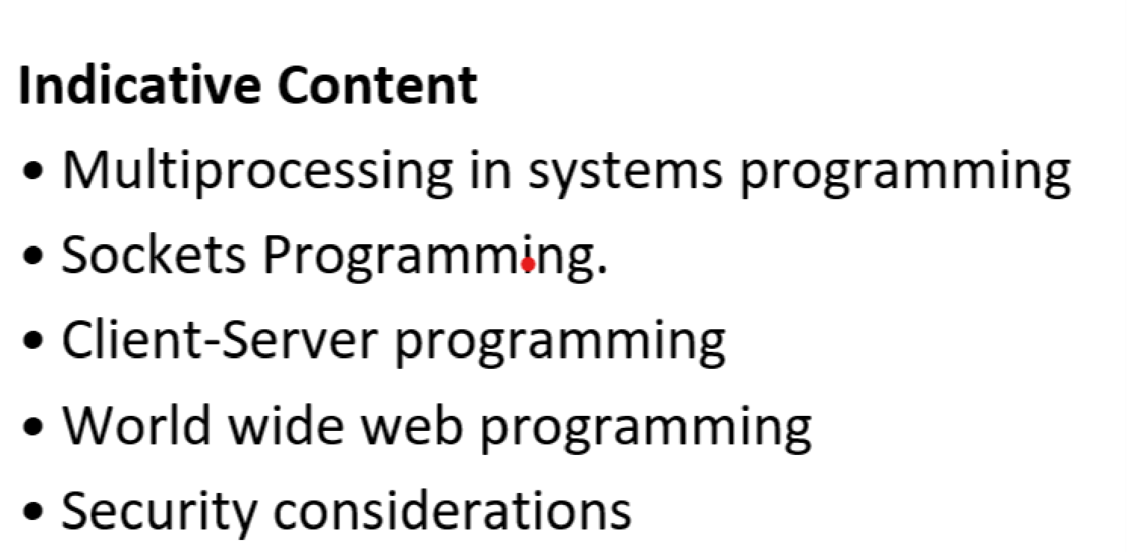
# MCN 8104 - Network Programming

Odongo Steven Eyobu, Ph.D

sodongo@cis.mak.ac.ug



**Indicative Content**

- Multiprocessing in systems programming
- Sockets Programming.
- Client-Server programming
- World wide web programming
- Security considerations

# Organization

- Assessments
  - Course work (Assignments, tests/project, class discussions, participation and attendance) - 40%
  - Final exam/ Project (60%)

# At the end….

You should know…

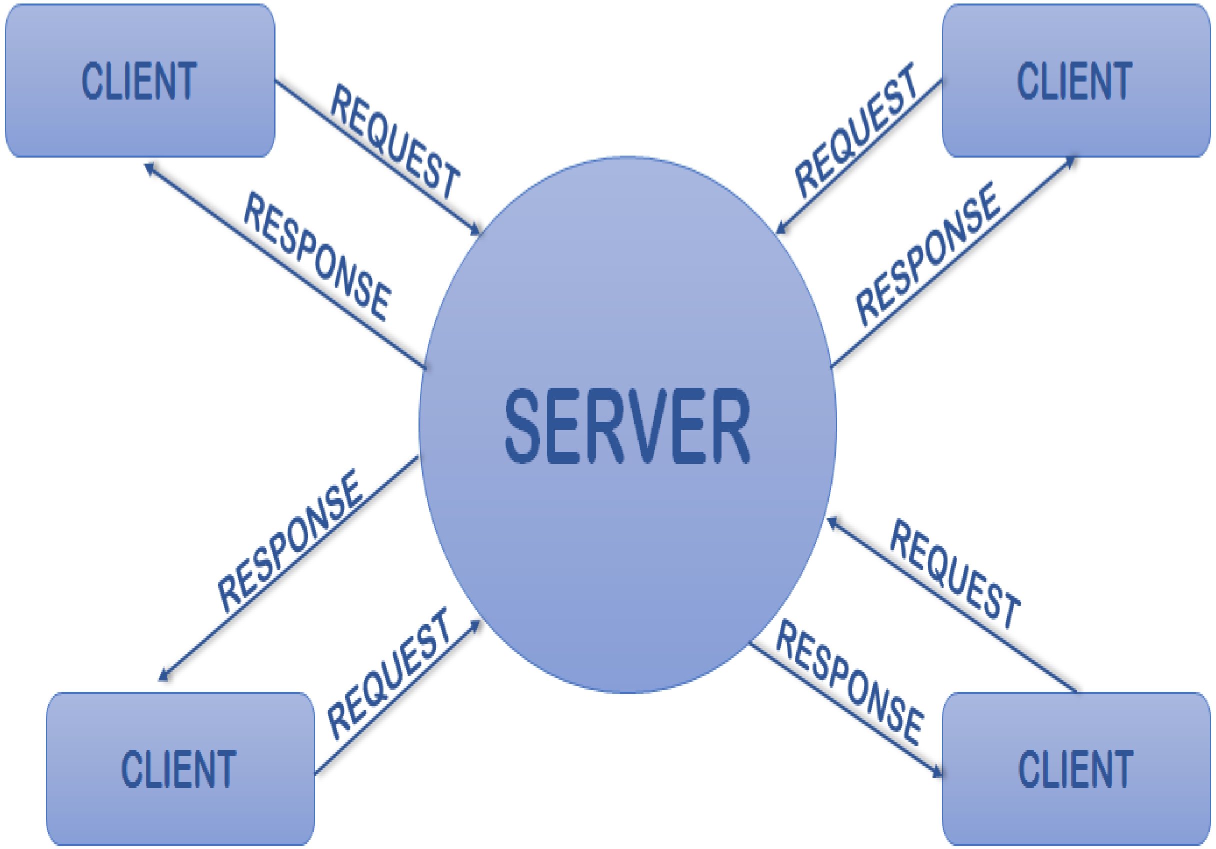
- What a socket is
- What you can do with a socket
- The difference between TCP/IP, UDP/IP
- How servers and clients communicate over sockets
- How to create a simple server
- How to create a simple client
- How to create a multithreaded server

# What is network programming?

- Network Programming involves writing programs that communicate with other programs across a computer network.

# Client/Server Network Architecture

# Client Server Network Architecture



A server hosts, delivers and manages most of the resources and services to be consumed by the **client**

A client is the receiving end of a service or the requestor of a service in a client/server model type of system. The client is most often located on another system or computer, which can be accessed via a network.

# Client-Server Networking Architecture

- Server
  - Application
  - Hardware
- Client
  - Application
  - Hardware

- Server types
  - Webservers
  - Mail servers
  - Fileservers
  - Etc…
- Client types
  - Thick
  - Thin
  - Hybrid (Thick &Thin)

# Client/Server Communication

- Three things must be known
  - Address
    - IP addresses
    - www.mak.ac.ug
  - Port
    - 0 - 65535 ~ 2**16
      - Defined ports, Reserved ports, well known ports
  - Protocol
    - Set of rules that the server and client agree to follow

# Client/Server Connections

- Transport layer protocols
  - TCP – Connection-oriented
  - UDP – Connectionless-oriented

# TCP vs UDP

- TCP : Transmission Control Protocol : Is a connection- based protocol that provides a reliable flow of data between two computers.
  - Used when two applications want to communicate reliably
  - It is Connection Based
  - Data is get in the same order it was sent (via Streams)
  - Transmission guarantied, or error is reported.
  - Example:
    - HTTP, FTP, SMTP, TELNET,

- UDP : User Datagram Protocol : Is a protocol that sends independent packets of data, called datagrams, from one computer to another with no guarantees about arrival.
  - Not connection based
  - Communication is not guaranteed
  - Datagram : A packet sent by UDP protocol.
  - The order of datagrams are not guaranteed.
  - Example:
    - Radio, Clock Server, Ping,

# Client server Connections

- Socket object
- Multiple socket objects ~ multiple connections
  - Every connection is identified by
    - Source address, source port #, Dest address, dest port #, protocol.

How many connections could your PC be having at a given time?

# Netstat

- **Stands for**: Network statistics

- **Function**: Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

- **Syntax**: netstat [address_family_options] [--tcp|-t] [--udp|-u] [--raw|-w] [--listening|-l] [--all|-a] [--numeric|-n] [--numeric-hosts][--numeric-ports][--numeric-ports] [--symbolic|-N] [--extend|-e[--extend|-e]] [--timers|-o] [--program|-p] [--verbose|-v] [--continuous|-c] [delay]

# Local host address / Loop back interface

- You don’t need internet or any network connectivity to successfully study this course.
- Client and server can be programmed to run on one machine.
- 127.0.0.1
- Loopback --- A connection to yourself

- Ping 127.0.0.1
- Ping 127.2.4.54

# Tools

- To familiarize yourself with
- Netstat
- Ping
- Telnet – RCP that enables you to connect to a server
  - $ telnet google.com 80
  - Connects to the google server at port 80

# Sockets API

- API
  - Blackbox with functions
  - Needs arguments to make use of the functions.
- Socket API's
  - A bunch of functions that we use to write clients and servers
    - Socket()
    - Bind() – associates a socket with a network interface
    - Listen()
    - Etc….

# Assignment 1

- Develop a simple webserver using python
  - Chapter 2, Socket Programming Assignment 1

- Write a client ping program using python
  - Chapter 2, Socket Programming Assignment 2

Reference text is **Computer Networking, A Top Down Approach. By KUROSE and ROSS.** Sixth Edition**.**

# Lecture 2.
# Basics of Communication

Moral: Write code for an environment we have knowledge about to give us better intuition.

# Basics of Communication

- Sender Receiver Model
  - Tx and Rx
  - Channel
  - Noise
  - Signal
  - Message
  - Encoding
  - Decoding
- The OSI model
  - Metadata at @ layer
    - Error detection
      - Even parity
      - Odd parity

- Types of Communication networks
  - Circuit switching networks
    - Dedicated links
  - Message switching
  - Packet switching networks
    - Adv.
      - Solves limited channel constraints
    - Disadv
      - More complexity
      - Order of reception
      - Duplicate packets
      - loops

- Topology view
  - Bus, ring, star,mesh
  - Advs
  - Disadvs

Ref. Chap. 1 &2

**Computer Networking, A Top Down Approach. By KUROSE and ROSS.**

# Assignment 2

- Develop a mail client using python
    - See detail in Chapter 2, Socket Programming Assignment 3

Reference text is **Computer Networking, A Top Down Approach. By KUROSE and ROSS.** Sixth Edition**.**