Outlines:

Network Architecture and Networking software

Mainframe Architecture
Client/server Architecture
Different Client/server models
File server Architecture
Upsizing
Downsizing
What does a client expect from as OS?
Analysis of client OS trends
Analysis of NT, UNIX clients
Analysis of server OS trends
Analysis of NT, UNIX clients

Mainframe Architecture

- Architecture is a set of defined terms and rules that are used as instructions to build products.
- Each generation of mainframe computers has included improvements in architecture, while remaining the most stable, secure, and compatible of all computing platforms.
- In computer science, architecture describes the organizational structure of a system.
- Architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components, and subsystems.
    - More and faster processors
    - More physical memory and greater memory addressing capability
    - Dynamic capabilities for upgrading both hardware and software
    - Increased automation of hardware error checking and recovery
    - Enhanced devices for input/output (I/O) and more and faster paths (channels) between I/O devices and processors
    - More sophisticated I/O attachments, such as LAN adapters with extensive inboard processing
    - A greater ability to divide the resources of one machine into multiple, logically independent and isolated systems, each running its own operating system
    - Advanced clustering technologies, such as Parallel Sysplex®, and the ability to share data among multiple systems.
- Despite the continual change, mainframe computers remain the most stable, secure, and compatible of all computing platforms.
- The latest models can handle the most advanced and demanding customer workloads, yet continue to run applications that were written in the 1970s or earlier.
- How can a technology change so much, yet remain so stable? It can by evolving to meet new challenges.
- In the early 1990s, the client/server model of computing, with its distributed nodes of less powerful computers, emerged to challenge the dominance of mainframe computers.

- Industry pundits predicted a swift end for the mainframe computer and called it a "dinosaur."
- In response, mainframe designers did what they have always done when confronted with changing times and a growing list of user requirements: They designed new mainframe computers to meet the demand.
- With the expanded functions and added tiers of data processing capabilities such as Web-serving, autonomics, disaster recovery, and grid computing, the mainframe computer is poised to ride the next wave of growth in the IT industry.

Reference:
http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp?topic=/com.ibm.zos.zmainframe/zconc_evolvarch.htm

File Server architecture

- development of microprocessor, PC and LAN enable "smart" clients and dumb server
- clients responsible for user interface, applications, file-based data management
- file server only services read and write requests -- applications run on the client
- Advantages
  - Flexible, inexpensive to deploy
- Disadvantage
  - low security and reliability

Client/Server Architecture

- It is an architecture in which a system's functionality and its processing are divided between the client PC (the front end) and databases server (the backend server).
- The system functionality, such as programming logic, business rules, programming logic, business rules, and data management is segregated between the client and server machine.
- Characteristics
  - Service
    - It is primarily a relationship between processes running on separate machine
    - The server serves the processes
    - Client consumes the service
    - Client/server should provide a client separation of functions based on the idea of service.
  - Shared resources
    - A server can serve many clients at the same time and regulate their access to share resources.
  - Asymmetrical protocols
    - Many to 1 relationship between client and server
    - Client always initiate the dialog by requesting a service.
    - Servers are passively awaiting requests from the clients
    - In some cases client passes a reference to a callback object when it invokes a service. This lets the server call back to client, thus client behave like server.
  - Transparency of location.
    - Server is a process that can reside on same machine or another.
    - Client server software usually makes the location of the server from the clients by redirecting the service calls when needed.
  - Mix and match
    - The ideal client/serer software is independent of hardware and OS platform.

    - We should be able to mix and match client and server platform.
- Message based exchanges
    - Clients and servers are loosely coupled systems that interact through a message passing mechanism
    - The message is the delivery mechanism for the service requests and replies.
- Encapsulation of services
    - The server is a "specialist".
    - A message tells a server what service is requested.
    - It is then up to the server to determine how to get the job done.
    - Server can be upgraded without affecting the clients as long as published message interface is not changed.
- Scalability
    - It can be scaled horizontally or vertically.
    - Horizental scaling means adding or removing client workstations with only a slight performance impact.
    - Vertical scaling means either migrating to a larger and faster server machine or distributed the processing load across multiple servers.
- Integrity
    - The serve code and server data is centrally managed, which results in cheaper maintenance and the guarding of shared data integrity.
    - At the same time, the client remains personal and independent.

Client/Server Models

- Client/server system can be classified on the way in which the systems have been built.
- The most widely accepted range of classifications has come from Gartner group, a market research firm in Stanford.
- The system we are using may slightly differ in terms of design, these models give us a good idea of how client/server systems can be built.
- Each model are not mutually exclusive to each other.
- More than one model can be combined to produce effective and efficient system.
- Client server system can change the model on upgration of application software.
- These models demonstrate the full definition of a client/server system in which a client issues requests and servers works done by one or more servers.
- Gartner classification
    - Model 1: Distributed Presentation
        - Both client server machine format the display presented to the end user.
        - The client machine intercepts display output from server intended fro a display device and reroutes the output throuth its own processes before presenting it to the user.
        - It provides terminal emulation on the client along with other application.
        - Easy to implement using software like WallDate's Rumba.
        - But is may provide no real business benefit other than to begin a migration to client/server.
        - Sometime more advanced terminal emulation may be used whereby the emulation screen is hidden and copy some of its contents normally a keyfields onto a VB or Delphy screen.

    - This copying is referred to as **screen scraping**.
    - This process helps to hide mainframe or midrange servers screen and present them with under PC interface such as windows/Linus/Unix
    - The major benefits of screening are that it allows a system to migrate from an old mainframe based system to a new client/server system in small incremental steps.
- Model 2
    - Some portions of program logic are also placed on client machine.
    - Allow some business/program logics as well as presentation logic to reside on the client PC.
    - Useful in PC-LAN Environment.
    - The logic can be of many types validation fields are commonly used in client PC as logic
    - Different in this model from above is that the host does not format the data with remote presentation model.
    - The client and server process communicate through more advanced protocols such as IBM Advanced Peer to peer communication (APPC)
    - The serve sends a raw data stream to the client.
    - The client formats the data and presents it to the end user.
    - All the core system application logic still resides on the server; some validation logic may be moved or duplicated on the client.
- Model 4: Remote Data
    - Client handles all the application logic and end user presentation and the server provides only data.
    - Client typically use remote SQL or Open Database Connectivity (ODBC) to access the data stored on the server.
    - Most common today.
- Distributed Data
    - Data distributed access multiple network systems
    - Data source may be distributed between the client and the server or multiple servers.
    - It requires and advanced data management scheme to enforce data concurrency, security and integrity across multiple.
    - It is the most difficult model to use.
    - It is complex and requires a great deal of planning decision making to used effectively.

Moving to Client/Server

- Some companies are downsizing their system from mainframes.
- Some companies are upsizing their system from PC to client/server system.
- Whether upsizing or downsizing to client/server systems, companies will encounter a tremendous amount of change, bother technical and cultural.
- Downsizing
    - Most system downsizing initiatives focus on the replacement of mainframe system with a more flexible and cost effective technical platform.
    - Companies have decided roles that their mainframe will play in their future.
    - Companies that downsizing have decided that the mainframe did a grate job during its time but that client/server system should replace it.
    - The three main issues for downsizing are:
        - The mainframes inability to support ad hoc queries.
        - Inability to quickly modify mainframe programs

- The high maintenance and support cost for keeping a mainframe running.

Ingregraing mainframe and client/server architecture.

- Companies that have mainframes are not eliminating them in favor of client/server systems. Instead, there are integrating the two technologies and taking advantage of both their strengths.
- These companies want to leverage their investments in mainframe application, database, hardware, infrastructure and operational procedures.
- They also realize that certain processes are better handled by a client/server system.
- Company decides on a project to project to project basic which environment to develop the application in according to internal criteria. For, e.g. the system that are batch processes and do not require user interaction may be written from the mainframe; the system that require a great deal of end user integration are developed on client/server.
- Some time these two technologies are so tightly integrated that the mainframe acts as the database server in the client/server system.

What Dose a client expects from an OS?

- Each of the three types of client expects different set of requirements on the OS.

Requirements from an OS	Non-GUI client		GUI client	OOUI client
	With out multitasking	With multitasking		
Request/reply mechanism	Yes	Yes	Yes	Yes
File transfer mechanism to move pictures, text and database snapshot	Yes	Yes	Yes	Yes
Preemptive multitasking	No	Yes	Desirable	Yes
Task priorities	No	Yes	Desirable	Yes
Interprocess communications	No	Yes	Desirable	Yes
Threads for background communications with server and receiving callbacks from servers	No	Yes	Yes	Yes
OS robustness, including intertask protection	No	Yes	Desirable	Yes

Extra:

OS services (in old syllabus)

- In distributed computing environments the OS functions are either base or extended services.
- Base services are part of the standard OS.
- The extended services are add-on module software components that are layered on top of the based services
- extended services provides the advanced system software that exploits the distributed potential of networks.
- Functionally equivalent extended services are usually provided by more than one vendor.
- List of **base services**
  - Task Preemption (Multi tasking)
  - Task priority

- Semaphores (Process management)
- Interprocess communication
- Threads
- Intertask protection
- Multiuser High performance file system
- Effective memory management

- List of **extended services**
  - Network OS extensions
  - Binary Large Objects support in OS and network
  - Authentication and authorization services
  - Server management
  - Database and Transaction services
  - Internet

Assignment:

- What are different base services provided by OS explain? (Write for 6 Marks)
- What are different extended services provided by OS, explain? (Write for 6 Marks)
- Describe the structure of Server program.
  (Hints
  connection oriented Vs connection less server
  statefull vs stateless server
  Iterative Vs concurrent server
)

Server Scalability (in old syllabus)

- The limits really depends on the type of services required by clients
- One safe rule is that clients will always want more services so scalable services are frequently an issue.
- Different levels of servers
  PC server → Asymmetric multiprocessing server → symmetric multiprocessing server → Multi-server clusters.
- It starts with a single PC server that reaches it's limits with the top of the line processor and I/O power.
- If that is not enough power, the client server model allows to divide the work among different servers.
- If not enough super-servers are used, these multi processing super servers know no upper limit to power but they must know how to work together with other servers.
- Asymmetric multiprocessing users a master processor as the master processor to do all the jobs and other processors will be controlling specific services like I/O.
- Symmetric multiprocessing server can use all it's processor as main processor.
- Clusters multi servers are used in environments that require more processing power than that powered by a single server system.
- Either SMP or uniprocessor servers are used.
- client/server model is upwardly scalable
- When more processing power is needed we can add more servers thus creating a pool of servers.
- If not enough the server can change with more powerful servers.

Clients

- client/server application are client centric
- client side provides the look and feel for the services a system provides.
- Different classes of clients are
    - Non GUI clients
    - GUI clients
    - OOUI clients

Non GUI clients

- It generate server requests with a minimal amount of human interaction.
- It falls into two sub categories
    - Non GUI clients that do not need multitasking, e.g.:
        - ATM
        - bar code readers
        - cellular phone
        - fax machine
    - Non GUI clients that need multitasking
        - robots
        - testers

daemon programs

GUI clients

- Simple GUI clients are application where occasional request to the server result from a human interacting with a GUI.
- Simple GUI interface is a good fit for mainstream, OLTP-type business application with repetitive task and high volumes.
- They also make good front end clients for database servers.
- It is a graphical renditions of the dialogs that previously ran on dumb terminals
- Consist of
    - dialogs
    - color
    - menu
    - bar
    - scroll boxes
    - pull down menu
    - pop-up windows
- It uses the object action model where users can select objects and then select the actions to performed on the chosen objects.
- Most dialogs are serial in nature.
- Example: Windows 3.x and simple form based web pages.

Object oriented User Interfaces (OOUI) clients

- It provides a highly iconic, object oriented user interface that lets us directly manipulate objects on a screen typically by drag and drop.
- They are used by information workers doing multiple variable tasks whose sequence cannot be predicted.
- Example, executive and decisions support system.
- Examples of OOUIs
    - OS/2 workplace shell
    - nextStep (now Mac OS X)
    - Mac OS

- to some extent windows 98
- OO like web interface with Java 2, java beans and Ajax
- Linux

- The desktop can contain multiple workplaces running concurrently

GUI	OOUI
• A graphic application consist of an icon, a primary window with a menu bar, and one or more secondary windows	• A graphic application consists of a collection of cooperating user objects. Every thing you see is an object.
• The focus is on the main task	• Each object is represented by an icon and has at least one view.
• Ancillary task are supported by secondary windows and pop-ups	• Objects can be reused in many tasks
• Users must follow the rigid task structure	• The application boundaries are fuzzy
• An application represents a task	• The user defines what's an application by assembling a collection of objects
• Icon represent a running application	• Icon represent objects that my be directly manipulated.
• Users start application before selecting an object to work with.	• Users open the object on the desktop, which causes a windows view of the object to be displayed.