Cognizant

Passion for making a difference

Handout: ESC

Version: ESC/Handout/0408/1.0

Cognizant
500 Glen Pointe Center West
Teaneck, NJ 07666
Ph: 201-801-0233
www.cognizant.com

TABLE OF CONTENTS

Introduction

About this Module

This module is a basic information and instructional handout on ESC

Target Audience

This module is specifically intended towards the Entry Level Trainees

Module Objectives

The module will teach the student on the following areas:

- Hardware Organization
- How MVS handles the Storage?
- Input and Output
- Subsystems
- Emulator
- Dataset Organization
- Application development
- Application Development Canvas
- Legacy to web transformation

Session 01: Introduction to Enterprise System

Learning Objectives

After completing this session, you will be able to:

- Define Mainframe
- Describe the History of Mainframe
- Explain the importance of Mainframe

What is Mainframe?

A mainframe is a continually evolving general purpose computing platform incorporating in it architectural definition the essential functionality required by its target applications. Mainframes are larger and expensive computers. Mostly used by government institutions and large companies for mission critical applications. These typically include bulk data processing such as censuses, industry/consumer statistics, ERP, and financial transaction processing.

Nowadays, Mainframes have been combined with distributed architectures to provide massive storage and to improve system security, flexibility, scalability, and reusability.

MVS (Multiple Virtual Storage) is the commonly used operating system on the IBM mainframe computers. Subsystems are software products to perform specialized functions.

The essential characteristics of a mainframe are: rapid and continuing evolution, general purpose orientation, hardware implemented solutions, and the criticality of user input to all of these processes.

Mainframes combine three important features:

Maximum reliable single-thread performance: Some processes, such as the merge phase of a sort/merge (sorting can be subdivided...) MUST be run single thread. Other operations (balancing b-trees, etc) are single thread and tend to lock out other accesses. Therefore, single thread performance is critical to reasonable operations against a Database (especially when adding new rows).

Maximum I/O Connectivity: Mainframes excel at providing a convenient paradigm for HUGE disk farms; While SAN devices kind of weaken this to some degree, SAN devices mimic the model of the Mainframe in connectivity "tricks" (at least internally).

Maximum I/O Bandwidth: Despite the huge quantities of drives that may be attached to a mainframe, the drives are connected in such a way that there are very few choke-points in moving data to/from the actual processor complex.

How Mainframe is different from PCs?

All system architectures are best at different jobs; each is a set of compromises. Mainframes are more expensive because the compromises are less, well, compromised. The CPU performance is not always greater (in MIPS) than other processes, but the actual priority here is not raw performance but reliability. Mainframes, due to their great cost (and trouble in amortizing this across outages) often allow for "graceful degradation" and servicing while the system is running. While this is not a universal trait, it's interesting to see this priority setting the line in the sand between performance and price.

Mainframes vs. supercomputers

Both types of systems offer parallel processing. Supercomputers typically expose it to the programmer in complex manners, while mainframes typically use it to run multiple tasks. One result of this difference is that adding processors to a mainframe often speeds up the entire workload transparently.

Supercomputers are optimized for complicated computations that take place largely in memory, while mainframes are optimized for comparatively simple computations involving huge amounts of external data. For example, weather forecasting is suited to supercomputers and insurance business or payroll processing applications are more suited to mainframes.

Supercomputers are often purpose-built for one or a very few specific institutional tasks (e.g. simulation and modeling). Mainframes typically handle a wider variety of tasks (e.g. data processing, warehousing). Consequently, most supercomputers can be one-off designs, whereas mainframes typically form part of a manufacturer's standard model lineup.

Mainframes tend to have numerous ancillary service processors assisting their main central processors (for cryptographic support, I/O handling, monitoring, memory handling, etc.) so that the actual "processor count" is much higher than would otherwise be obvious. Supercomputer design tends not to include as many service processors since they don't appreciably add to raw number-crunching power.

Brief History of Mainframes

Several manufacturers produced mainframe computers from the late 1950s through the 1970s. The group of manufacturers was first known as "IBM and the Seven Dwarfs": IBM, Burroughs, UNIVAC, Control Data, Honeywell, General Electric and RCA.

Later, shrinking, it was referred to as IBM and the BUNCH. IBM's dominance grew out of their 700/7000 series and, later, the development of the 360 series mainframes. The latter architecture has continued to evolve into their current zSeries/z9 mainframes which, along with the then Burroughs and now Unisys.

MCP-based mainframes, are among the few mainframe architectures still extant that can trace their roots to this early period.

That said, while they can still run 24-bit System/360 code, the 64-bit zSeries and System z9 CMOS servers have nothing physically in common with the older systems. Notable manufacturers outside the USA were Siemens and Telefunken in Germany, ICL in the United Kingdom, and Fujitsu, Hitachi, Oki, and NEC in Japan.

The Soviet Union and Warsaw Pact countries manufactured close copies of IBM mainframes during the Cold War; the Strela is an example of an independently designed Soviet computer.

Companies found that servers based on microcomputer designs could be deployed at a fraction of the acquisition price and offer local users much greater control over their own systems given the IT policies and practices at that time.

Terminals used for interacting with mainframe systems were gradually replaced by personal computers.Consequently, demand plummeted and new mainframe installations were restricted mainly to financial services and government.

In the early 1990s, there was a consensus among industry analysts that the mainframe was a dying market as mainframe platforms were increasingly replaced by personal computer networks.

The growth of e-business also dramatically increased the number of back-end transactions processed by mainframe software as well as the size and throughput of databases.

Another factor currently increasing mainframe use is the development of the Linux operating system, which can run on many mainframe systems, typically in virtual machines. Linux allows users to take advantage of open source software combined with mainframe hardware RAS.

Rapid expansion and development in emerging markets, particularly China, is also spurring major mainframe investments to solve exceptionally difficult computing problems, e.g. providing unified, extremely high volume online transaction processing databases for 1 billion consumers across multiple industries (banking, insurance, credit reporting, government services, etc.)

Why Mainframe?

The CPU speed of mainframes has historically been measured in millions of instructions per second (MIPS). MIPS have been used as an easy comparative rating of the speed and capacity of mainframes.

The smallest System z9 IBM mainframes today run at about 26 MIPS and the largest about 17,801 MIPS. IBM's Parallel Sysplex technology can join up to 32 of these systems,making them behave like a single, logical computing facility of as much as about 569,632 MIPS.

The MIPS measurement has long been known to be misleading and has often been parodied as "Meaningless Indicator of Processor Speed." The complex CPU architectures of modern mainframes have reduced the relevance of MIPS ratings to the actual number of instructions executed.

Likewise, the modern "balanced performance" system designs focus both on CPU power and on I/O capacity, and virtualization capabilities make comparative measurements even more difficult.

IBM has long published a set of LSPR (Large System Performance Reference) ratio tables for mainframes that take into account different types of workloads and are a more representative measurement.

However, these comparisons are not available for non-IBM systems. It takes a fair amount of work (and maybe guesswork) for users to determine what type of workload they have and then apply only the LSPR values most relevant to them.

To give some idea of real world experience, it is typical for a single mainframe CPU to execute the equivalent of 50, 100, or even more distributed processors' worth of business activity, depending on the workloads. Merely counting processors to compare server platforms is extremely perilous.

Some universal facts about Mainframe

- The mainframes hold approximately “70% " of the entire data stored in this planet.
- The latest ones can support over 25,000 users.
- They can support devices spread over 26 miles using fiber optics.
- The size of the "basic" OS is about 14 GB.
- This OS comes in about 57 magnetic tapes, But latest with 4 tapes.
- It takes ATLEAST a month to install and customize a basic mainframe OS configuration.
- It takes around 3 full days for a mainframe to get up and running.
- Basic configuration for development environment costs 8 crores (hardware + Software)
- A full team of qualified system operators are always required to be on stand by to ensure successful operation
- Recent mainframes have variable speed fans in the cabinet to keep it cool.
- Have u ever heard of a mainframe system being hacked??? Mainframes are one of the most secure data installations ever.
- You cannot buy the mainframe OS, but you’ll have to pay a license to use it. The cost, 1.5 crores a year.
- Finally, mainframes today r the most preferred data servers for even the most hi-fi of the organizations!
- So far no virus has attacked Mainframes.
- Most of the peoples working on Mainframe had never seen the mainframe.

- About 60% of the business applications around the world are on the mainframes. For a good user interface, changing these applications' interface with available tools cost far less than developing new applications. Means, service providers like CTS get business for maintaining and converting legacy applications than getting a web development project.

Summary

- Mainframes are larger and expensive computers.
- Subsystems are software products to perform specialized functions.
- The CPU speed of mainframes has historically been measured in millions of instructions per second (MIPS).
- The complex CPU architectures of modern mainframes have reduced the relevance of MIPS ratings to the actual number of instructions executed.

Test Your Understanding

1. CPU speed of mainframes has historically been measured _____.
    - KBPS.
    - MIPS.
    - GBPS
    - None of the earlier.

2. The smallest System _____mainframes today run at about _____ and the largest about _____
    - z9 IBM, 26 MIPS, 17,801 MIPS.
    - z9 IBM, 29 MIPS, 18,701 MIPS.
    - z6 IBM, 26MIPS, 17,801 MIPS.
    - None of the earlier

Session 02: Hardware Organization

Learning Objectives

After completing this session, you will be able to:

- Describe the History of Mainframe
- Define Sysplex
- Identify Operating System

Overview

A typical mainframe installation will normally run the applications covering one of the following:

- Large business problems--ones that involve hundreds of end users, or deal with a very large volume of work that can be counted in the millions of transactions per day.
- Work that consists of small work units, such as online transactions or large work units that can be subdivided into smaller work units, such as queries.
- Concurrent applications on different systems that need to directly access and update a single database without jeopardizing data integrity and security.

All of the above characteristics of the application will demand a higher throughput or higher availability. The Sysplex architecture will help in achieving the same for the application.

What is Sysplex?

A Sysplex is a collection of MVS systems that cooperate, using certain hardware and software products, to process work.

The multiple processors (CPC - Central Processor Complex) are connected using Coupling Facility (CF) on CF channels.

Evolution of Sysplex

A Sysplex is the most recent development in the evolution of IBM large systems. Large system configurations have evolved from a single system uniprocessor through tightly coupled multiprocessors, to a loosely coupled configuration, to the Sysplex.

Single system - Uniprocessor

A system is made up of hardware products including a central processor (CP), and software products, with the primary one being an operating system such as MVS. Other types of software--system application programs, end-user application programs, tools--run on the system. The CP is the functional hardware unit that interprets and processes program instructions. The CP and other system hardware, such as channels and storage, make up a central processor complex (CPC).

The System/390 architecture defines that a single CP process one and only one instruction from a program at a time. The MVS operating system manages the instructions to be processed and the

resources required to process them. When a single copy of the MVS operating system (MVS image) manages the processing of a CPC that has a single CP, the system configuration is called a uniprocessor. The following fig 2.1 represents the above configuration.

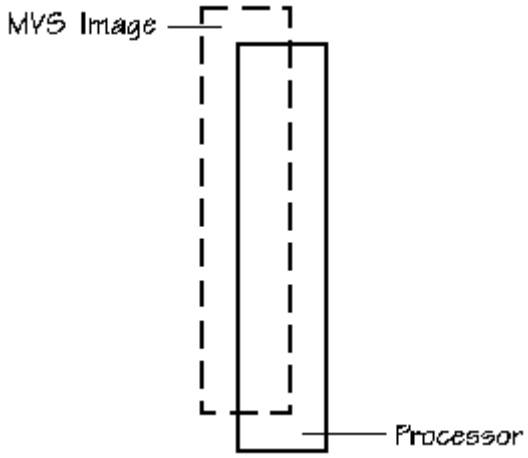


Fig 2.1 - Single System Uniprocessor

Capacity

Equal to the size of the largest single CP

Availability

Contains single points of failure

Systems Management

Easy to manage work

Tightly coupled Multiprocessor

When more CPs are added to the central processor complex (CPC), capability of processing program instructions simultaneously is added. When all the CPs share central storage and a single MVS image manages the processing, work is assigned to a CP that is available to do the work. If a CP fails, work can be routed to another CP. This hardware and software organization is called a tightly coupled multiprocessor. A tightly coupled multiprocessor has more than one CP and a single MVS image sharing central storage. The CPs are managed by the single MVS image, which assigns work to them. The following fig 2.2 represents the configuration.

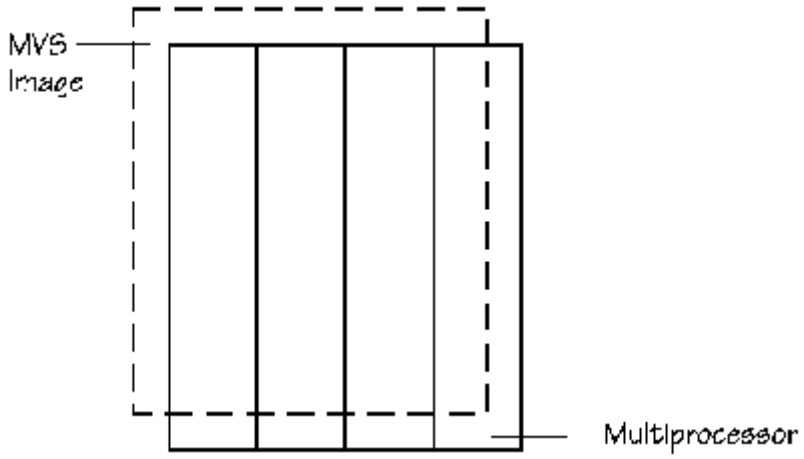


Fig 2.2 - Tightly Coupled Multiprocessor

Capacity

Increased over that of a uniprocessor but limited by the maximum number of CPs in the CPC

Availability

Increased over that of a uniprocessor but limited by some characteristics including single point of failure.

Systems Management

Easy to manage work

Loosely coupled Multiprocessor

A tightly coupled multiprocessor provides CP backup in case of failure. But software backups are not possible and only one MVS image can run in the configuration.

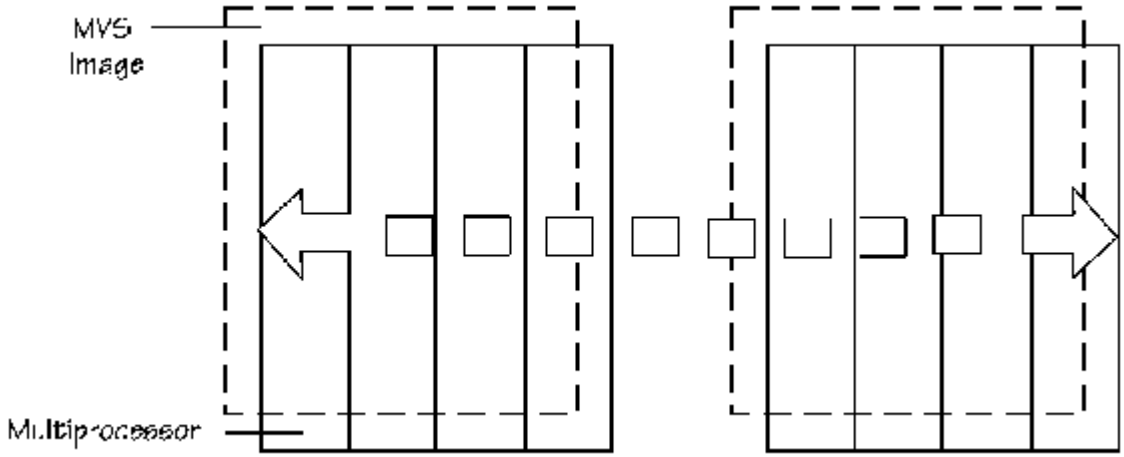
Systems outside a Sysplex can coordinate more than one MVS image with the MVS job entry subsystem components (JES2 and JES3) and global resource serialization. These components establish a means of sharing a work input queue across a number of systems to allow shared data sets, printers, and consoles. This type of system configuration is called loosely coupled.

A loosely coupled configuration has more than one CPC, possibly tightly coupled multiprocessors, sharing DASD but not central storage. The CPCs can connect by channel-to-channel communications and are managed by more than one MVS image. Work is distributed from a shared job queue to each MVS.

Although a loosely coupled configuration increases system capacity, it is not as easy to manage as either a uniprocessor or a tightly coupled multiprocessor. Each system must be managed separately, often by a human operator, who monitors product-specific messages on a set of consoles for each system.

Products and applications that need to communicate and are running on separate systems have to create their own communication mechanism. These varied communication mechanisms add to the

difficulty of managing a loosely coupled configuration. The following fig 2.3 represents the above configuration.



Capacity

Increased system capacity over tightly coupled multiprocessors and a uniprocessor

Availability

Increased system availability over tightly coupled multiprocessors and a uniprocessor

Systems Management

Requires additional systems management--separate MVS images communicate to share data sets, printers, and consoles

Base Sysplex

To help solve the difficulties of managing many MVS systems, IBM introduced the MVS ***Sys***tems com***plex*** or Sysplex in September of 1990. The base Sysplex lays the groundwork for simplified multisystem management through the cross-system coupling facility (XCF) component of MVS/ESA. XCF services allow authorized applications on one system to communicate with applications on the same system or on other systems. In a base Sysplex, CPCs connect by channel-to-channel communications and a shared dataset to support the communication. When more than one CPC is involved, a Sysplex Timer synchronizes the time on all systems.

The base sysplex is similar to a loosely coupled configuration in that more than one CPC (possibly a tightly coupled multiprocessor) shares DASD and is managed by more than one MVS image. A Sysplex is different from a loosely coupled configuration because through XCF, there is a standard communication mechanism for MVS system applications. The following fig 2.4 represents the above configuration.

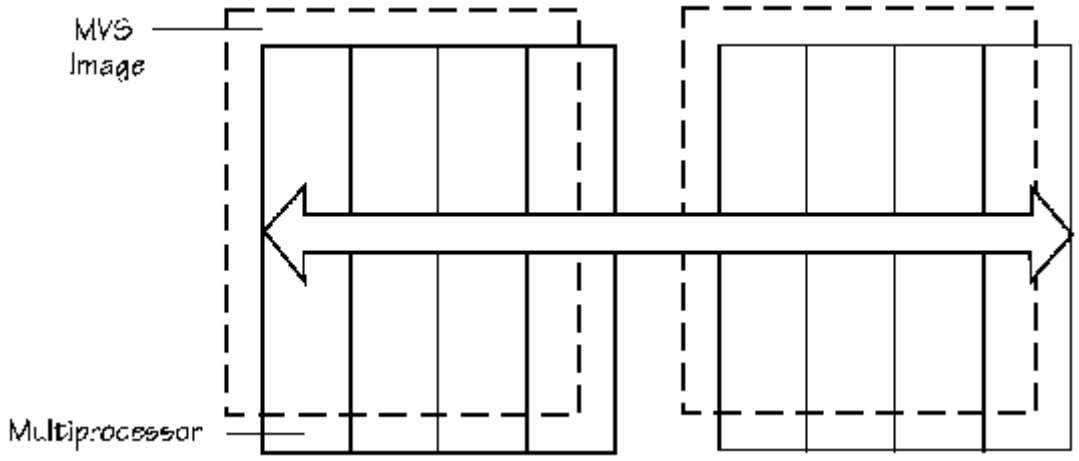


Fig 2.4 - Base Sysplex

Capacity

The same system capacity as loosely coupled

Availability

The same system availability as loosely coupled

Systems Management

- Better and simpler systems management than loosely coupled
- Greater degree of communication and cooperation among systems
- Introduction of XCF as a common communication mechanism that provides high availability
- A more unified system image--single MVS console to manage all components

Parallel Sysplex

Since the introduction of the Sysplex, IBM has developed technologies that enhance Sysplex capabilities. The Parallel Sysplex supports a greater number of systems and significantly improves communication and data sharing among those systems.

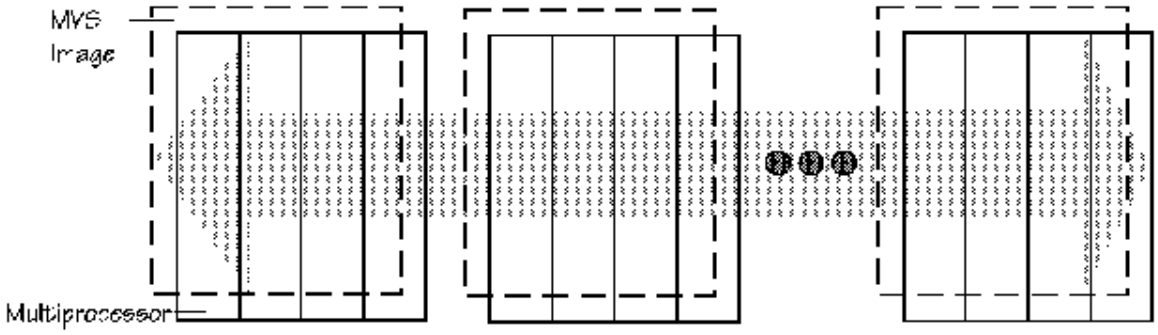
High performance communication and data sharing among a large number of MVS systems could be technically difficult. But with the Parallel Sysplex, high performance data sharing through a new coupling technology (coupling facility) gives high performance multisystem data sharing capability to authorized applications, such as MVS subsystems.

Use of the coupling facility by subsystems, such as Information Management System (IMS), ensures the integrity and consistency of data throughout the entire Sysplex.

The capability of linking together many systems and providing multisystem data sharing makes the Sysplex platform ideal for parallel processing, particularly for online transaction processing (OLTP) and decision support.

In short, a Parallel Sysplex builds on the base Sysplex capability, and allows you to increase the number of CPCs… and MVS images that can directly share work. The coupling facility enables high performance, multisystem data sharing across all the systems. In addition, workloads can be

dynamically balanced across systems with the help of new workload management functions. The following fig 2.5 represents the above configuration.



Capacity

- Increased system capacity over loosely coupled and base sysplex
- Ability to add incremental capacity to match workload growth

Availability

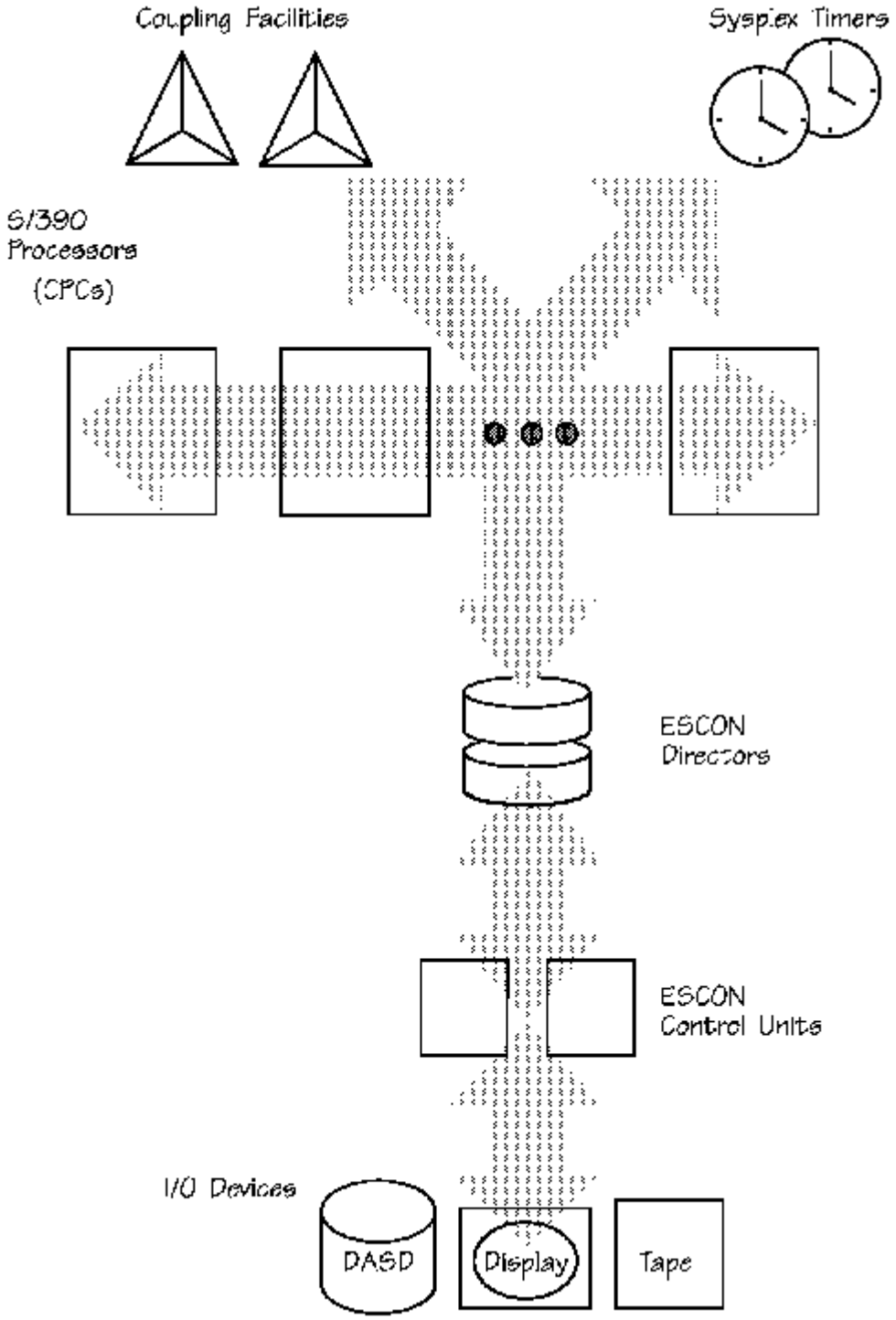
Increased system availability over loosely coupled and base sysplex

Systems Management

- Better systems management than with the base sysplex
- Multisystem data sharing capability
- Multisystem workload balancing
- Enhanced single-system image

Parallel Sysplex Architecture

A Sysplex is not a single product. Rather, a Sysplex is a collection of products, both hardware and software, that work together. The following diagram fig 2.6 is a typical Parallel Sysplex architecture.



The **Coupling facility** is a CMOS processor. It allows high performance multisystem data sharing among certain S/390 processors.

The **Sysplex Timer** is a unit that synchronizes the time-of-day (TOD) clocks in multiple CPCs in a Sysplex. The time stamp from the Sysplex Timer is a way to monitor and sequence events within the Sysplex.

ESCON channels have a channel-to-control-unit I/O interface that uses optical cables as a transmission medium. **ESCON Directors** (ESCDs) add dynamic switching capability for ESCON channels, further increasing connectivity and device sharing.

ESCON I/O and the interconnect technologies become very important in a Sysplex to speed access to shared data on disk or tape, and to enhance communication among systems. In addition, they offer improved availability.

Summary

- A Sysplex is a collection of MVS systems that cooperate, using certain hardware and software products, to process work.
- The base Sysplex lays the groundwork for simplified multisystem management through the cross-system coupling facility (XCF) component of MVS/ESA.
- The Parallel Sysplex supports a greater number of systems and significantly improves communication and data sharing among those systems.
- ESCON channels have a channel-to-control-unit I/O interface that uses optical cables as a transmission medium.

Test Your Understanding

1). When was S390 developed?
   - 1960
   - 1970
   - 1980
   - 1990

2). What is sysplex?
   - Collection of MVS systems
   - Collection of mainframes
   - collection of applications
   - collection of users

Session 03: How MVS Handles the Storage

Learning Objectives

After completing this session you will be able to:

- Explain the storage used in MVS

Primary Storage and Virtual Storage

If any Operating system has to be a multi-user Operating system, the Main memory (will be referred as Real Storage) has to be shared between multiple users. If multiple users have to use the real storage the real storage has to be partitioned so that the users will use the system without interfering with others. Partitioning the real storage will have the following consequences

- No. of concurrent users will be limited
- The size of the real storage available for the individual user will be reduced

To counter these problems the concept of Virtual Storage came into effect.

When the user waits on an I/O operation or has used up his timeslice, his program will be swapped out to the Secondary storage Device. Later when the same user is due for processing he will be swapped into the real storage. This entire process of swapping in and swapping out is totally transparent to the user. So, the user feels that he had the entire memory for the entire duration of the execution of his program, but actually he shared the real storage with all the other users. This is the basic concept of "Virtual" storage.

Multiple Virtual Storage

Address Spaces

Address space is nothing but a range of addresses a processor can access. The size of the address space will be 2^n-1 for a 'n' bit processor.

MVS provides each and every user with one Address space. (16MB for 24 bit mode and 2GB for 31 bit mode)

Swapping and Paging

When multiple address spaces reside under the MVS operating system, the real storage cannot always hold all the address spaces. When the address space is 'not in use', they are swapped out into Secondary storage device onto swap datasets.

Within a address space, the real storage will not hold the whole of the address space in the real storage. The address space will be organized as 4K blocks called pages. Individual pages are used 'on-demand'; i.e., the pages are brought into the real storage only when they are referenced. The least recently used page will be thrown back into the secondary storage. This process of getting the page into the real storage and throwing the page out into the DASD (Direct Access Storage Device) is called paging in and paging out (collectively called paging).

Addressing Mode

Addressing modes are an aspect of the instruction set architecture in most central processing unit (CPU) designs. The various addressing modes that are defined in a given instruction set architecture define how machine language instructions in that architecture identify the operand (or

operands) of each instruction. An addressing mode specifies how to calculate the effective memory address of an operand by using information held in registers and/or constants contained within a machine instruction or elsewhere.

Architecture of a Virtual Address space

The virtual address space provided for every user consists of 3 regions.

- System area
- Private area
- Common area

System area contains the system nucleus. This controls virtual storage swapping and paging. This program executes in real mode and is shared among all instances of the virtual address spaces.

Common area contains the additional components of the operating system. This area also is shared among all the instances of the Virtual address spaces but some components are pageable.

Private area this area contains the user programs, user data, OS services requested by the user application etc…

Dynamic Address Translation

Given the organization of Virtual address space it must be discussed about the mapping of the 'Virtual' addresses to real addresses. This is achieved by Dynamic Address Translation.

In the process of translation, two types of units of information are recognized--segments and pages. A segment is a block of sequential virtual addresses spanning 1M bytes and beginning at a 1M-byte boundary. A page is a block of sequential virtual addresses spanning 4K bytes and beginning at a 4K-byte boundary.

The virtual address, accordingly, is divided into three fields. Bits 1-11 are called the segment index (SX), bits 12-19 are called the page index (PX), and bits 20-31 are called the byte index (BX).

The segment table is looked up using the Segment Index (SX) and the address of the Page table will be retrieved from the contents of the segment table. Once the base location of the Page table is identified, the page table is looked up using the page index (PX). The page table will contain the information of the availability of the page in the real storage and if available in the real storage, will contain the real address of the page. The byte Index (BX) will be added to the page address to reference the real address.

Dataspace & Hiperspace

Dataspace and hiperspace are additional address spaces designated for data. The difference between dataspace and hiperspace is more on its usage and its control. Dataspace us fully controlled by the user programs and are available at real storage subjected to swapping and paging. Hiperspace are used by MVS and is available to user programs as 4K blocks. These will reside only in expanded storage and will never be brought to the main storage, hiperspaces normally are hi speed buffers which will simulate a DASD for the user applications.

VTOC

Each Disk unit is referred as a Volume. A Volume has a label, which is an unique Volume serial number, which is always stored on the 3rd record at track 0 and cylinder 0. The volume label also contains the address of the VTOC (Volume Table of Contents). VTOC contains the information about individual datasets9files) and their locations in the disk.

Catalog

To access a dataset it will be required to remember the Volume serial number and the name of the dataset. Typically, a mainframe system will have multiple volumes and it will be very difficult to remember the serial numbers of individual volumes. So, MVS maintains a catalog of all datasets available in the system. A dataset can be directly referenced from the catalog using just the dataset name. The catalog internally maintains the volume serial number of the volume in which the dataset resides.

Summary

- The multiple in MVS indicates that a separate virtual memory is maintained for each of multiple task partitions.
- The process of getting the page into the real storage and throwing the page out into the DASD (Direct Access Storage Device) is called paging in and paging out (collectively called paging).
- The virtual address space provided for every user consists of 3 regions.
    - System area
    - Private area
    - Common area

- A segment is a block of sequential virtual addresses spanning 1M bytes and beginning at a 1M-byte boundary.
- A page is a block of sequential virtual addresses spanning 4K bytes and beginning at a 4K-byte boundary.

Test Your Understanding

1). The size of the address space for a ‘n’ bit processor is:
    - 2^{n-1}
    - $2^{(n/2)-1}$
    - $2^{n}-1$
    - None of the earlier
2). What are the 2 types of units of information that are recognized in the process of a translation?
    - Segments and Tables
    - Pages and Indexes
    - Tables and Indexes
    - Segments and Pages

Session 04: Input and Output

Learning Objectives

After completing this session you will be able to:

- Identify Input and Output

Overview

The terms "input" and "output" are used to describe the transfer of data between I/O devices and main storage. An operation involving this kind of transfer is referred to as an I/O operation.

In ESA/390, as in ESA/370 and 370-XA, the facilities used to control I/O operations are collectively called the channel subsystem. (I/O devices and their control units attach to the channel subsystem.)

Mainframe Channels -- IBM hosts are connected to each other and to communication controllers through high-performance communication subsystems called mainframe channels.

A mainframe channel is an intelligent processor that manages the protocol on the communications media and controls the transfer of data to and from main central processing unit (CPU) storage.

Input/output processors (IOPs) are responsible for communicating between the host CPU and the channel. One IOP controls multiple channels. There is no relationship between the number of CPUs and the number of IOPs.

The channel relieves the mainframe CPU from the tasks associated with direct communication with I/O devices, thereby saving processing resources and allowing concurrent data processing and I/O processing.

General Concepts

Channels use one or more channel paths as the actual links between mainframes and I/O devices.

IOPs test for channel path availability, choose an available channel path, and initiate and sometimes terminate I/O operations.

There are two types of channel paths: bus and tag (also known as the Original Equipment Manufacturers' Interface (OEMI) or parallel I/O interface) and Enterprise System Connection, or ESCON (also known as the serial I/O interface).

I/O devices are directly attached to control units. Control units provide the logical capabilities necessary to operate and control I/O devices. In so doing, control units must interpret channel commands and adapt them to fit the characteristics of different I/O devices. For bus-and-tag channel paths, each channel is connected to the first control unit in a chain consisting of from one to eight control units. The chain acts as a bus, connecting each channel to a series of control units.

Control units can be housed separately, or they can be physically and logically integrated with I/O devices, channels, or even host CPUs.

A particular type of control unit will be used to control the operations of several similar I/O devices. For example, a tape control unit will be used to control the operations of several attached tape drives.

IOPs within the mainframe initiate I/O activity. Channels transfer data directly between main storage and the appropriate control unit.

Using channels, channel paths, control units, IOPs, and I/O devices, I/O activity is completely separated from mainframe execution of data processing programs. The mainframe CPU simply notifies the I/O subsystem that information needs to be transferred or received, and the channel does the rest.

The primary elements of a mainframe channel are shown in the following Fig 4.1

Host Computer

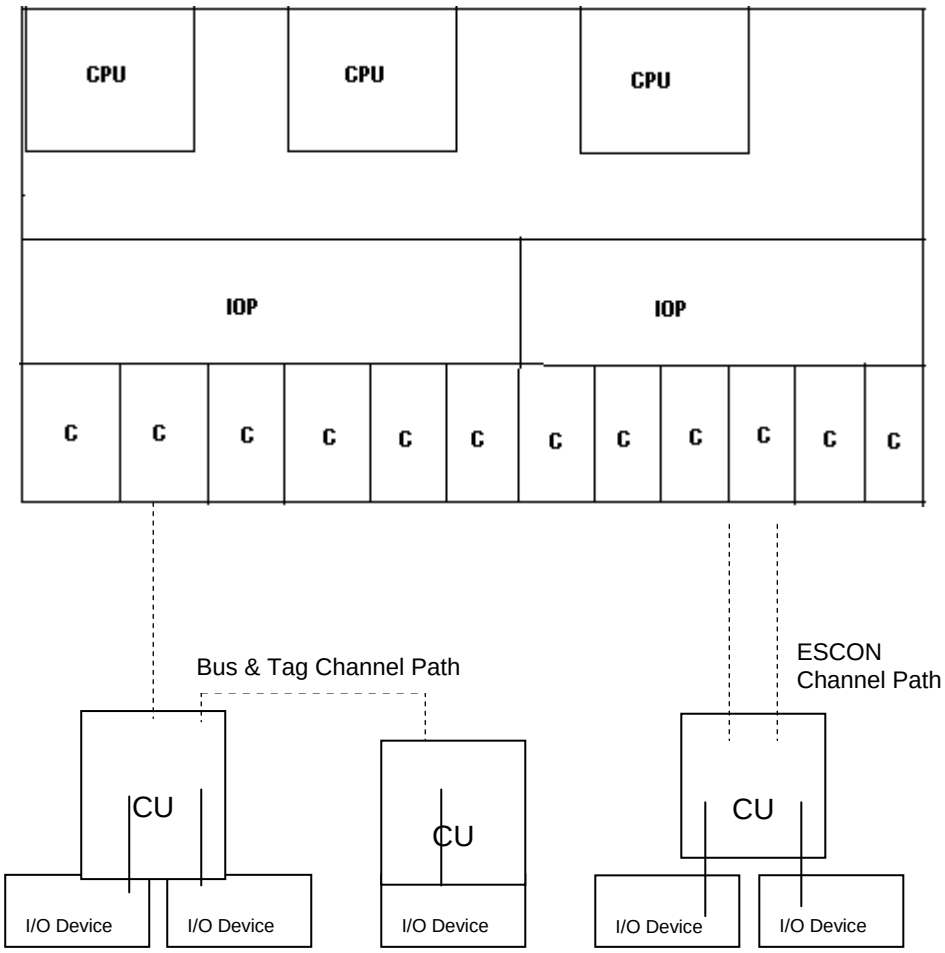


Fig 4.1 - Primary elements of a Mainframe Channel

C – Channel
IOP – I/O Processor
CU – Control Unit
CPU – Central Processing Unit

Sub channels

Each I/O device is represented by one subchannel.

Subchannels are similar to virtual circuits, providing information about the associated I/O device (including I/O operations and other functions) and its attachment to the channel.

A channel program is a collection of channel command words (CCWs) to be executed by the channel with the specific purpose of executing an I/O operation. Once the I/O subsystem has requested a subchannel operation, the host CPU is released for other work.

The channel then assembles or disassembles data and synchronizes the movement of data bytes between the I/O device and main memory.

I/O Transfer

IBM channels support two I/O transfer modes: burst mode and byte-multiplex mode. In burst mode, the I/O device monopolizes a channel path as long as required to move a burst of information. During burst-mode transfers, no other device can use the channel path. However, several I/O devices can share the ability to perform burst mode transfers over a given channel path.

In byte-multiplex mode, the I/O device is logically connected to the channel path for only a short period of time. Multiple devices using this mode can share a channel path, transferring small segments of information. Typically, devices with high data transfer rate requirement use the channel path's burst mode facility, whereas slower devices run in byte- multiplex mode.

Devices within a channel subsystem are addressed in different ways. The channel path identifier is a system-unique 8-bit value assigned to each channel path.

A sub channel number is a system-unique 16-bit value used to address a subchannel. I/O devices are addressed using a 16-bit value called the device number. The device number is independent of channel protocols, CPU model, or system configuration, and is used when system operators wish to communicate with systems. Finally, the device identifier is a protocol- and channel path-dependent address used to communicate with I/O devices.

PROTOCOL_FOR_MAINFRAME_COMMUNICATION:

PROTOCOL: A set of standards for the format and control of data being communicated within a system. Typically across a network, the definition of the communications between two pieces of hardware or software: format of each transmission, allowable responses and perhaps even timing.

Some of the protocols for Mainframe communication

- PRPQ1: Programming Request for Price Quotation.

IBM terminology for a customer request for a price quotation on alterations or additions to the functional capabilities of system control programming or licensed programs.

- PRPQ2: Product Request Price Quotation.

Product from IBM which is not on the price list – you have to go cap in hand to your IBM sales office to ask them to quote you a price. The PRPQ is a mechanism for controlling the spread of products which IBM is unenthusiastic about selling or supporting.

- PS/CICS: Personal Services/Customer Information Control System.

The mainframe version of Personal Services. Replaced by OfficeVision. PS/CICS for VSE was withdrawn December 1996.

- PSCA: Personal System Communications Adapter.

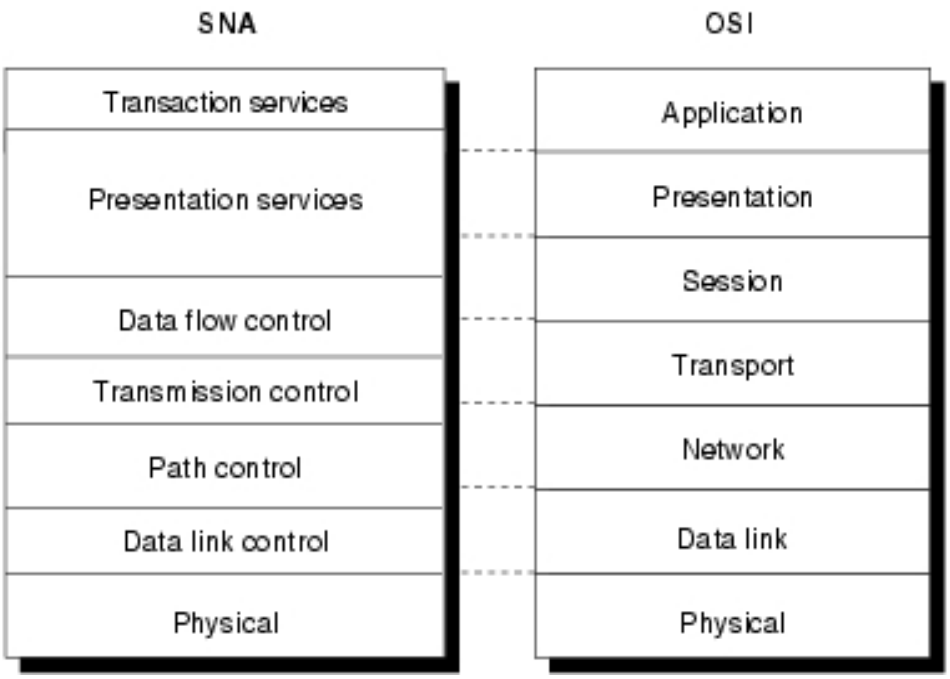
An obsolete adapter card that was once used to connect a microchannel-based PC to a mainframe parallel channel.

- PSCP: Presentation Services Command Processor.

A facility found in Tivoli Net View for z/OS, for formatting displays to be presented at a terminal and processing requests from a **terminal.**

WHY SNA networks are not transferred to TCP/IP?

SNA is one of the network protocol developed by IBM for IBM's mainframe computers. SNA - IBM's specification for the method of controlling data transmissions where data flow is simultaneous and bi-directional (the PC and host send and receive data at the same time).



SNA Maps to All Seven Levels of the OSI Model

IBM SNA model components map closely to the OSI reference model. The descriptions that follow outline the role of each SNA component in providing connectivity among SNA entities.

Data Link Control (DLC) defines several protocols, including the Synchronous Data Link Control (SDLC) protocol for hierarchical communication, and the Token Ring Network communication protocol for LAN communication between peers.

Path control performs many OSI network layer functions, including routing and datagram segmentation and reassembly (SAR)

Transmission control provides a reliable end-to-end connection service, as well as encrypting and decrypting services.

Data flow control manages request and response processing, determines whose turn it is to communicate, groups messages, and interrupts data flow on request.

Presentation services specifies data-transformation algorithms that translate data from one format to another, coordinate resource sharing, and synchronize transaction operations.

Transaction services provide application services in the form of programs that implement distributed processing or management services.

SNA does not define specific protocols for its physical control layer. The physical control layer is assumed to be implemented via other standards.

A key construct defined within the overall SNA network model is the path control network, which is responsible for moving information between SNA nodes and facilitating internetwork communication between nodes on different networks.

The path control network environment uses functions provided by the path control and Data Link Control (DLC). The path control network is a subset of the IBM transport network.

Summary

- Mainframe Channels -- IBM hosts are connected to each other and to communication controllers through high-performance communication subsystems called mainframe channels.
- Channels use one or more channel paths as the actual links between mainframes and I/O devices.
- IBM channels support two I/O transfer modes: burst mode and byte-multiplex mode.
- PROTOCOL: A set of standards for the format and control of data being communicated within a system.

Test Your Understanding

1). IBM hosts are connected to each other and to communication controllers through high-performance communication subsystems called_____
   - Mainframe Channels.
   - Channel Subsystems.
   - Control Units.
   - None of the earlier.

2). _____provides a reliable end-to-end connection service, as well as encrypting and decrypting services.
   - Data Link Control (DLC).
   - Data flow control
   - Path control
   - Transaction services.
   - None of the earlier

Session 05: Subsystems

Learning Objective

After completing this session, you will be able to:

- Describe Subsystems

Logical Partition

Logical Partitioning (PR/SM, MLPF or MDF) can be thought of as a layer between the operating systems and the underlying hardware. This layer presents a virtual image of the hardware to the OS similar to the OS presenting a virtual image of the hardware to the address spaces. Each S/390 hardware will have one or more distinct physical engines. The logical partition allows multiple logical partitions to share the physical engine and the number of logical partitions that can share an engine is installation specific. Each OS/390 will have its own subsystems. If a partition is not using capacity up to its limit that resource is made available to other partitions. This capability increases configuration options and can reduce the need for different channel, control units etc. because they can be shared. Logical partitions provide isolated beds for verifying changes to the systems and application software. The following fig 5.1 depicts the LPAR Overview.

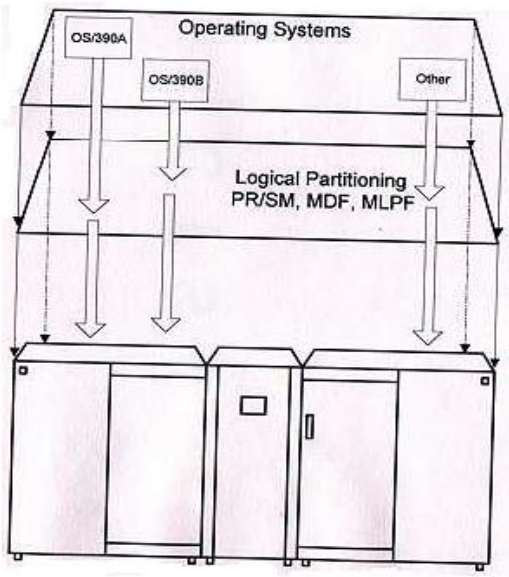


Fig 5.1 - LPAR Overview

PR/SM

PR/SM stands for Processor Resource/Systems Manager. It facilitates logical partitioning of S/390.

MDF

MDF stands for Multiple Domain Features. It enables the system resources to be logically partitioned up to 15 logical processing systems or domains. The number of partitions however depends on the availability of the resources. MDF allows dynamic CPU, I/O and storage

reconfiguration for domains, thus providing enhanced availability. MDF is a product of Amdahl and is compatible with PR/SM.

MLPF

MLPF stands for Multiple Logical Partitioning Facility. Like MDF it enables the system resources to be logically partitioned. MLPF is a product of Hitachi Data Systems.

Subsystems

MVS as an OS provides lot of different software products to perform several functions like database management (hierarchical/network/relational), transaction processing, application development etc.

These software products are not built as part of the OS but when installed run in a separate address space. While these products are not essential to maintain data for a business, the availability of these products makes data management and reporting easier. The following are the characteristics of a subsystem.

- A software product that operates in its own address space under the control of OS/390
- What happens in the subsystem's address space is entirely under the control of the subsystem. OS/390 does not have any control on what gets executed inside the subsystem address space
- Subsystems however use MVS to gain access to external resources like DASD space, Printers etc.,
- Different subsystems available under MVS are
    - TSO
    - CICS
    - IMS
    - DB2
    - JES
    - VTAM

JES

JES stands for Job Entry Subsystem. IBM's MVS and z/OS operating systems use a job entry subsystem (JES) to receive jobs into the operating system, schedule them for processing by MVS or z/OS, and to control their output processing. Each job is described to the operating system by system administrators or other users in job control language (JCL). The operating system then sends the job to the JES program. The JES program receives the job, performs the job based on priority, and then purges the job from the system. JES manages batch jobs for MVS. If JES does not come up nothing else could run including all the other subsystems. The importance of JES can be understood from the fact that when JES amends it is JES that could be brought up again. No other subsystem can be brought up because all other subsystems run as separate jobs. JES performs the following functions

- Read jobs into the system
- Convert them into internal machine-readable form
- Select them for processing
- Process their output
- Purge the completed jobs

Different Types of JES

- JES2
- JES3

JES3 allows central control of the processing of jobs using a common work queue.

JES3 has more network style dependency than JES2; as networking and inter-system dependencies have developed, this has become more practical than the single platform environment and single task processes that JES2 addresses.

Both OS/390 and MVS provide an interactive menu for initiating and managing jobs

Note: In a single processor installation both of them perform the same functions

What is a JOB?

A job is a set of tasks that have to be performed in a sequence by the system to satisfy some objective. There can be different objectives in a installation. To create a new file and copy the contents of an existing file could be one objective. To take a file as input and produce a sorted output could be another objective. Using the standard utility programs supplied along with MVS, one can satisfy these objectives. While the above mentioned jobs may be very simple, batch applications developed to address business problem could involve the application programs to perform operations with respect to the business problem.

How can a job be submitted?

A job can be submitted by creating a JCL and submitting that JCL to JES. This JCL will include one or more steps, that when executed will meet the defined objective. When a JCL is submitted as a JOB it goes through different phases before completion. The following diagram 5.3 pictorially explains what happens within the system when a JCL is submitted.

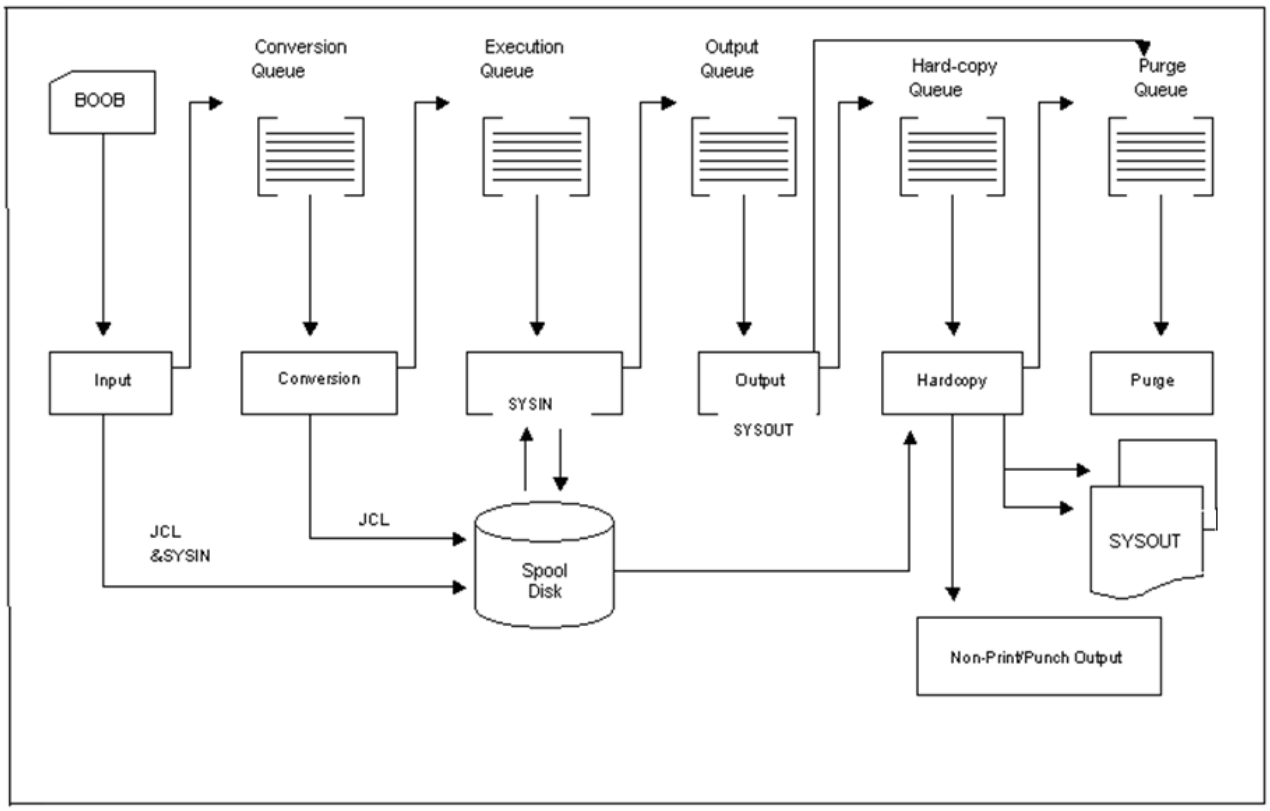


Fig 5.3 – Different phases a JCL goes through

- JES will scan the input stream, assign a job identifier to each job and places each job's JCL; optional JES control statements, and SYSIN data onto Spool data sets.
- When the initiator completes a job, it will request for a new job from JES
- JES will select job(s) based on the job class(es) that are assigned to the initiator and the priority order in which the job classes should be searched
- JES will select the best job available from the spool data sets and return it to the initiator for processing and subsequent running.
- Initiator will place the job on the conversion queue
- The converter takes the job's JCL, merges it with JCL from a procedure library (such as SYS1.PROCLIB), and converts the composite JCL into converter/interpreter text that both JES and the job scheduler functions of MVS can recognize into the conversion queue.
- MVS initiators requests for a JOB and JES will respond by selecting an appropriate job from the Execution Queue
- All the resources requested by the job step will be allocated by the initiator
- The job will be executed until there are no more steps in the job
- During the execution process the job will be in constant interaction with the SPOOL
- During execution any output that is created by the job or any system message that must be printed will be moved to the output queue.
- After the job finishes JES will analyze the characteristics of the job's output in terms of its output class and device setup requirements
- Output queue can have output that is to be processed locally or output to be processed at a remote location

- When all processing completes JES will release the spool space assigned to the job

TSO

TSO Stands for Time-Sharing Option. Time-sharing means that many persons can access MVS concurrently but each is unaware that the operating system is being accessed by others - so it appears to each TSO user that he or she is the only user on the system. This is a base element of OS/390. It allows users to communicate with MVS interactively and share computer time and resources. Each user will be given a unique id (8 characters maximum) and an address space. Every user will be provide with a profile and a log. TSO provides commands to manage datasets viz. Opening datasets for Edit, Copy datasets, Delete datasets, Rename datasets etc. All these command can be entered through an ISPF panel. To gain access to other subsystems a user has to log to TSO. TSO interacts with users in either a line-by-line mode or in a full screen, menu-driven mode. In the line-by-line mode, the user enters commands by typing them in at the keyboard; in turn, the system interprets the commands, and then displays responses on the terminal screen. But most mainframe interaction is actually via ISPF which allows for customized menu-driven interaction; the combination is referred to as TSO/ISPF. TSO also can be used to provide a Unix-style environment on OS/390 and z/OS via the UNIX System Services command shell, with or without ISPF.

ISPF

ISPF is software that runs as a part of TSO and provides dialog management service. Dialogs allow a TSO user to issue commands directly or indirectly from panels (screens).

The ISPF and its Program Development Facility (ISPF/PDF) work together with TSO/E for users to interact with system resources through panels.

It is software that runs as a part of TSO and provides dialog management service. Dialogs allow a TSO user to issue commands directly or indirectly from panels (screens).

ISPF provides an interface based on panels and action bars, and exploits many of the usability features of the Mainframe.

ISPF provides a suite of application development tools. These tools include an editor, tools to view job output, a TSO command processor, panels to perform TSO operations on datasets indirectly, panels to display catalog information etc.

It includes a screen editor, the user interface of which was emulated by some microcomputer editors sold commercially starting in the late 1980s, including SPFPC.

ISPF will maintain a profile for each user. This profile controls the communication between the system and the user's terminal. Profile contains various information like the terminal characteristics, PF key setting etc.

CICS

CICS Stands for Customer Information and Control System. CICS is a Transaction processing system.

- Works with VTAM to support large networks of terminals that can run interactive application programs written in various languages
- Provides multi-user and multiprogramming facilities
- Can work as a transaction manager
- Programs running in CICS region can access data from IMS and DB2 databases and from VSAM datasets
- CICS has its highest profile among financial institutions such as banks and insurance companies, over 90 percent of Fortune 500 companies are reported to rely on CICS (running on z/OS) for their core business functions
- Each CICS program is initiated using a transaction id. CICS screens are sent as maps using a programming language such as COBOL.

CICS is available on all IBM platforms

IMS

IMS Stands for Information Management System. IMS consists of two parts

- DL/I (Hierarchical Database)
- DC (Data Communications)

In a hierarchical database, data elements are organized in a hierarchical structure. In other words a hierarchical structure maintains a parent and child relationship between the independent and the dependent elements. Access to the child elements must be through the parent element(s). Data elements are grouped in segments and a segment is the unit of communication between IMS and an application program.

- With both the parts used, IMS can be a Transaction Manager as well as a database. IMS TM can handle this whole process thousands (or even tens of thousands) of times per second
- IMS/DC implements multiuser and multiprogramming facilities

DB2

DB2 Stands for DataBase2. DB2 is the Relational DBMS on MVS. Any relational model provides improved data availability, data integrity and data security because the model defines these features as part of the database. Since DB2 is a relational model database data from the tables can be accessed using SQL statementsDB2 can be implemented as a distributed database, which enables it to be accessed from applications running on any platform. DB2 can store enormous amount of data. Each table can grow up to 1 terabyte.

Almost all the new application development tools like VB, JAVA, ASP, JSP etc. provide gateway to DB2 running on MVS. Through these gateways data on the enterprise system can be accessed.

- Does not have a separate Transaction Manager unlike CICS/IMS
- Works with CICS/IMS in On-line environment and with application programs in a batch mode

VTAM

VTAM Stands for Virtual Telecommunication Access Method. It controls local and remote terminals for other subsystems

- Part of the comprehensive telecommunications product called SNA (Systems Network Architecture)
- VTAM runs in its own address space
- Provides centralized control over all terminal devices attached to OS/390
- VTAM application programs that run in other address spaces, communicate with those terminal devices indirectly by issuing the request to VTAM which in turn services the request for the appropriate terminal

RACF

RACF Stands for Resource Access Control Facility. It is the Security server to protect valuable corporate data. It provides the security tools that an installation needs to implement a security system. It is a combination of modules that interface with all the MVS subsystems. RACF is not a subsystem because it is not started as a separate task. In addition to being one of the most mature and scalable security monitors in computing, it has some interesting features that are not often found in Microsoft Windows or UNIX environments. For example, it can set permissions for file patterns — that is, set the permissions even for files that do not yet exist. Those permissions are then used should the file (or other object) be created at a later time.

- Maintains separate profiles for individual users and the resources in its own database
- Allows access to data as defined in the profile for the user
- Identify and authenticate users
- Authorize users to access the protected resources
- Log and report various attempts o unauthorized access to protected resources

Provides security for resources under CICS/IMS/DB2RACF

Internal Readers

- An internal reader is a program that other programs can use to submit jobs, control statements, and commands to JES.
- JES initialization statements define these internal readers which JES2 also allocates during its initialization processing

Initiators

An initiator is a system program that either the operator starts or that JES starts automatically when the system is initialized.

- Runs in the system region of the address space that's eligible for batch processing
- Has one or more job classes associated with it and can run jobs that are
- It reads and processes operation control language statements from the system input device.
- After JES selects a job and passes it to the initiator, the initiator invokes the interpreter to build control blocks from the converter/interpreter text that the converter created for the job
- Allocates the resources specified in the JCL for the first step of the job
- Starts the program requested in the JCL EXEC statement

Queues

A queue is a line or list formed by items in a system waiting for processing. Queues are intermediate storage locations where jobs are stored during the various stages of a batch run. There are different queues in a batch run.

- Conversion Queue – Jobs waiting to Run
- Execution Queue – Jobs currently running
- Output Queue – Jobs waiting for their output to be produced
- Hard-copy Queue – Jobs having their output produced
- Purge Queue – Completed jobs waiting to be purged from the system

Class

- A named set of job processing and scheduling rules
- Disciplined approach to specifying rules for JES to follow in processing jobs, scheduling jobs and assigning resources to jobs
- An important parameter to identify the job's importance
- For example CPU time for a job (all steps inclusive) is determined by the class in which the job is submitted

Spool

Spooling is a process by which the system manipulates its work. It provides temporary storage area for work that is yet to be completed

- Spool stands for Simultaneous Peripheral Operations Online
- Spool is a specialized DASD
- Using storage on direct access storage devices as a buffer storage to reduce processing delays
- Performing independent operations like printing while the system is busy with other work

Summary

- Subsystems are software products to perform specialized functions.
- JES, TSO, CICS, DB2, VTAM, IMS are the Commonly Available Subsystems.
- A job is a set of tasks that have to be performed in a sequence by the system to satisfy some objective.
- IMS consists of two parts
  - DL/I (Hierarchical Database)
  - DC (Data Communications)
- DB2 Stands for DataBase2. DB2 is the Relational DBMS on MVS.
- DB2 can store enormous amount of data. Each table can grow up to 1 terabyte.

- A queue is a line or list formed by items in a system waiting for processing. Queues are intermediate storage locations where jobs are stored during the various stages of a batch run.

Test Your Understanding

1). What kind of database is DB2?

- Hierarchical database
- Relational database
- Flatfile database
- None of the earlier

2). What does VTAM stand for?

- Virtual Telecommunication Access Model
- Virtual Technology Access Method
- Virtual Telecommunication Access Method
- None of the earlier

Session 06: Emulator

Learning Objective

After completing this session, you will be able to:

- Explain the Emulator

Overview

Emulation is the way by which PC can connect to a mainframe.

"Emulators" are special software that does the above, developed and supported by many vendors including IBM.

This section outlines the need and types of emulators.

Dumb terminals

Dumb terminals were used in earlier days to connect to mainframe, which is located in a remote location. Dumb terminals do not have any processing capabilities as they do not have any microprocessor/CPU inside them. Due to this they will be very slow in processing user's requests. They do not support many display features like color customization, Screen size customization, etc. They have poor edit features, as they do not support the standard GUI features like cut, copy & paste. It does not allow any other application to run simultaneously as well.

Need for emulators

The increasing need to use a PC for non-programming purposes, such as word-processing, spreadsheet, communications etc. created the need to have a PC on the desk, next to the IBM Terminal. Since this created a space crunch in the offices, a new class of IBM terminals came into existence. These "terminals" are PCs / workstations made to work like an IBM terminal. This is done by a process called Emulation. Here, each PC will run an emulator program. This program converts the PC data from ASCII to EBCDIC, and sends it to the mainframe. Upon receipt of data from the mainframe, the emulator converts it from EBCDIC to ASCII. These emulator programs are of the Terminate and Stay Resident (TSR) type. **Emulators a**llow remote PCs to communicate with IBM-Mainframe. Emulator session is another application like Word / Excel in your PC

Types of emulators

Line terminal emulators

These are the first generation emulators. Traditionally, ASCII (also known as line-mode or line-oriented) terminals connected to an IBM system had been limited to using linemode, where each line of output from the host computer is written at the bottom of the screen and then the terminal scrolls everything on the screen up one line to make room for the next line of output. Although this mode of operation is suited for hardcopy devices with continuous forms of paper, it does not take

advantage of the capabilities of modern video display terminals. As a result, using a line-oriented terminal to display and edit data and text can be very tedious.

Full screen/3270 Emulation

3270-type (also known as full-screen) terminals can display an entire screen of text or data in one operation (typically 24 lines of 80 characters each), then wait for you to signal that you are ready to go on to the next screen of information. Editing text and data is easy, since you can move the cursor to the area on the screen containing the data to be changed and then type new data over the old. 3270 emulation exploits modern video display terminals. P-COM is the full screen emulator by IBM.

Personal Communications (PCOM)

Overview

Personal Communications enables access to applications and data residing on mainframe. This market leading emulator, developed by IBM, contains state-of-the-art, comprehensive tools that simplify connectivity and access to host data.

PCOM provides full screen and full function emulation. In addition to this it also provides the following functions:

File transfer: PCOM allows you to do file transfer between the host and your PC. Files can be transferred in text or binary mode. Send File to Host allows you to send files from your PC to the host system; with one command, you can send a single or several files. If you often send the same list of files, you can save the list of file-names and send all the files with one command.

Receive File From Host allows you to Receive files from a host system to your PC; with one command, you can receive a single or several files. If you often receive the same list of files, you can save the list of file-names and receive all the files with one command.

GUI features: Many of the standards GUI features like cut, copy & paste can be done in PCOM.

Record and playback scripts for automation: We can record the sequence of keystrokes for repetitive tasks and play back the same for automating such repetitive tasks.

APIs for VTAM based programming: Application programming interface for VTAM programming is also supplied with PCOM. VTAM is discussed in detail later in this document.

Configuring PCOM

PCOM need to be configured before it can be used. The following are the key configuration parameters:

- **Session type**: This can either be a display or printer.
- **Screen size**: This will determine the number of lines and columns that are displayed by the emulator. It can be 24 X 80, 43 X 80, or a higher resolution
- **Communication link**: Communication types have to be configured. It can be TCP/IP, APPC, IEEE standard, and so on
- **IP address**: Specify IP address of the host machine.

Customizing PCOM

PCOM allows a lot of customization for user friendliness. This section lists down some of the frequently used customizable features.

Customizing keyboard

We can customize keyboard layout in the emulator session. For example, if you are working for an UK customer, then you are likely to use British currency symbol '£' quite often in your programs and files. If you need to type it with the standard keyboard, it will be a tedious task. Instead, you can map any of the keys (say, numeric key 4+Shift) to the symbol '£'.

Apart from this you can also enable Keystroke Buffering. This allows the buffering of keystrokes when input is inhibited

Summary

- Emulators are special software developed by many vendors to interface mainframe.
- There are two types of Emulators:
    - Line terminal emulators.
    - Full screen/3270 emulators.
- PCOM provides full screen and full function emulation.

Test Your Understanding

1). What is an emulator?
    - Special software developed to interface mainframe
    - Tool to integrate mainframe applications with web applications
    - Special software to run jobs in mainframe
    - None of the earlier

2). What does emulator do?
    - Simulate mainframe data sets in PC
    - Simulate the mainframe environment in PC
    - Simulate mainframe jobs in PC
    - None of the earlier

Session 07: Application Development

Learning Objective

After completing this session, you will be able to:

- Describe Application Development

Applications

Applications usually evolve out of business requirements/problems faced by the customer. Applications are also developed to replace manual repetitive operations, with automation. The purpose of doing this is, to ensure more productivity and more accuracy.

Some of the examples of applications are Payroll, Billing, and Stock Control, Reservation systems. In all these applications, there are predefined rules, functions and processes. When an application is studied from IT perspective, for automation purposes, the business rules translate

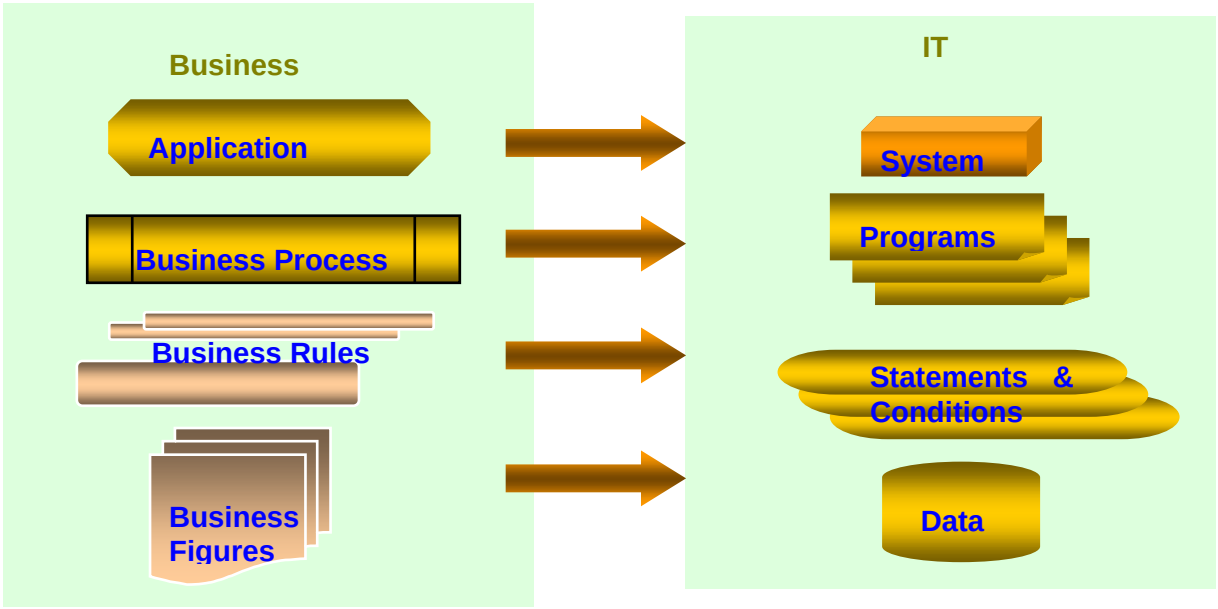


Fig 8.1: Application Vs System

themselves in to conditions and validations of data maintained within the system. The business processes translate themselves into programs and files and records maintained in the manual scenario translate themselves into databases. In short the Application itself translates into a System. This is explained in figure 8.1.

A collection of business processes is an application. For example in a payroll processing application, Calculation of salaries for employees is a process, Printing the Pay slips is a process, Updating the Year-to-date figures is a process, Handling full and final settlements is a process.

Each of these processes has a defined set of tasks/activities. The data that is maintained by the application needs to be validated. This forms the part of business rules. This is explained in figure 8.2.

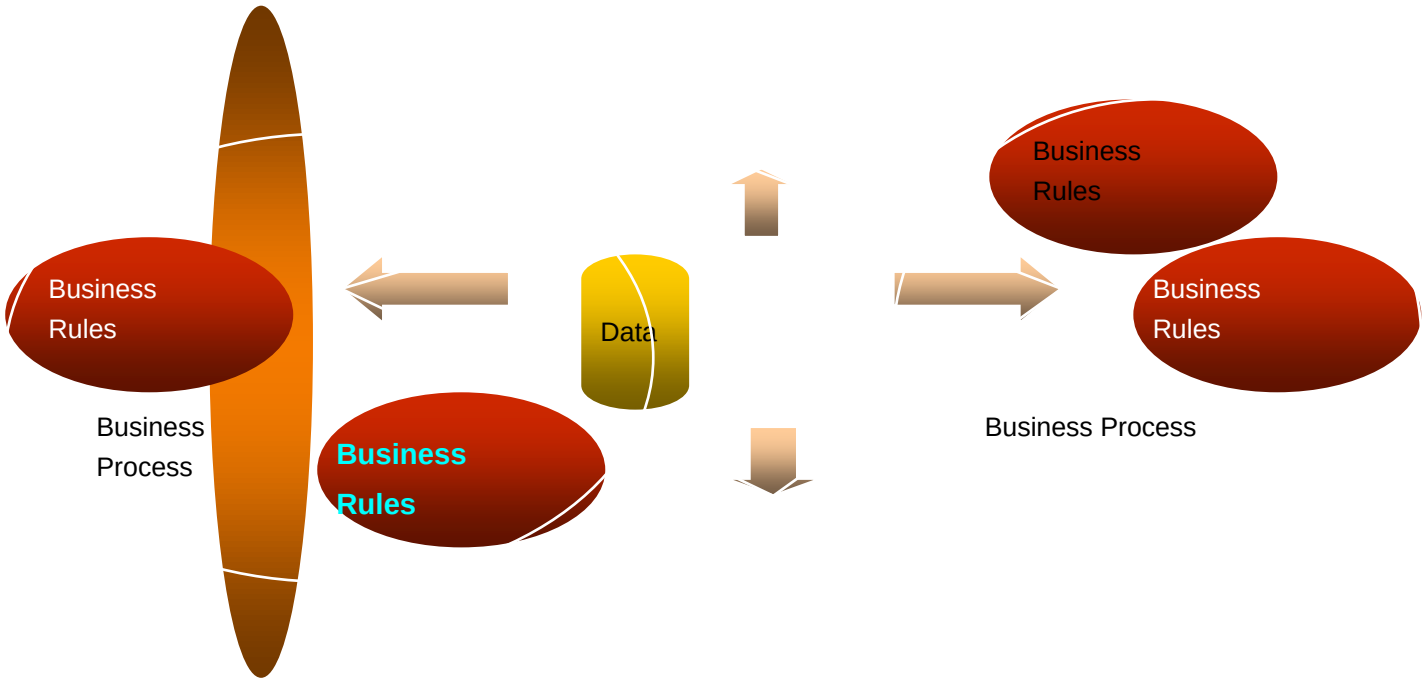


Fig 8.2: Application Overview

Application development involves studying the requirements of an application, besides what the users want. Based on these, the database manager to be used, the user interface to be used, the database structure, the user menus etc are designed, and the application is built around these details.

While designing the application the following points need to be considered

- Break up into functions, and assign functions to programs
- Decide the data structure
- Format the report layouts
- Develop the user interface screens / menus

After the study of the requirements, the entire application is broken into independent functional units. Then these functional units are assigned to programs. The data structure, which is the most important part of the application, is finalized, after analyzing what data is to be stored, and how. The report formats are finalized according to the users' requirements. The user interface, in the form of menus / data input screens are also designed.

The next sets of activities that need to be considered are

- Language(s) to be used
- Database manager to be used
- User interface to be used
- Develop programs with embedded function calls
- Compile and link the programs

Now comes the technical part of the system. Based on database requirements, user interfaces requirements etc., the database manager and user interfaces are finalized. Once this is done, there is a platform - a base for development. This is explained in figure 8.3.

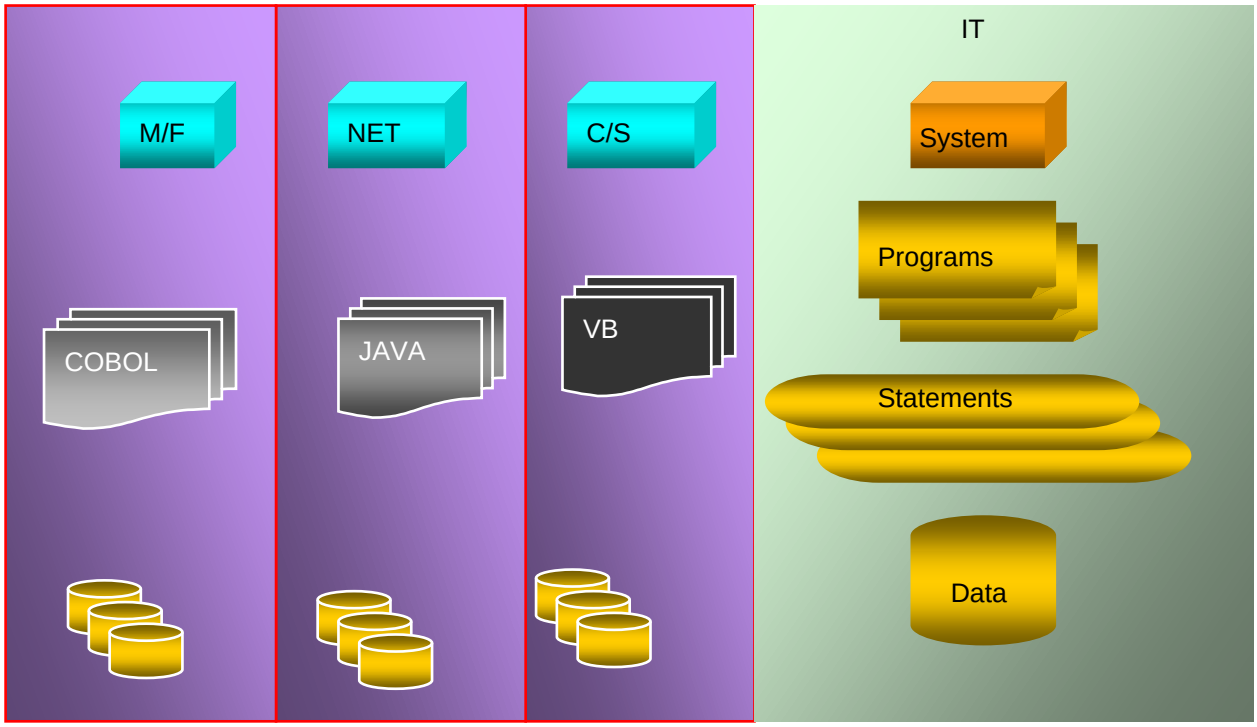


Fig 8.3: Application Development platform overview

Business Processes

Processing is the act of doing something (e.g. organizing and manipulating) with data to produce output.

Basically the business processes that make up different applications can be categorized under 2 broad areas from IT perspective. They are

- Batch Processing
- Online Processing

In simple terms, if the application does not require any user interfaces (user intervention) then it can be termed as a batch application. Ex: Printing of Bills system, Payroll processing etc.

Those applications that require user interfaces are the online applications. Typical examples are Railway reservation system etc.

DB2
VSAM
IMS

UDB

ORACLE
SQL

Batch Processing

Using **batch processing** input data is grouped together into **batches**. No data is processed until a complete batch has been collected. Then all of the data in a batch is processed in one go. There may be a long time delay between data being input into the system and the data being processed.

In this type of system the important data that the computer stores all of the time is kept in a file called the master file. The data in the **master file** is **sorted** into order using one of the fields in the records in the file, known as the primary key field. The primary key field must uniquely identify each record in the file.

Each piece of input data (which will update the contents of the master file) is known as a **transaction**. All the input data is put together into a batch in a file called the **transaction file**. There are three different types of transaction that any processing system will have to cope with.

- Add a new record to the master file.
- Delete a record from the master file.
- Amend / update an existing record in the master file.

At some predetermined time (e.g. the end of the day or week) the computer system will **process** the data stored in the transaction file and make any changes that are necessary to the master file as a result of the transactions. This will produce an **updated master file** and an **error report** detailing any transactions that could not be processed for some reason.

Online Processing

This is also called as Transaction processing. Transaction processing is **interactive**, i.e. processing takes place as a "conversation" between the user and the computer: The computer responds to the user's input by outputting some data before the user can input any more data.

Transaction Processing is also known as interactive, online or Pseudo Real Time Processing.

For some applications the **master file** needs to be kept up to date all of the time. For example in a travel agent whenever a seat is booked on a flight the number of seats that remain available on the flight must be reduced by one immediately. If this update was not done until the end of a day (as might happen in a batch processing system) then the flight could become overbooked with the same seat being booked more than once.

Transaction processing systems are used whenever the **master file must be kept up to date**. A transaction processing system operates like this:

- When a transaction is entered it is placed in a **queue** of transactions waiting to be carried out. The transactions are processed in the order that they are placed into the queue. If there are many people using the system at the same time then there could be lots of transactions being made from different computer terminals.
- The computer system will process one transaction at a time. Once the system starts processing a transaction it will not process any other transactions until the current transaction is finished. When a transaction is processed the master file is updated immediately. Therefore the master file is always kept up to date.

Under normal circumstances a well-designed transaction processing systems will process a transaction within a few seconds of the transaction being inputted. During peak processing times

(E.g. Saturday morning in a travel company) the time that it takes for a transaction to be processed will increase. For most systems this is acceptable. This is tabulated in figure 9.4.

Comparison	Batch	Transaction
Identifying Features	Input data is collected into batches in a transaction file which is sorted and validated with old master file before producing the new master file	The master file is always kept up to date. Transactions are placed in a transaction queue and processed in the order that they are added to the queue.
Example Applications	1. Payroll	1. Ticket Booking
	2. Printing Bills	2. Seat Booking
	3. Multiple Choice Exams	3. Stock Control
		4. Banking
Speed of Response	No data is processed until a complete batch is collected. It may be a long time e.g. a day before data is processed.	Under normal loads data is processed within a few seconds. Under heavy loads processing times can be longer.

Summary

- Application is a collection of business processes.
- Business processing options are online processing and batch processing.
- Interactive applications are online applications that run on CICS / IMS subsystem.
- Non interactive application are batch applications that are submitted through a JCL

Test Your Understanding

1). Transaction Processing is also known as:
   - Interactive
   - Online.
   - Pseudo Real Time Process.
   - All the earlier.

2). Payroll processing is an example of:
   a) Batch processing.
   b) Transaction processing.
   c) Business processing.
   d) Online processing.
      - Only a not b
      - Both b and c.
      - Only c not a, d, and b
      - Both a and d.
      - None of the earlier.

Session 08: Overview on Legacy Application Modernization

Learning Objective

After completing this session, you will be able to:

- Understand Legacy Application Characteristics & Challenges.
- Understand the need for Modernizing Legacy Applications.
- Describe the Styles of Legacy Application Modernization.
- Understanding the need for Legacy Applications Integration
- Understand on the Integration scenario and Typical Mainframe Integration Patterns

Legacy Application Characteristics

"Legacy" refers to existing IT assets like mainframe applications that have been deployed anywhere from yesterday to twenty years ago. Legacy software and applications are often considered as

- Running "Business Critical" processes.
- Result of acquisition, take-over or merger.
- Containing valuable & embedded business logics that is unique & difficult
- Containing tightly coupled services
- Containing multiplicity of interfaces

Legacy Application Challenges

Challenges associated with Legacy system are

- Inconsistent underlying architecture & independently developed applications
- Having overlapping and redundant functionality as well as data
- Representative of many years of developments, enhancements & modifications
- Highly complex environment which incur huge maintenance cost
- Having redundant and non-reusable code

Understanding the need to Modernize Legacy Applications

Most companies have significant investment in their legacy assets. Management is typically not open to ripping out and replacing aging applications in spite of the many shortcomings. Primary pain points of legacy applications usually falls into one of the followings:

- **High cost of ownership**, including maintenance, operation & upgrade of software / hardware
- **Slow time to market,** because of complex and poorly understood code
- **Limited business agility and flexibility**
- **Monolithic Architecture** with little modularity together with redundant code
- **Fragmented and redundant data**

- **Closed and outdated technology** that is difficult to integrate and interface
- **Decreasing vendor support**
- **Lack of application knowledge**
- **Shrinking talent pool of developers skilled in legacy systems**

Styles of Legacy Application Modernization

Four major styles of application modernization and transformation exist

- **Transform the user experience:**
  Change the current 3270 green screens to Web user interface, which is fairly simple, feature rich & cost efficient and does not change the underlying code.

  - 3270 Bridge
  - CICS Web Interface
  - HATS – Host Access Transformation Services

- **Transform the application connectivity:**
  Change the way that users or client application can connect with the existing applications. This is used to integrate heterogeneous applications.
  - Accessing DB2 from Java program - JDBC
  - IMS Connect
  - WebSphere Connectors
  - CICS Transaction Gateway

- **Transform the architecture:**

  - **Data Architecture:**

    Change the application database or how database is accessed by the applications.
    - Migration from IDMS to DB2.
    - Migration from IMS to DB2.
    - Accessing database using Data Access Module or Stored Procedures.
  - **Application Architecture:**

    Change the way the application is built by making it tiered, more modular and making it easier to compose business services with the right level of granularity.
    - Multi Tier Architecture / Distributed application architecture
    - Service Oriented Architecture using CICS Web Services

- **Platform Architecture**

  Change the underlying application platform itself.

  - Java on Linux running in Z Enterprise system

- **Transform the batch processing environment:**

  Change the existing batch environment to modern and high performance batch processing environment capable of meeting new requirements even if existing batch process is implemented optimally.

  - Generating documents in PDF
  - Processing XML
  - Sending E-mails
  - Implementing new industry standards
  - Batch window needs to be shortened & made more efficient
  - Running batch at any time

Understanding the need for Legacy Applications Integration

Mainframe applications rarely exist in isolation. Often several applications need to work together by exchanging information both in real-time or in deferred mode to accomplish mission critical business functions. Enterprises are typically comprised of several types of applications operating in multiple tiers of different operating system platforms. They can be roughly termed as

- Custom built applications
- Legacy applications
- ERP applications
- Third party applications
- Web Based applications
- Merger / Acquisition applications etc

Integration Scenario

Integration means connecting computer systems, companies and people. Some of the common integration scenarios are

- **Information Portals:**

  Many business users have to access more than one system to answer a specific question or to perform a single business function. Sometimes UI integration helps provide sophisticated user interaction.

- **Data Replication:**

  Many business systems require access to the same data. For example, a customer's address may be used in the customer care system (where the customer calls to change it), the accounting system (to compute sales tax), the shipping system (to label the shipment) and the billing system (to send an invoice). Many of these systems are going to have their own data stores to store customer related information. When a customer calls to change his or her address all these systems need to change their copy of the customer's address. This can be accomplished by implementing an integration strategy based on data replication.

- **Shared Business Functions:**

  In the same way that many business applications store redundant data, they also tend to implement redundant functionality. Multiple systems may need to check whether a social-security number is valid, whether the address matches the specified postal code or whether a particular item is in stock. It makes business sense to expose these functions as a shared business function that is implemented once and available as a service to other systems.

- **Service Oriented Architectures:**

  Shared business functions are often referred to as services. A service is a well-defined function that is universally available and responds to requests from "service consumers".

- **Distributed Business Processes:**

  One of the key drivers of integration is the fact that a single business transaction is often spread across many different systems. In most cases, all relevant functions are incorporated inside existing applications. What is missing is the coordination between the applications. Therefore, we can add a business process management component that manages the execution of a business function across multiple existing systems.

- **Business to Business Integration:**

  Integration frequently occurs between business partners. A customer may contact a retailer to inquire on the price and the availability of an item. In response, the retailer may ask the supplier for the status of an expected shipment that contains the out-of-stock item.

Typical Mainframe Integration Patterns

Mainframe integration usually refers to the need to share information in real time or batch between a software application running on a mainframe and other computing environments. Legacy integration patterns can be classified into 4 major categories.

- **File Transfer**
- **Shared Database**
- **Remote Procedure Invocation**
- **Messaging**

Major patterns of application integrations are

- **Mainframe Database Adapters** – a database adapter connects directly to the mainframe database. These adapters can normally read, write, erase and update data in the mainframe database as well as deliver the mainframe data via certain protocols.
- **Mainframe ODBC drivers** – Another approach is to use an ODBC driver, such as the one built-in to the SNA server or one of many available from third-party software vendors. The ODBC driver has the advantage of being more widely accessible by third-party software vendors
- **File Transfer Protocol** – FTP uses TCP/IP and provides a way to transmit files between diverse systems including mainframes, Linux systems, IBM, UNIX systems and Windows servers.
- **Proprietary API –** These APIs serves specific integration need of a specific application. They are essentially single purpose (single integration point based)
- **3270 Adapters -** These adapters will publish a secure Web service that enables bi-directional communication with mainframe systems and applications via UI stream known as 3270.
- **Message Queues – MQ** – WebSphere MQ on the mainframe provides message brokering capabilities and greatly enhances the system integration. In many respects, messaging is the best gateway to enterprise system integration.
- **CICS –** is a mainframe transaction server designed for mostly interactive rapid, high-volume online processing and can also perform background processes. It provides simple and secure entry-point into the mainframe application environment without the need for extensive programming.

Summary

- Legacy Software and applications are considered as running "Business critical" Processes.
- Shrinking talent pool of developers skilled in legacy systems stresses the need for modernizing legacy applications.
- Transforming the user experience, application connectivity, architecture are some of the styles of legacy application modernization...

- Mainframe integration involves sharing information between mainframe and other computing systems.

Test Your Understanding

1). Which of the following is not a reason for the need to modernize legacy applications?
   a) High cost of ownership
   b) Poor system performance
   c) Shrinking talent pool
   d) decreased vendor support

2). Which of the following provides message broking capabilities?
   a) CICS
   b) DB2
   c) Breaker
   d) Websphere MQ

Glossary

MAINFRAME: A mainframe is a continually evolving general purpose computing platform incorporating in it architectural definition the essential functionality required by its target applications.

SYSPLEX: A Sysplex is a collection of MVS systems that cooperate, using certain hardware and software products, to process work.

ADDRESS SPACE: The size of the address space will be 2^n-1 for a 'n' bit processor.

PAGING IN & PAGIN OUT: The process of getting the page into the real storage and throwing the page out into the DASD (Direct Access Storage Device) is called paging in and paging out (collectively called paging).

DATASPACE-Dataspace is fully controlled by the user programs and is available at real storage subjected to swapping and paging.

HIPER SPACE: Hiperspace are used by MVS and is available to user programs as 4K blocks.

MAINFRAME CHANNELS: IBM hosts are connected to each other and to communication controllers through high-performance communication subsystems called mainframe channels.

PROTOCOL: A set of standards for the format and control of data being communicated within a system.

SNA: SNA does not define specific protocols for its physical control layer. The physical control layer is assumed to be implemented via other standards.

JOB: A job is a set of tasks that have to be performed in a sequence by the system to satisfy some objective.

ISPF: It provides an interface based on panels and action bars, and exploits many of the usability features of the Mainframe.

CICS: CICS Stands for Customer Information and Control System. CICS is a Transaction processing system.

VTAM: VTAM Stands for Virtual Telecommunication Access Method. It controls local and remote terminals for other subsystems

INTERNAL READRES: An internal reader is a program that other programs can use to submit jobs, control statements, and commands to JES.

INITIATORS: An initiator is a system program that either the operator starts or that JES starts automatically when the system is initialized

QUEUES: A queue is a line or list formed by items in a system waiting for processing.

SPOOL: Spooling is a process by which the system manipulates its work. It provides temporary storage area for work that is yet to be completed.

EMULATORS: “Emulators” are special software that does the above, developed and supported by many vendors including IBM.

PROBLEM ANALYSIS: problem analysis is usually to find and fix the underlying error so that the problem will not recur.

DUMP: Dumps is a Record of the contents of some parts of storage at some point in time.

BX-Byte Index

CCWs- Channel Command Words

CF- Coupling Facility

CICS- Customer Information and Control System

CPC- Central Processor Complex

DB2- DataBase2

IMS- Information Management System

IPCS- Interactive Problem Control Systems

ISPF- Interactive System Productivity Facility

JES- Job Entry Subsystem

LSPR -Large System Performance Reference

MCH- Machine Check Handler

PCOM- Personal Communications

PDF- Program Development Facility

PX-Page Index

RACF- Resource Access Control Facility

SAR- Segmentation And Reassembly

SNA- Systems Network Architecture

SX-Segment Index

SYSPLEX- ***Sys***tems com***plex***

TSO- Time-Sharing Option

VTAM- Virtual Telecommunication Access Method

VTOC- Volume Table of Contents

References

Websites

www.bitpipe.com
www.enterpriseappspipeline.com

Books

- IBM's System/390 by Computer Technology Research Corp.

STUDENTS NOTES: