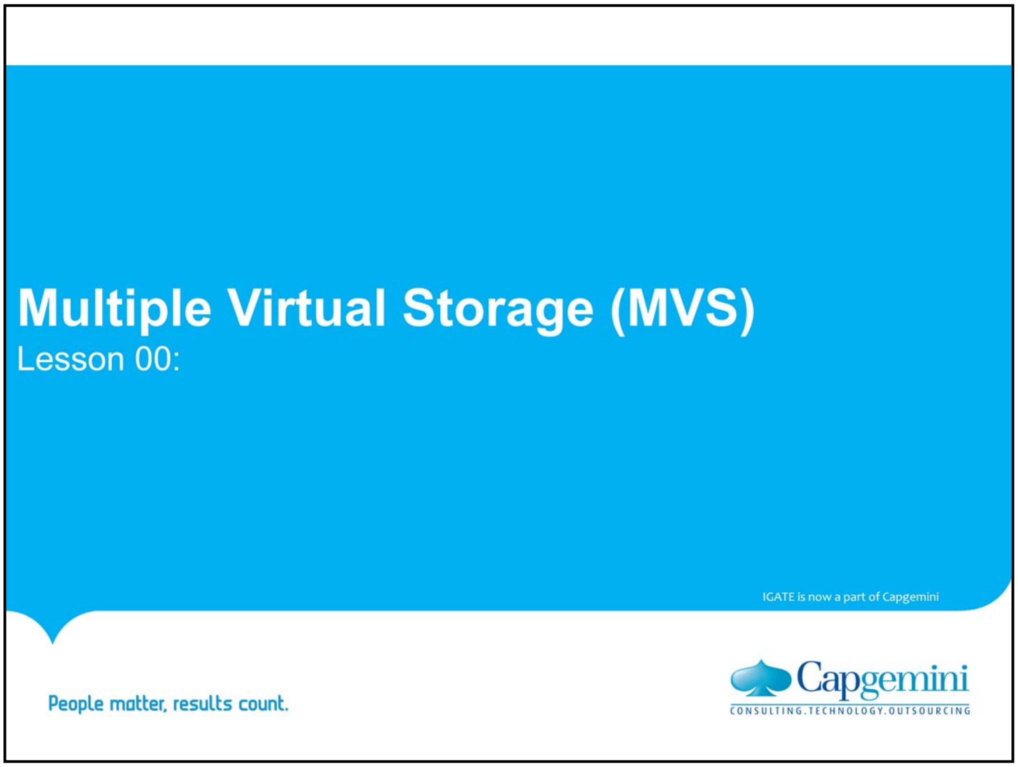

Multiple Virtual Storage (MVS)
Lesson 00:
IGATE is now a part of Capgemini
People matter, results count.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Document History

Date	Course Version No.	Software Version No.	Develope r / SME	Change Record Remarks
07-July-2005	1.0			Created
26-Oct-2009	2.0D		Padmaja Purandare	Revamped
07-Dec-2009			CLS Team	Review
10-Oct-2010	3.0		Vaishali Kasture	Revamped
30-June-2011	4.0		Rajita Dhumal	Revamped
21-June-2016	5.0		Veena Keshavalu	Revamped post the integration

 2

Course Goals and Non Goals

- Course Goals
  - Basic understanding of MVS job and data management
  - Basic understanding of IBM Mainframe system
  - Introduction to Mainframe environment
  - Work with basic ISPF commands and dataset utility

- Course Non Goals
  - JCL coding

 3

Pre-requisites

- Basics of Computer

Intended Audience

- Developers
- Programmers

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Day Wise Schedule

- Day 1
  - Lesson 1: Introduction to OS Concepts and MVS
  - Lesson 2: MVS Environment Concepts
  - Lesson 3: MVS Evolution
- Day 2
  - Lesson 4: Typical IBM Mainframe Systems
  - Lesson 5: MVS Concepts and Terminology
  - Lesson 6: MVS Data Management

Day Wise Schedule

- Day 3
  - MVS subsystems
  - System Generation & Initialization
  - Job Management Overview
  - Appendix
    - Appendix A – Compilation process (Translators / Linkage / Loader)
    - Appendix B - Bibliography/References

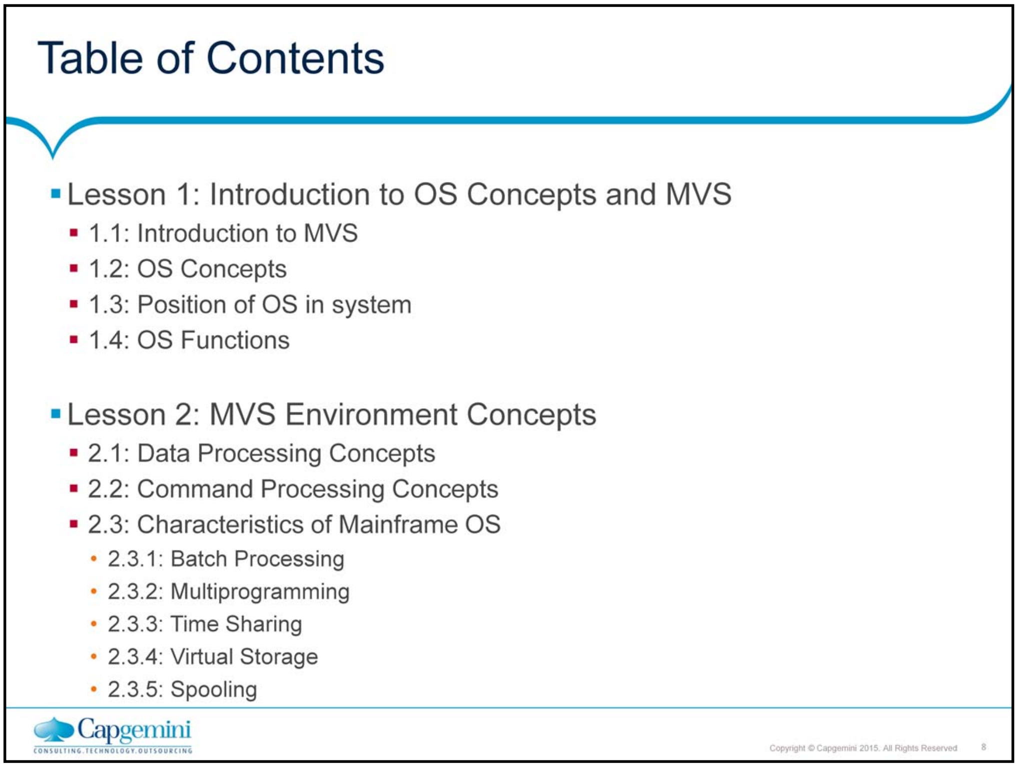


Table of Contents
Lesson 1: Introduction to OS Concepts and MVS
1.1: Introduction to MVS
1.2: OS Concepts
1.3: Position of OS in system
1.4: OS Functions
Lesson 2: MVS Environment Concepts
2.1: Data Processing Concepts
2.2: Command Processing Concepts
2.3: Characteristics of Mainframe OS
2.3.1: Batch Processing
2.3.2: Multiprogramming
2.3.3: Time Sharing
2.3.4: Virtual Storage
2.3.5: Spooling
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
8

Table of Contents

 9

Table of Contents

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

Table of Contents

 11

Table of Contents

 12

Table of Contents

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Table of Contents

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14

References

- Books:
  - The MVS JCL Primer; by Saba Zamir, Chander Ranade
  - IBM Mainframe Handbook
- URLs:
  - http://hansen-family.com/mvs/MVS%20Commands.htm
  - http://ibm.com

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 15


Next Step Courses
JCL
COBOL
VSAM
DB2
CICS
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
16

Multiple Virtual Storage (MVS)

Lesson 1: Introduction to OS Concepts and MVS

Lesson Objectives

- In this lesson, you will learn the following topics:
  - Basic concepts of OS
  - Need and importance of OS
  - Introduction to MVS

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

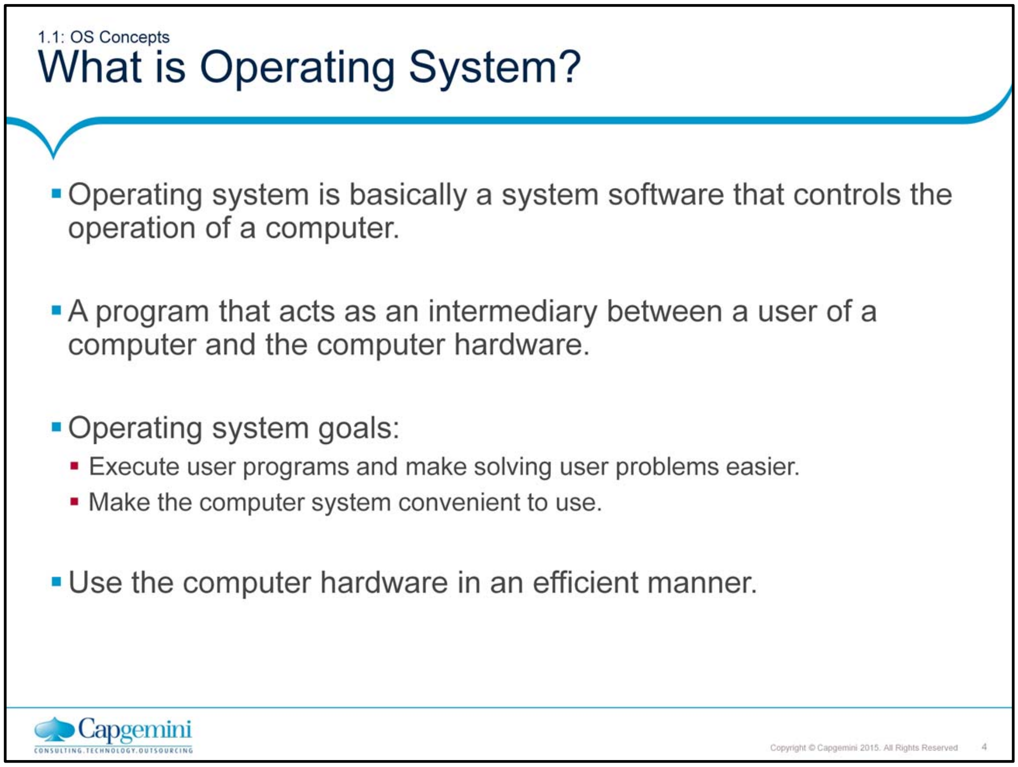
1.1: OS Concepts

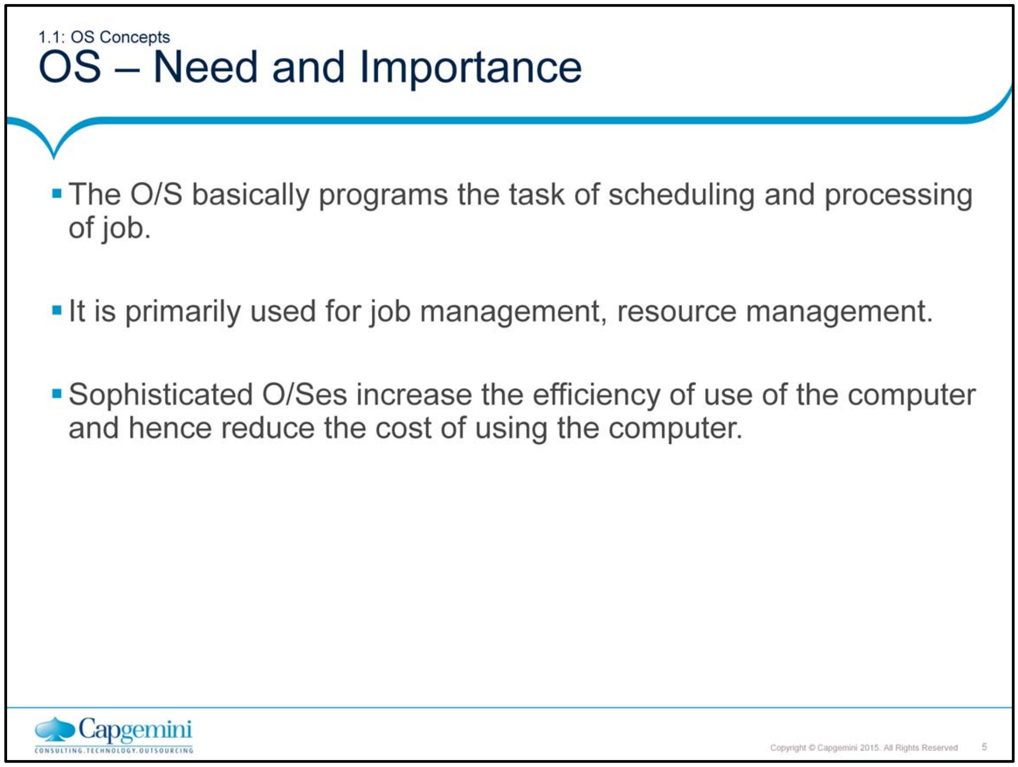
Overview

- OS concepts:
  - What is an Operating System (O/S)
  - Need and importance of O/S
  - Position of O/S
  - Key Functions of O/S
  - About MVS

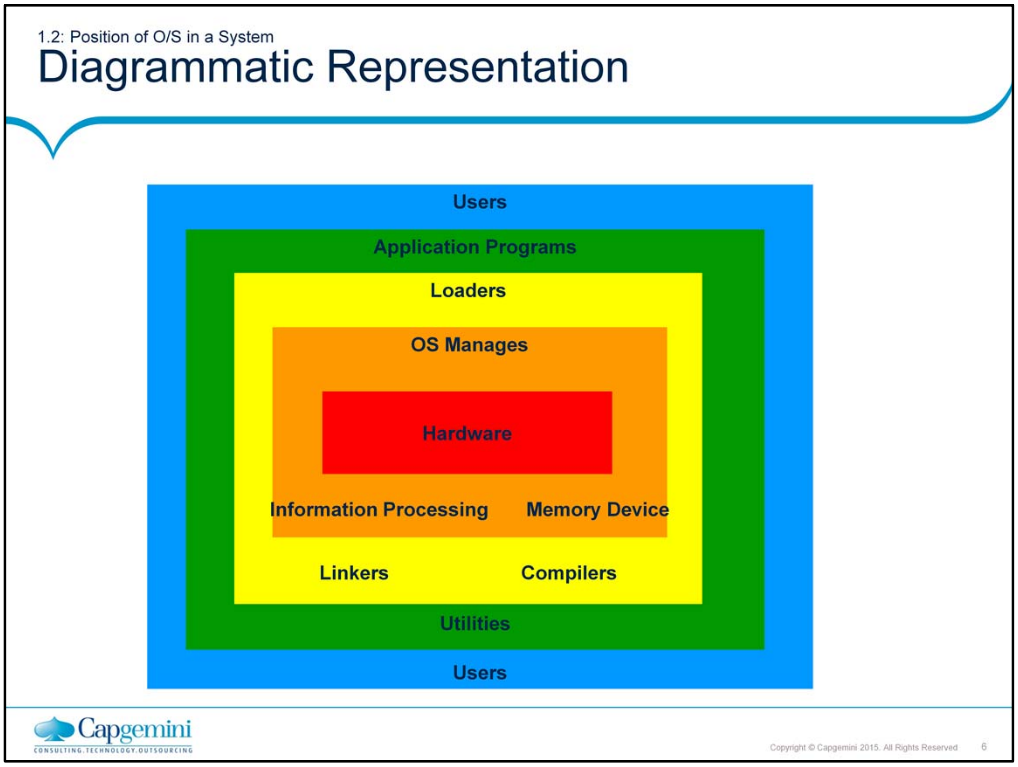
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3


1.1: OS Concepts
What is Operating System?
Operating system is basically a system software that controls the operation of a computer.
A program that acts as an intermediary between a user of a computer and the computer hardware.
Operating system goals:
Execute user programs and make solving user problems easier.
Make the computer system convenient to use.
Use the computer hardware in an efficient manner.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
4



OS Concepts:
An operating system is a resource manager that manages all resources and decides between conflicting requests for efficient and fair resource use.
OS acts as a control program that controls execution of programs to prevent errors and improper use of the computer.

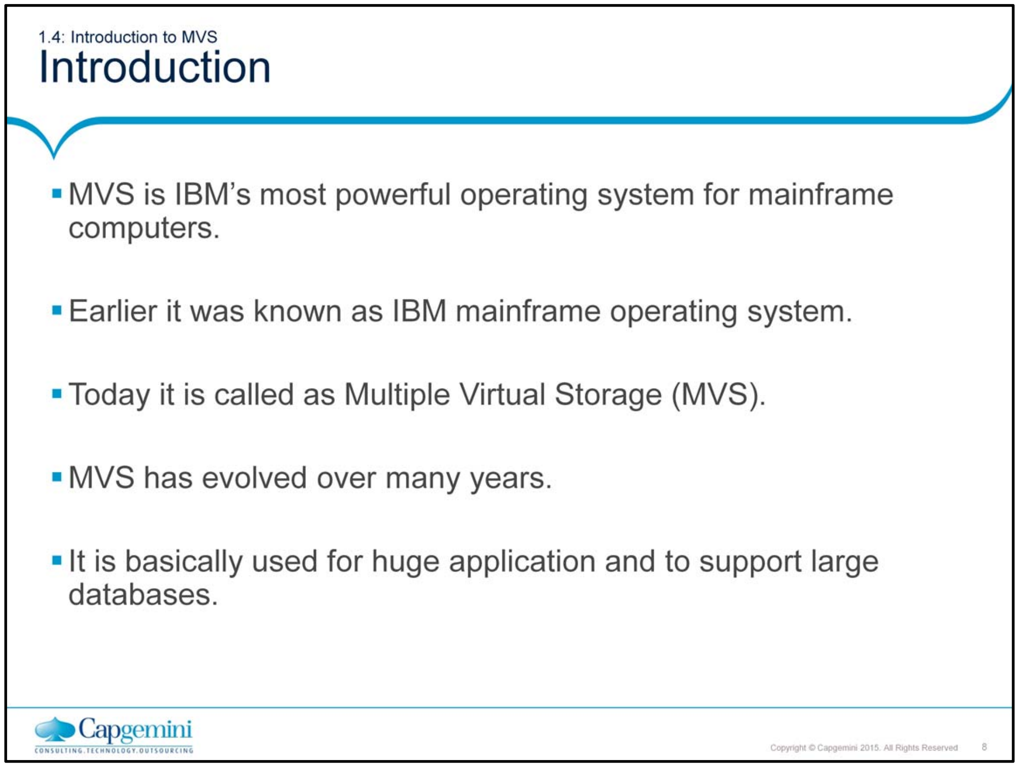


Components of OS:
An OS comprises of the following systems:
Hardware: It provides basic computing resources (CPU, memory, I/O devices).
Operating system: It manages, controls, and coordinates the use of the hardware among the various application programs for the various users using the system programs.
System Programs: It comprises the loader, linkers, and compilers used for handling the application programs.
Applications programs: They define the ways in which the system resources are used to solve the computing problems of the users (compilers, database systems, video games, business programs).
Users: They comprise people, machines, other computers.

1.3: OS Functions

Key Functions

- Following are the key functions of an OS:
  - It keeps track of resources namely memory, I/O devices, processor, job, and information.
  - It decides who gets what resource and how much.
  - It allocates the resources when needed.
  - It reclaims the resources back after it has been used, so that they may be allocated subsequently.



Introduction to MVS:

"Mainframe environment", which by default means "IBM Mainframe" for which you need to have a basic idea of the IBM mainframe operating system. Today, it is know as MVS, which expands to Multiple Virtual Storage. The MVS operating System was created by IBM and is said to be 'propriety' OS.

The MVS operating system has evolved over many years and has adapted to the changing technology and modern day requirements. It has the capacity to support a large number of peripherals like disks, tapes, printers, Network devices, and so on. MVS is designed to work with many hundreds of users working together, located in the same locality or across continents.

The applications on these "Legacy systems" are typically where there is a huge amount of data and a large user base. Since the user base of MVS is very large, a change is not easy to implement. Companies that own these mainframes are typically those that are very big inherently or have to deal with vast amounts of data, which has to be processed fast.

The costs of the Mainframes are very high and the customer base is mostly made up of long-term customers with huge application and large databases to support. Most of these applications are 'Mission Critical' applications. It is therefore imperative that any change to MVS also be backward compatible.

Examples: Banking sector, Insurance Sector, Newspapers, Material and Inventory, Airlines, Credit Card System, Billing, Accounting, Shipping and others.

1.4: Introduction to MVS

About Mainframe System…

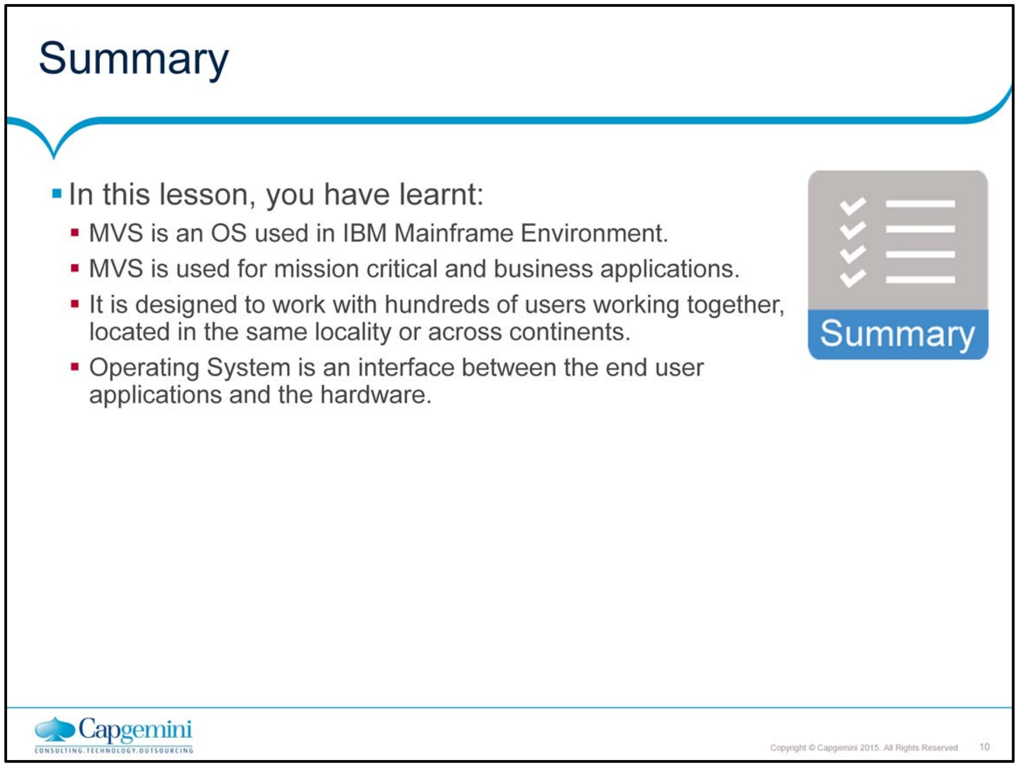
- Capability to handle huge DATA
- High Speed Processing
- Multiple Applications
- Multiple Users
- Reliable
- Meant for Non-Stop Operation
- Modular easily expandable
- Good Connectivity to Non-IBM Systems

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Availability of the Mainframe is really very high hence used in applications where downtime would be costly or catastrophic. They support high-speed memory (Mainframes are measured in millions of integer operations per second (MIPS)). Mainframes deals with massive databases and files. 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


Summary
In this lesson, you have learnt:
MVS is an OS used in IBM Mainframe Environment.
MVS is used for mission critical and business applications.
It is designed to work with hundreds of users working together, located in the same locality or across continents.
Operating System is an interface between the end user applications and the hardware.
Summary
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
10

Review Question: Crossword

1	2. I							3	
							4	O	
	5								
6									
		7							
							8		
				9					

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Clues.
Across:
Type of memory used in MVS
4. OS executes programs are called as _________
5. Early it was called as _______ environment
7. M in MVS
9. One of the resource which OS manages

Below:
2. The proprietary company of MVS
3. Type of System program
6. One of MVS feature
7. OS for IBM
8. Printers, Disk are type of ____ devices

Multiple Virtual Storage (MVS)

Lesson 2: MVS Environment Concepts

Lesson Objectives

- In this lesson, you will learn the following:
  - The manner in which data and commands are processed in general
  - The various MVS characteristics in detail

Note: We will be covering the basic characteristics of Mainframe OS in depth. Before that, let us check how data processing is done.

2.1: Data Processing

Data Processing Concepts

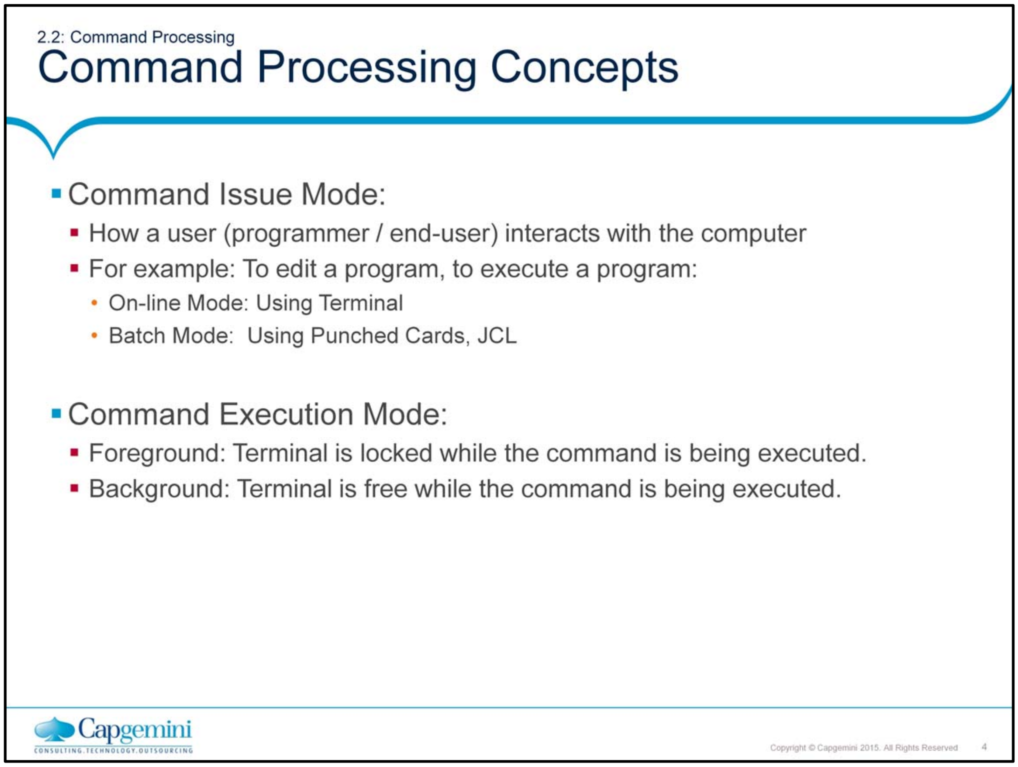
- For executing Business applications, we can have:
  - On-Line Mode / Foreground Mode:
    - It provides an interactive mode for the end user to execute the application programs.
  - Batch Mode / Background Mode:
    - It provides a non-interactive mode for executing the programs.
    - Programs are normally submitted as batch job for execution.
    - Instructions to execute are maintained in a separate command file.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3

Data Processing:
Data Processing Concepts:
Online environment:
End user performs business functions. These application programs work interactively with End user. Execution is done in foreground mode, that is normally using the terminal devices for taking the input or displaying the output. Any transactions made are immediately updated in database to reflect the changes. It is typically used for transaction processing, queries, and master updates functions.

Batch Environment:
Some application programs are executed in background mode in which the business functions are periodically executed automatically. These functions are triggered by end user “as and when” required. The Operations department is responsible for monitoring the execution. A command file is created to execute these functions. This command file may consist of multiple programs / system utilities to execute the job. It is typically used for bulk transaction processing, report printing, periodic processing (for example: invoice generation, payroll calculation).


2.2: Command Processing
Command Processing Concepts
Command Issue Mode:
How a user (programmer / end-user) interacts with the computer
For example: To edit a program, to execute a program:
On-line Mode: Using Terminal
Batch Mode: Using Punched Cards, JCL
Command Execution Mode:
Foreground: Terminal is locked while the command is being executed.
Background: Terminal is free while the command is being executed.
Copyright © Capgemini 2015. All Rights Reserved
4

2.3: Basic Characteristics of Mainframe O/S

Characteristics

- The Mainframe O/S performs the following tasks:
  - Batch Processing
  - Multiprogramming
  - Multiprocessing
  - Time sharing
  - Virtual Storage
  - Spooling

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

2.3: Basic Characteristics of Mainframe O/S

Concept of Batch Processing

- On a Mainframe, batch processing is the normal way of using the computer system and has been for decades.
- Work is processed in units called jobs.
- Job is an execution of one or more programs in pre-defined sequence.
- JCL supplies specifications of a job such as programs to be executed, their sequence, where those are stored, what files are used, and where the output is to be held.
- Application programs are executed in background mode

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Batch Processing:
When batch processing is used, work is processed in units called “jobs”. A job may cause one or more programs to be executed in sequence. The system collectively processes batch jobs. The operating system is responsible for scheduling the jobs according to priority and the resources they require.

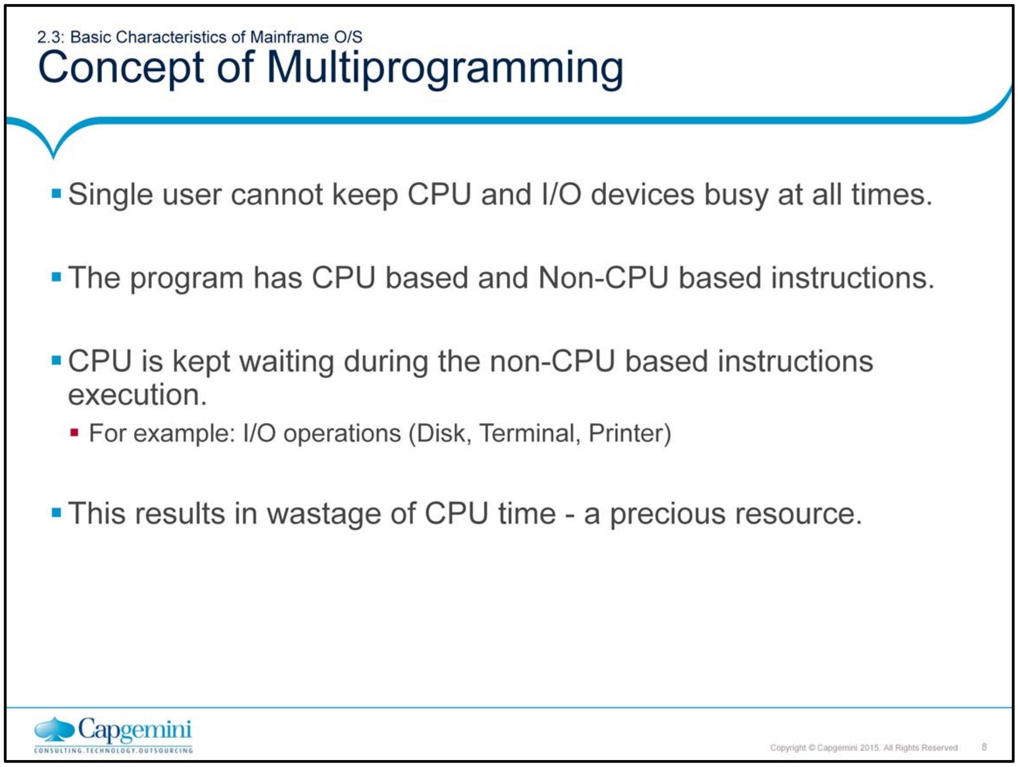
How are jobs executed in batch mode:

When a user submits a job to the system, that job is added to a list of other jobs, perhaps submitted by other users, that are waiting to be executed.
As the processor becomes available, the job scheduler of the O/S selects the next job to be executed from this list .
The job scheduler can make decisions about the order in which jobs should be executed.
Jobs with higher priority can be given preference over jobs with lower priority.

How are jobs executed in batch mode:

5. Priority of a job is specified in the JCL command file that supplies specifications of a job such as programs to be executed, their sequence, where those are stored, what files are used, where the output is to be held, and so on.
6. Once the execution is done, the output is held in the output queue or printed directly.

Example:
A large company would use batch processing to automate their payrolls where in it would find the list of employees, calculate their monthly salary (with tax deductions) and print the corresponding payslips. Batch processing is useful for this purpose since these procedures are repeated for every employee each month.



Multiprogramming:
While executing a program, some processing instructions – like reading data from disk – take much longer time than others.
As a result, most programs are idle, thus waiting a large percentage of the time for the completion of I/O operations. This also leads to wastage of CPU time.
So, how can we go about making efficient CPU utilization?

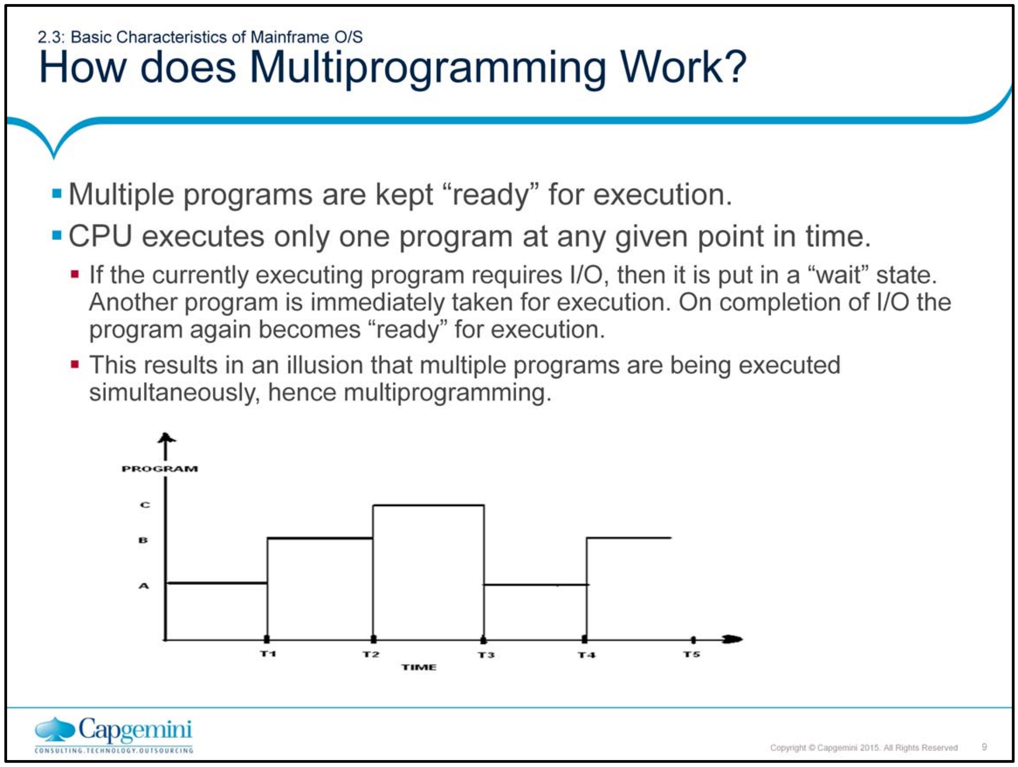
Multiprogramming means OS allows one or more programs to execute concurrently.

Most programs spend most of their time waiting for I/O operations to complete.

So while one program waits for an I/O operation, the CPU can execute instructions for another program.

CPU executes only one program at any given moment of time.

Note: CPU performs concurrent execution of program and not simultaneous execution.



The above slide shows a diagrammatic representation of Multiprogramming. When one program goes for I/O, the O/S takes up second program for execution, and when that goes for I/O, then O/S starts executing another program.

Program doing I/O, goes in “wait” state. After I/O is over, it gets into “ready” state for execution.

The O/S keeps track of which programs are in “ready” state and which programs are in “wait” state.

Note: At any moment, the CPU is executing only one program. It is the concurrent execution of multiple programs.

2.3: Basic Characteristics of Mainframe O/S

Why Multiprogramming?

- Multiprogramming organizes program execution so that CPU always has one job to execute.
- Multiprogramming simply reclaims the CPU during idle periods to let other programs execute.
- Multi-programming results in better and efficient CPU utilization.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

2.3: Basic Characteristics of Mainframe O/S

Multiprogramming Overheads

- Multiprogramming Overheads:
  - Program Queue Management
  - Program Status Management
  - Context Switching during Changeover
  - Multiple programs must be in main memory
  - Management of Common Resource Sharing (e.g. Printer)
  - It is critical to determine optimum level of Multiprogramming to maintain certain service level.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

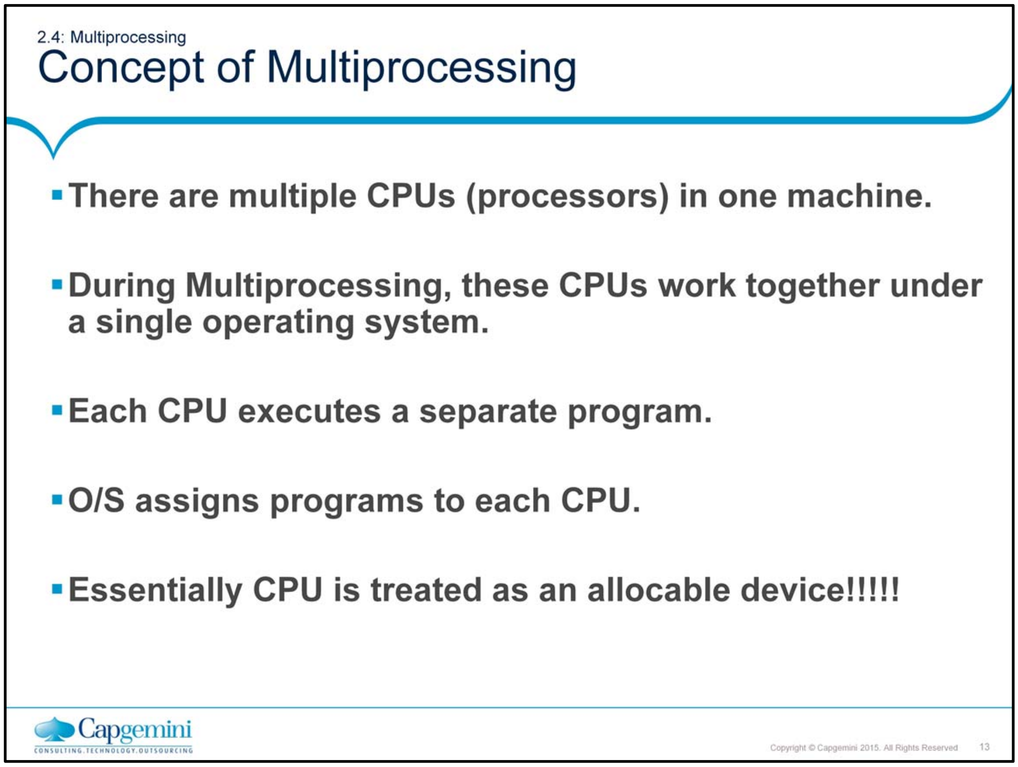
 11

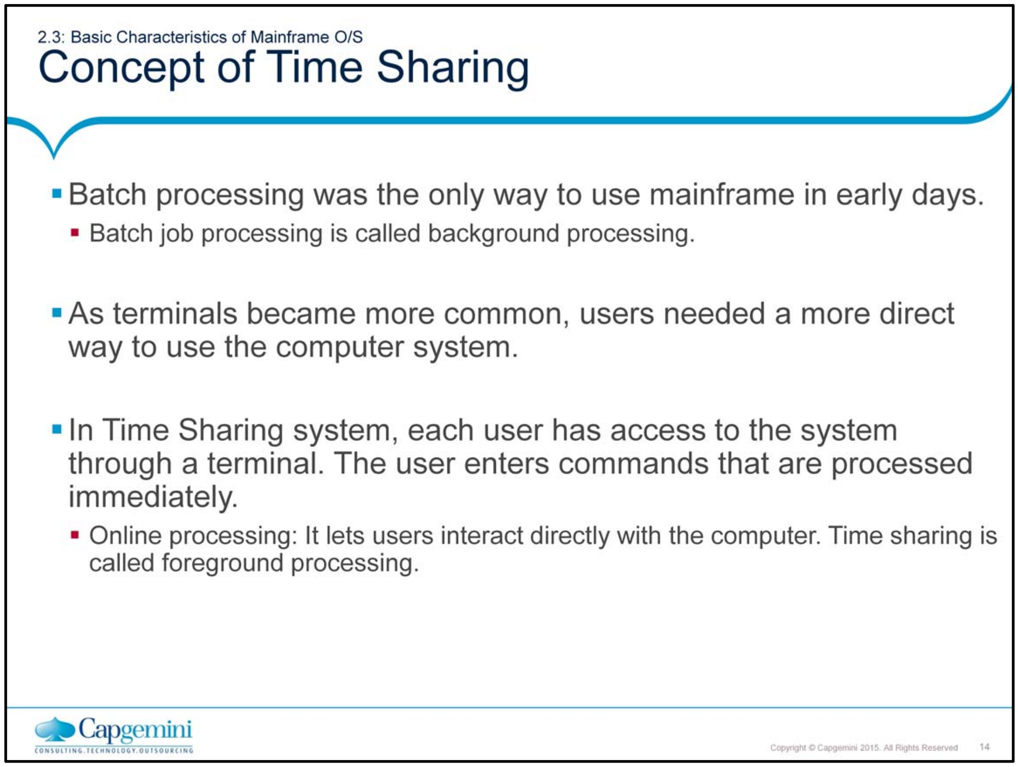
2.3: Basic Characteristics of Mainframe O/S

Relevance of Multiprogramming:

- Multi-programming is applicable even for single user system
- Multi-programming is a must for multi-user system

 12

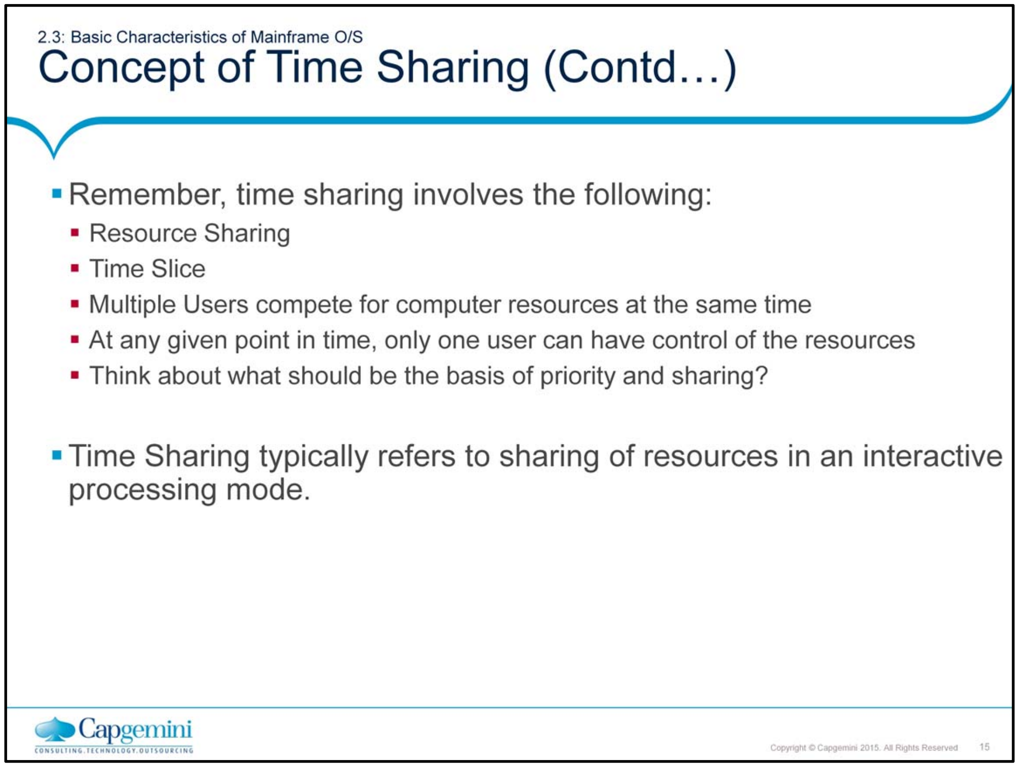

2.4: Multiprocessing
Concept of Multiprocessing
There are multiple CPUs (processors) in one machine.
During Multiprocessing, these CPUs work together under a single operating system.
Each CPU executes a separate program.
O/S assigns programs to each CPU.
Essentially CPU is treated as an allocable device!!!!!
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
13



Time sharing:
Time sharing is a logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, thus creating interactive computing.
In timesharing:

- Response time should be < 1 second.
- Each user has at least one program executing in memory ⇨process
- If several jobs ready to run at the same time ⇨ CPU scheduling
- If processes don't fit in memory, then swapping moves them in and out to run.
- Virtual memory allows execution of processes not completely in memory.

On-line communication between the user and the system is provided. When the operating system finishes the execution of one command, it seeks the next "control statement" not from a card reader, but rather from the user's keyboard.
This involves the CPU allocating individual time slices to a number of users on the computer system. As the number of users increase, the response time for each terminal declines. The speed of the CPU compared to that of the VDU and terminal is so much faster that it gives the user the impression that s/he is the sole user of the system.



Time Sharing:
Time Slice:
Each user is given control of resources for a pre-defined period, that is a time slice. The control is passed on to the next in the queue user at the end of time slice (even if first user's work is incomplete). If the user requires I/O before the time slice is over, then the control is handed over to the next user (since CPU cannot do anything until I/O is complete).
One can decide on the time sharing priority based on usage of one or more algorithm:

- First come first served?
- Priority based?
- Whosoever can grab it - Law of Jungle?
- Equal - Democratically?
- Need based?

Usually the combination of 2 and 4 is used, that is all are equal but some are more equal!!!!!

Priority:
Each user / function is assigned a priority level.
The higher priority users are serviced first in a round robin fashion.
Only if the higher priority users are in "wait" state for I/O completion, then the users in lower priority are serviced.

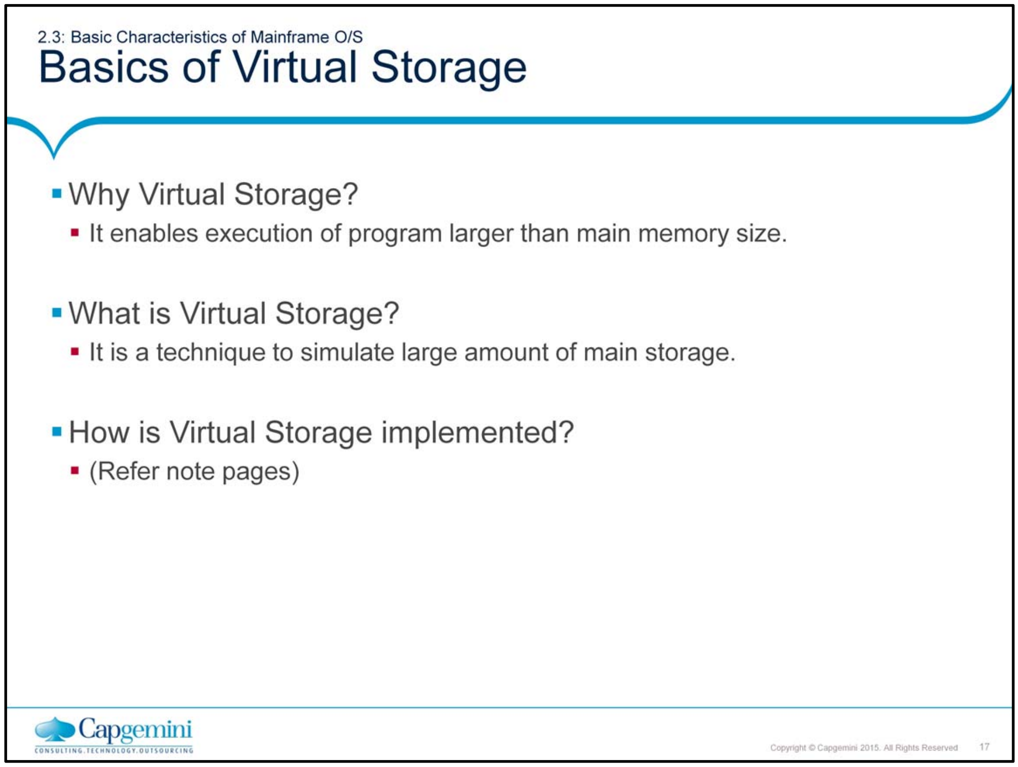
2.3: Basic Characteristics of Mainframe O/S

Concept of Virtual Storage

- Virtual Storage is a technique that lets a large amount of main storage (memory) be simulated by a processor that actually has a smaller amount of real storage installed.
  - For example: A processor that has 4 MB of real storage might use virtual storage to simulate 8 MB of main storage.
  - To do this, the computer uses disk storage as an extension of real storage.
  - From the user's point of view, virtual storage appears to be real storage.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16



Virtual Storage:
Basics of Virtual Storage:
Why Virtual Storage?
It is required to enable execution of programs which are larger than the main memory size.
What is Virtual Storage?
It is a technique to simulate large amount of main storage.
In reality, main storage is much less.
For example: Real main storage is 16MB but virtual storage is 2GB.
How is Virtual Storage implemented?
Program executable code is generated assuming virtual storage size.
At any given moment, only part of the program is loaded in main memory, that is only the current program instruction and the data it accesses needs to be in real storage (memory). Other instructions and data can be placed temporarily on disk and recalled into main storage when needed.
Address translation mechanism is used to map virtual address to actual address.
Operating systems with virtual storage, transfer the instructions and data between real storage and disk, as and when they are needed.
It is feasible because only the instruction currently being executed and the corresponding data need to be in the main storage.

2.3: Basic Characteristics of Mainframe O/S

Advantages and Overheads

- Advantages of Virtual Storage:
  - Main memory can be shared by multiple programs.
  - It enables effective use of the limited main storage.
- Overheads of Virtual Storage:
  - Address mapping
  - Keeping track of what is in memory and what is not
  - Data / Instructions need to be “brought in” main memory as and when required
  - “Remove” from main memory what is not currently required (to make room for instructions of other program)
  - Memory Management

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

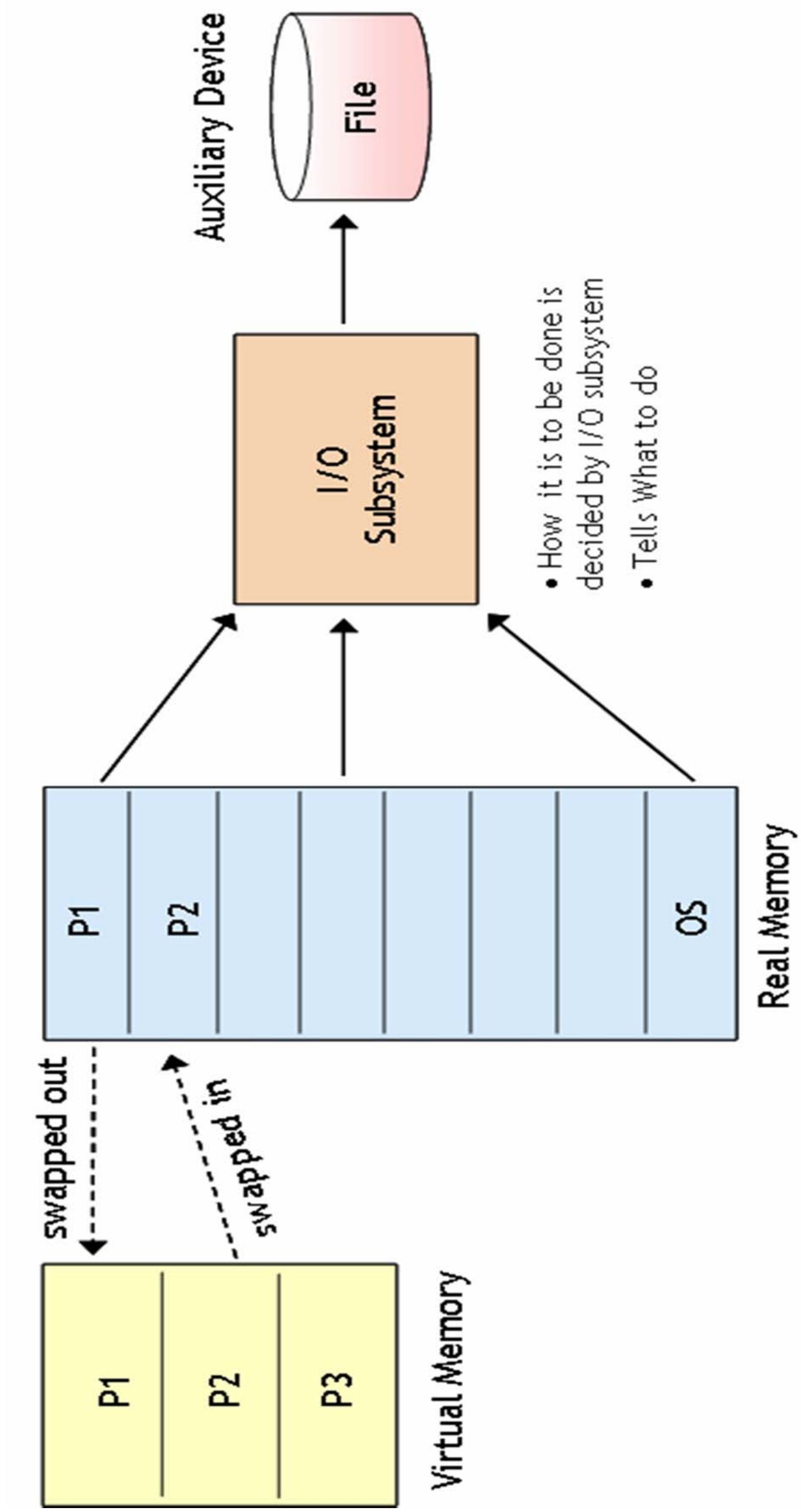
 18

Virtual Storage:
Advantages and Overheads:
Memory Management:
Problem: Anything that is to be executed must be in memory (Memory limitation).
Solution: (1) Place task in real memory; (2) Place task in virtual memory

Real Memory implementation:
- Code and data are in real memory.
- Size of code and data is limited by size of installed memory
- Good performance, low overhead
- Possible wastage of memory

Virtual Memory implementation:
- Based on assumption for a task, not all code and data is needed in real memory at all times.
- It is implemented on a combination of real plus auxiliary storage.
- Operating system takes responsibility of bringing rest part of tasks in real memory when required.
- Advantage: Code and data size are independent of the real memory.

Virtual Storage:



2.3: Basic Characteristics of Mainframe O/S

Concept of Spooling

- In Multiprogramming, common problem is of sharing access to I/O devices among programs.
  - For example: Two programs writing to printer
  - However, if two programs that are executing at the same time, try to write output to a printer, then the output from both programs will be intermixed in the printout and multiprogramming will not hold true.
  - Another way is to share access to input and output devices among the programs that execute together in multiprogramming O/S.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

Spooling:
SPOOL is an acronym for simultaneous peripheral operations on-line.
Spooling refers to putting jobs in a buffer, a special area in memory or on a disk, where a device can access them whenever it is ready.
The most common spooling application is print spooling.
Spooling also lets you place a number of print jobs on a queue instead of waiting for each one to finish before specifying the next one.

2.3: Basic Characteristics of Mainframe O/S

How is Spooling implemented?

- **Output to printer is intercepted and written to a disk that is "spooled"**
- **On completion of program "spooled", output is queued for Printing.**
- **This queue is processed by O/S print routine.**
- **The O/S print routine is multi-programmed along with application programs.**

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21

Spooling:
Spooling is used to provide shared access to printer devices.
Spooling manages printer output for applications by intercepting it and instead directing it to a disk device. When the program is finished, the O/S collects its spooled print output. and directs it to the printer.
In a multiprogramming environment, each program's spooled output is stored separately on disk so that it can be separately printed.
Since disk devices are much faster than printers, programs that produce spooled print output can execute faster than programs that access printers directly.

Summary

- In this lesson, you have learnt:
  - The basics of command processing using:
    - the batch mode
    - the online processing mode
  - Batch processing is the oldest form of processing data. For example: punch card.
  - The mainframe characteristics like time sharing, multiprogramming, spooling, and virtual storage that form the basic for the MVS O/S

Summary

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Review Question

- Question 1: Virtual Storage simulates ___ memory.
  - real
  - virtual
  - disk

- Question 2: Spooling is same as buffering.
  - True / False

- Question 3: In multiprogramming, the ___ is reclaimed during the idle cycles.
  - CPU
  - memory
  - output

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

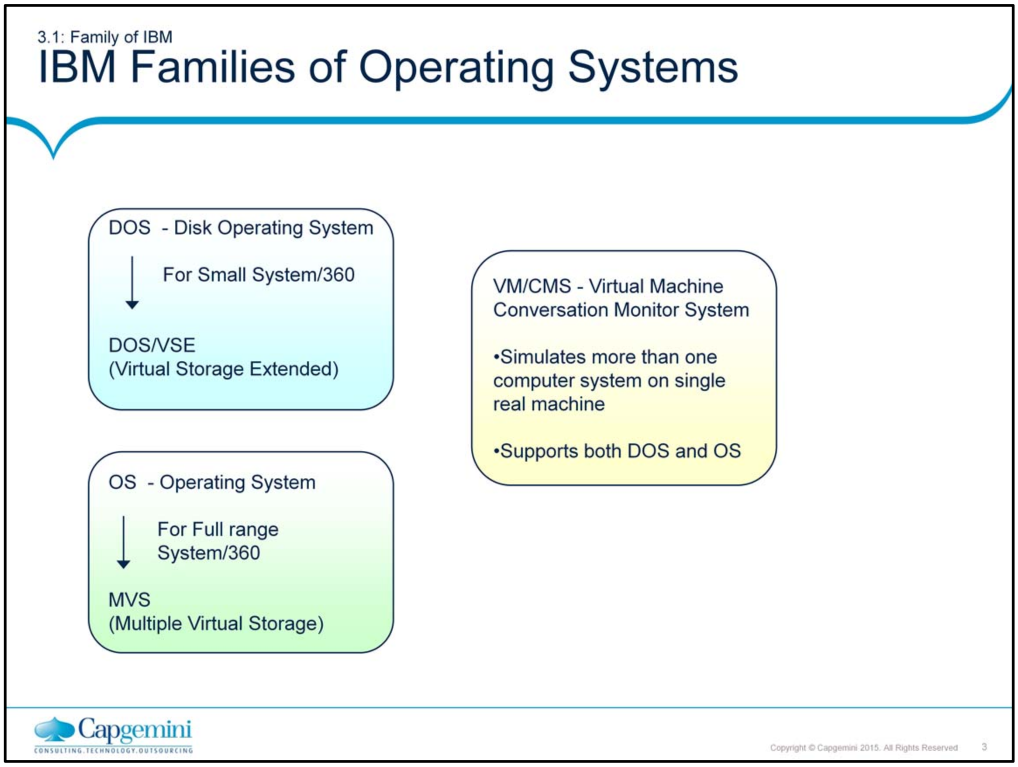
 23

Multiple Virtual Storage (MVS)

Lesson 3: MVS Evolution

Lesson Objectives

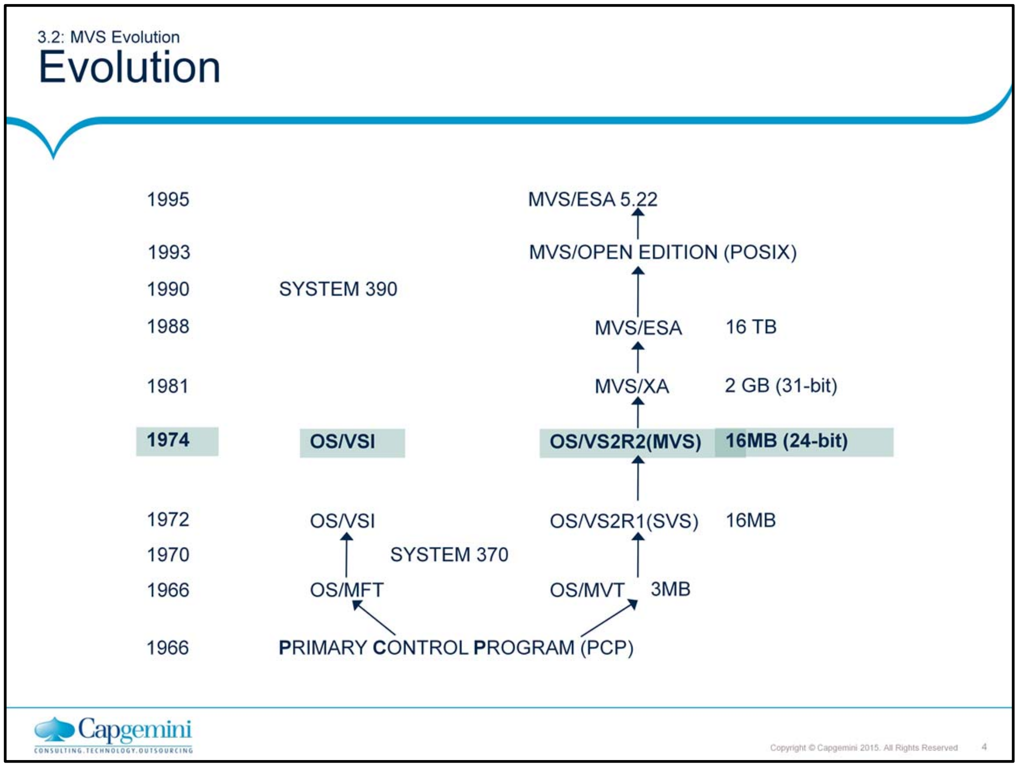
- In this lesson, you will learn the following topics:
  - Evolution of MVS
  - Concepts of PCP, MFT, and MVT



Family of IBM:
Migrating from DOS to OS is a major change.
VM is the third IBM Operating system. It uses multiprogramming and virtual storage techniques to simulate more than one computer system (called as virtual machine) on a single real system.
VM provides a a special operating system called CMS that lets a single terminal user use a virtual machine interactively.
VM is not very popular. Today most of the sites use MVS.

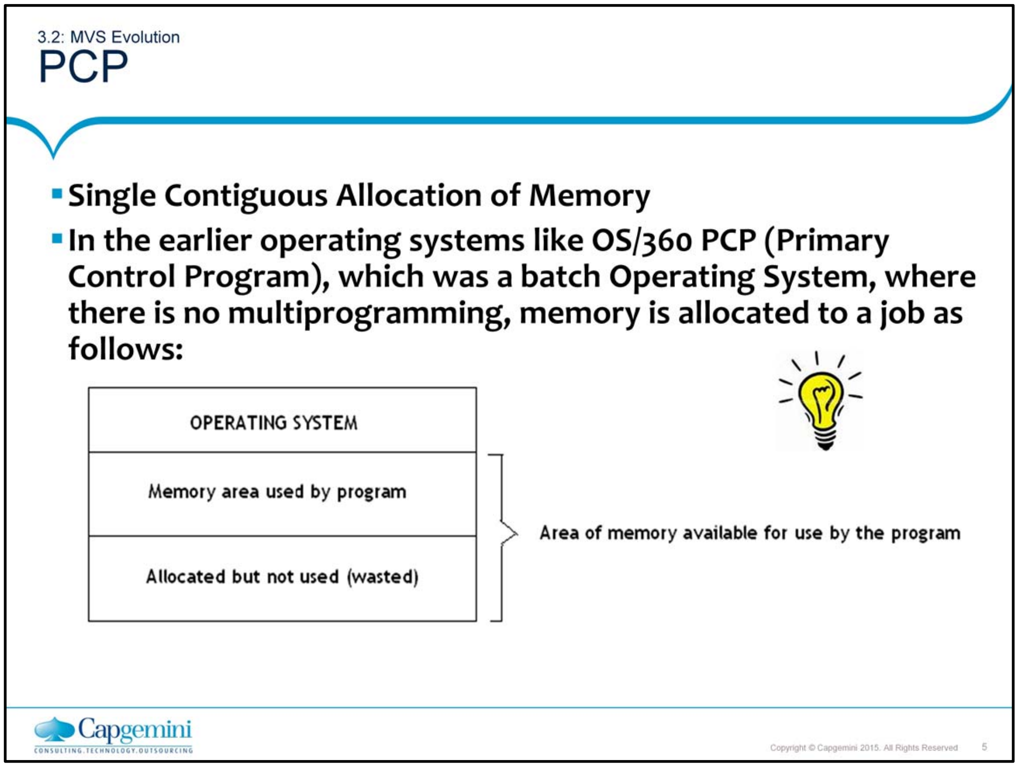
DOS/VSE

DOS stands for Disk Operating System and was designed by IBM for small mainframe computer configurations with limited processing requirements.
This was introduced in the mid-1960s and has evolved significantly and is today called DOS/VSE (Virtual storage extended).
For larger configurations, IBM designed OS.



MVS Evolution:
The above slide lists the chronological order of MVS Evolution.
Following are the full forms of the various acronyms used in the slide.
PCP : Primary Control Program
SVS : Single Virtual Storage
MVS : Multiple Virtual Storage
MFT : Multiprogramming with Fixed number of Task
MVT : Multiprogramming with Variable number of Task
MVS/XA : Multiple Virtual Storage / Extended Architecture
MVS/ESA : MVS Enterprise System Architecture.

Note that originally MVS was called as OS/VS2 (Release 1)



MVS Evolution:

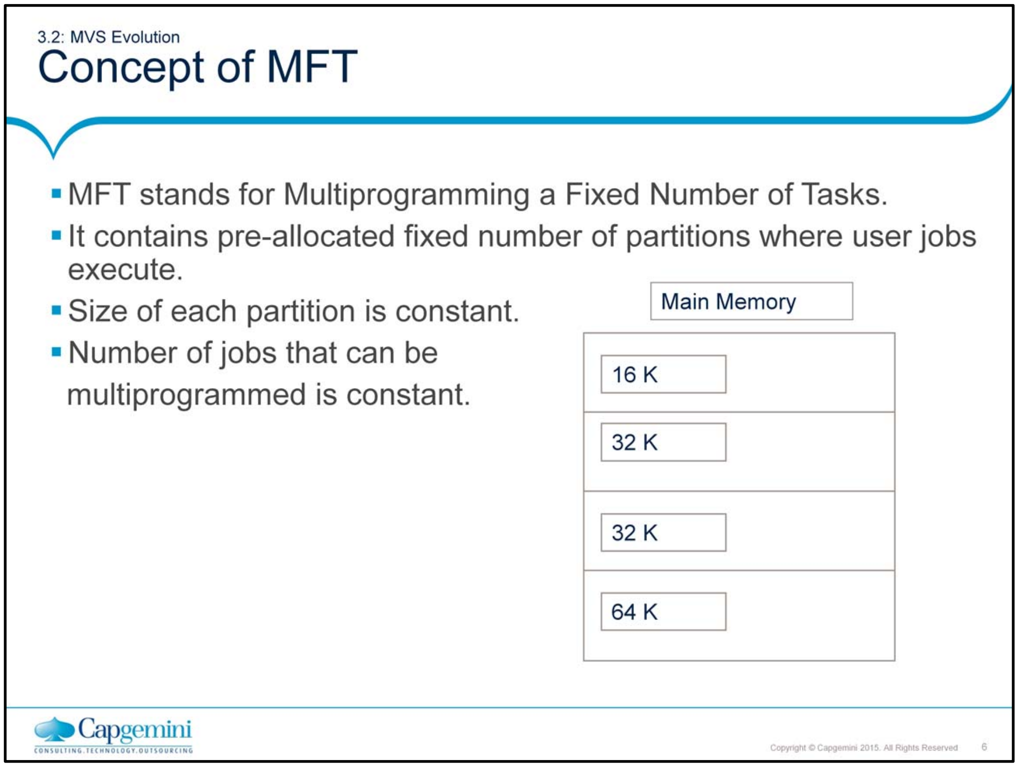
PCP:

In the above diagram, the OS occupies a portion of the memory and the remaining area of the memory is allocated to the user program. If the program is smaller than the remaining memory, then there is some unused space in the memory that goes waste. On the other hand the program that does not fit in memory cannot be executed at all.

Disadvantages:

- The memory is not fully utilized.
- CPU has to wait on the job when an I/O is being performed.
- Some portions of the program in the memory may not be accessed at all.

Hence there is a poor utilization of memory and the CPU and the job size is limited to maximum memory available, after use by the OS.

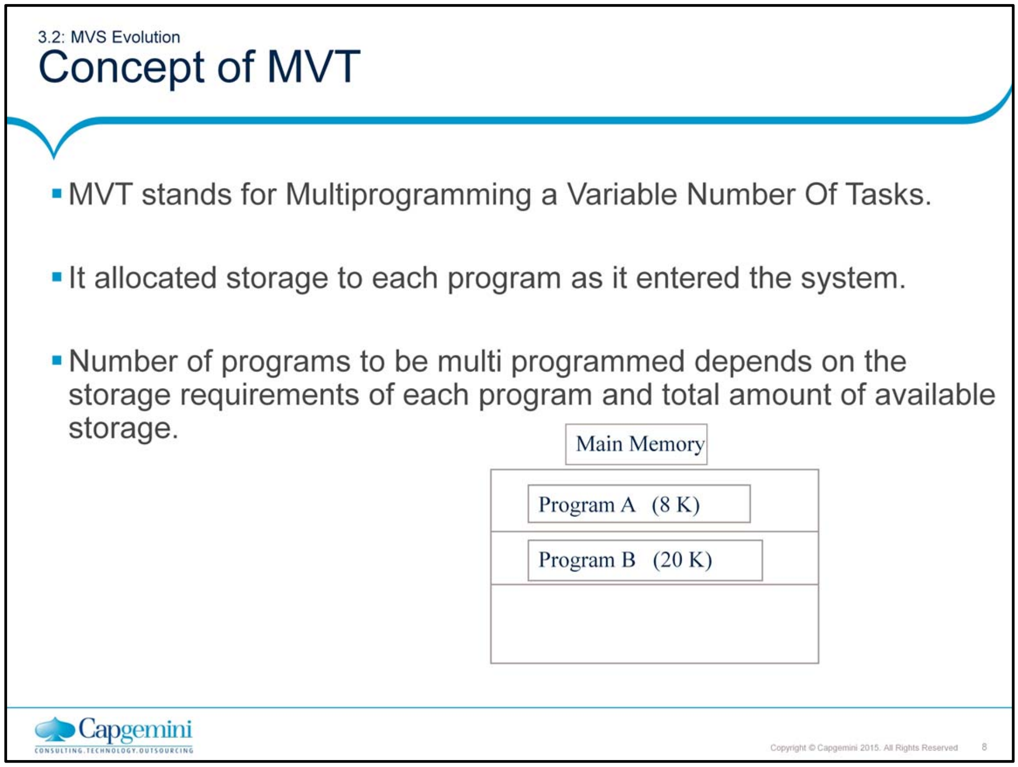


MFT:
This type of memory management is also called Static Partition Specification.
Here the size of each partition is decided at the time of installation of the OS. This remains fixed after the OS is loaded and cannot be changed.
To change the size of the partition, the OS must be reinstalled, during which the changes are incorporated.
A Partition Status Table, which is shown in the figure, is used to keep track of the use of partitions.

Partition No	Size	Status
1	8K	In use
2	32K	In use
3	32K	Not in use
4	120K	Not in use
5	520K	Not in use

MFT (contd.):

- When the user gives a job, s/he has to specify the maximum memory needed to run it.
- A partition of sufficient size is then found and assigned.
- This system is satisfactory when the size of the job is almost equal to the partition size.
- If the size of the job is diverse, then there might be a considerable wastage of memory space.
- For example: When a job requires 33K, then partition 4 is used (see table) resulting in a colossal waste of around 87K. At the time of execution of this 33K job, if another job of 80K is to be submitted, assuming partition 5 (520K size) is already in use, then the new job has to wait (as it can be accommodated only in partition 4 or 5).



MVT:
It becomes important that the appropriate partitions are allocated to the appropriate jobs depending upon the size, so that all partitions are effectively utilized with minimum space wastage. This is called Dynamic Specification. It implies that the partitions are created as and when job is processed.
Here too partition status tables are used. When a job is to be processed, a free contiguous area at least as large as the partition desired must be found. Second, if the area found is large than required, then it must be split into allocated and unallocated portions of the system.

3.2: MVS Evolution

Operating System / Virtual Storage (OS/VS 1)

- OS/VS1 was an enhancement on MFT.
- This is like MFT, but the partitions are created in virtual storage than in main storage.
- This virtual storage is back up auxiliary disk storage. The operating system takes care of bringing in active portions of the program from disk (virtual storage) into Central Storage as needed.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

3.2: MVS Evolution

Operating System / Virtual Storage (OS/VS 2)

- OS/ VS2 single virtual storage replaced MVT. This was an enhancement of MVT, with the addition of virtual storage.
- This freed the operating system, as did MVT, from the problem of locating a suitable partition to run the program.
- OS/VS 2 was used on System/370 machine.

 10

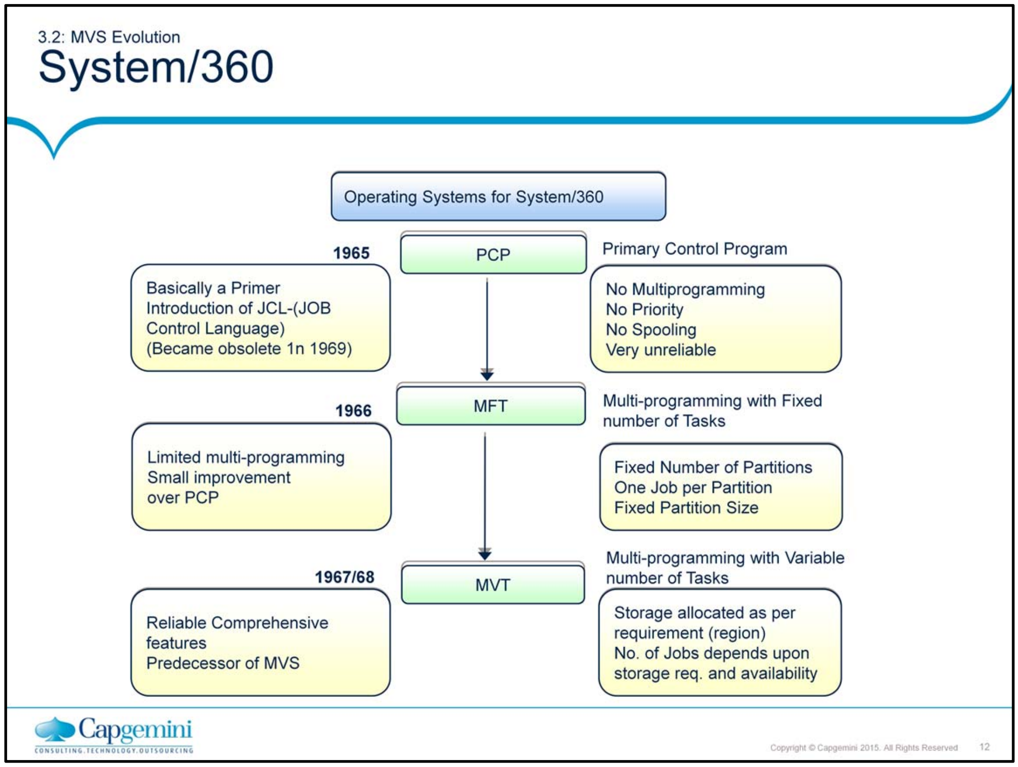
3.2: MVS Evolution

Multiple Virtual Storage (MVS)

- With the growth of OS code, the available area for execution of user programs reduced using single virtual storage.
- This problem was solved with the introduction of Multiple virtual storage (multiple address spaces in virtual storage) in 1974.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11



MVS Evolution:
OS System / 360:
In the earlier operating systems like OS/360 PCP (Primary Control Program), which was a batch Operating System, there is no multiprogramming.
After System/360, System/370 was introduced.
Operating System / Virtual Storage (OS/VS 2):
OS/ VS2 single virtual storage replaced MVT. This was an enhancement of MVT, with the addition of virtual storage.
This freed the operating system, as did MVT, from the problem of locating a suitable partition to run the program.
OS/VS 2 was used on System/370 machine.

Limitation System/360:
Limited and inefficient spooling
No Virtual Storage

These limitations was been overcome by using utilities HASP and ASP.

Utilities to overcome S/360 Limitations

- Two utilities were used to overcome the limitations of S/360 :
  - HSAP – Houston Automatic Spooling Priority
    - Developed unofficially (self initiative) by IBM employees
    - Distributed freely to MVT/MFT users
    - Became very popular
    - Eventually owned and supported by IBM
  - ASP - Attached Support Processor
    - Developed (officially) by IBM and intended for MVT
    - Several mainframes can work together under single O/S (predecessor of multi-processing?)
    - Provided better spooling capability
    - Relatively less takers

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

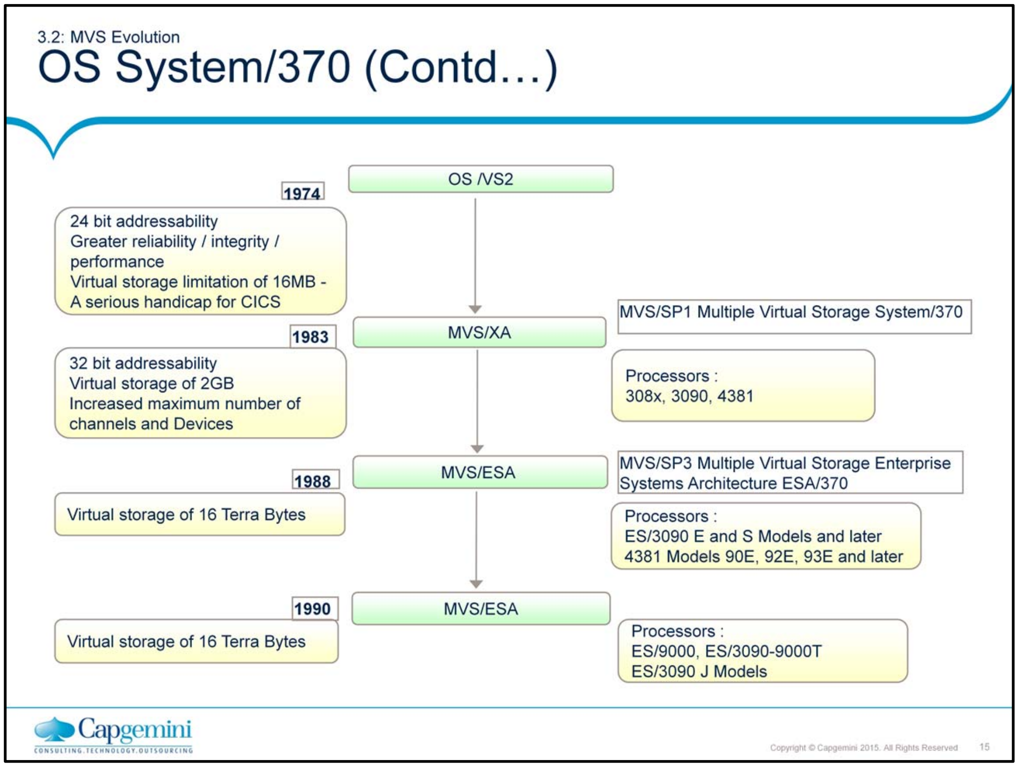
 13

3.2: MVS Evolution

System/370

- System/370 was announced in the early 70s and supported Virtual Storage.
- New Operating Systems OS/VS were introduced.
  - OS/VS1 (Virtual System 1) - adopted from MFT
  - OS/VS2 (Virtual System 2)
  - Version SVS - Single Virtual Storage; adopted from MVT
  - Version MVS - Multiple Virtual Storage
- It was completely rewritten (in 1974).
  - HASP and ASP were migrated to OS/VS2 under the names JES2 and JES3.

Note: Now, MVS and its derivatives are the mainstay of IBM O/S.


3.2: MVS Evolution
OS System/370 (Contd…)
OS /VS2
1974
24 bit addressability
Greater reliability / integrity / performance
Virtual storage limitation of 16MB - A serious handicap for CICS
MVS/XA
1983
MVS/SP1 Multiple Virtual Storage System/370
32 bit addressability
Virtual storage of 2GB
Increased maximum number of channels and Devices
Processors :
308x, 3090, 4381
MVS/ESA
1988
MVS/SP3 Multiple Virtual Storage Enterprise Systems Architecture ESA/370
Virtual storage of 16 Terra Bytes
Processors :
ES/3090 E and S Models and later
4381 Models 90E, 92E, 93E and later
MVS/ESA
1990
Virtual storage of 16 Terra Bytes
Processors :
ES/9000, ES/3090-9000T
ES/3090 J Models
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
15

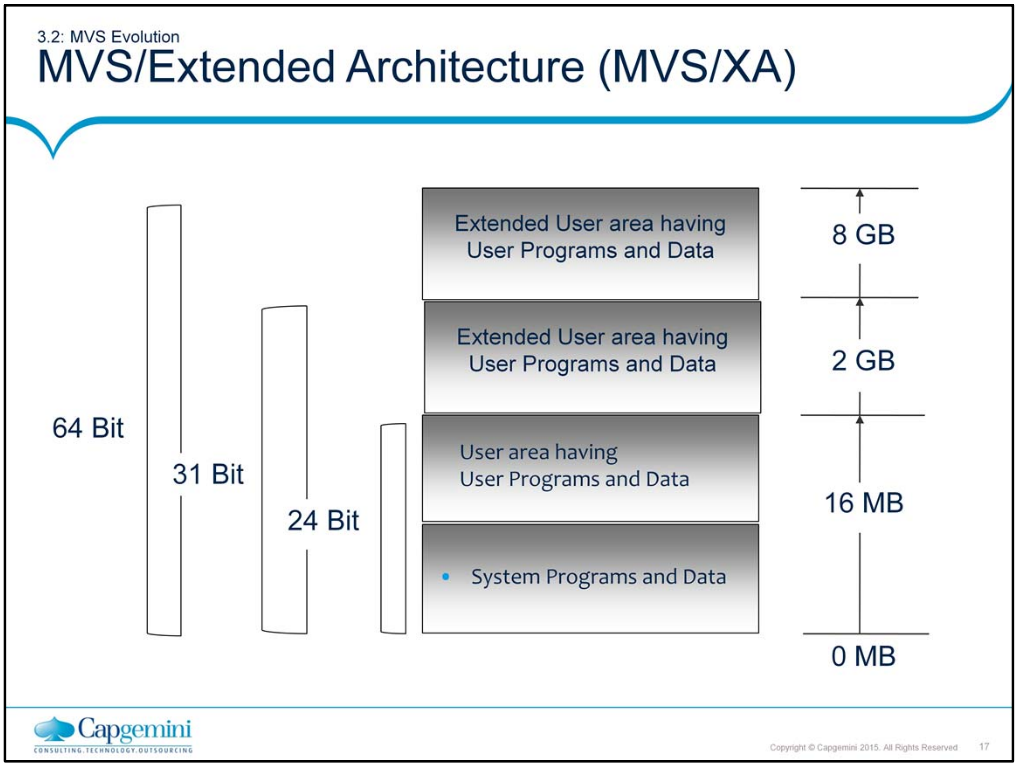
3.2: MVS Evolution

MVS/Extended Architecture (MVS/XA)

- The main improvements in this OS were:
  - The enhancement of address space size from 16 MB to 2 GB by the provision for 31 bit addressing.
  - Old programs compiled in 24 bit mode could be marked to run in 16 MB address space (i.e.. below the 16 MB line).
  - The new programs could call the old programs which run below the 16 MB line.
  - The provision of a Channel subsystem to free the processor from controlling I/O devices.
  - Development of Expanded storage.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16


3.2: MVS Evolution
MVS/Extended Architecture (MVS/XA)
64 Bit
31 Bit
24 Bit
Extended User area having User Programs and Data
Extended User area having User Programs and Data
User area having User Programs and Data
System Programs and Data
8 GB
2 GB
16 MB
0 MB
Capgemini
Copyright © Capgemini 2015. All Rights Reserved
17

3.2: MVS Evolution

MVS/Enterprise Systems Architecture (MVS/ESA)

- In 1988, IBM introduced MVS/ Enterprise Systems Architecture (MVS/ESA) and System 370 / ESA, as a further enhancement to MVS/XA.
- In MVS/ESA, the main additions are:
  - Usage of Data space and Hyperspace.

Multiple Virtual Storage Extended Architecture uses 2gb.
MVS Enterprise System Architecture uses more than 2gb.

3.2: MVS Evolution

OS / 390

- The major enhancement was introduction of System 390 machine supporting OS/390 operating system.
- The channel subsystem which so far used electric cables for data transmission to I/O devices now used the ESCON (Enterprise system connection) channels using fiber-optic cables which carry light pulses rather than electrical pulses. Use of light allows high speed information transfer.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 19

3.2: MVS Evolution

ZOS

- This is the latest development by IBM and it has all the features as earlier evolved
- The major enhancement being its addressability of 64bit which can offer up to 256 GB of address space for each user or subsystem (but is limited to only 8 GB).
- The installed central storage is 8 GB and DASD storage is around 420 GB.

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

Summary

- In this lesson, you have learnt about:
  - The evolution of the Mainframe Operating System
  - The System/360, which consisted of PCP and later came up with MFT and MVT version of OS
  - The System/370, which later came into existence with different version and releases
  - The original name for MVS was OS/VS2.
  - Overview of MVS/XAOS/390, VM, OS/Z

Review Question

- Question 1: PCP is called as ___.
  - Primary Control Program
  - Priority Control Program
  - Post Control Program

- Question 2: Partition status tables are used both in MFT and MVT.
  - True / False

- Question 3: ___ provided better spooling capability.
  - ASP
  - HASP
  - SVS
  - MVT

 22

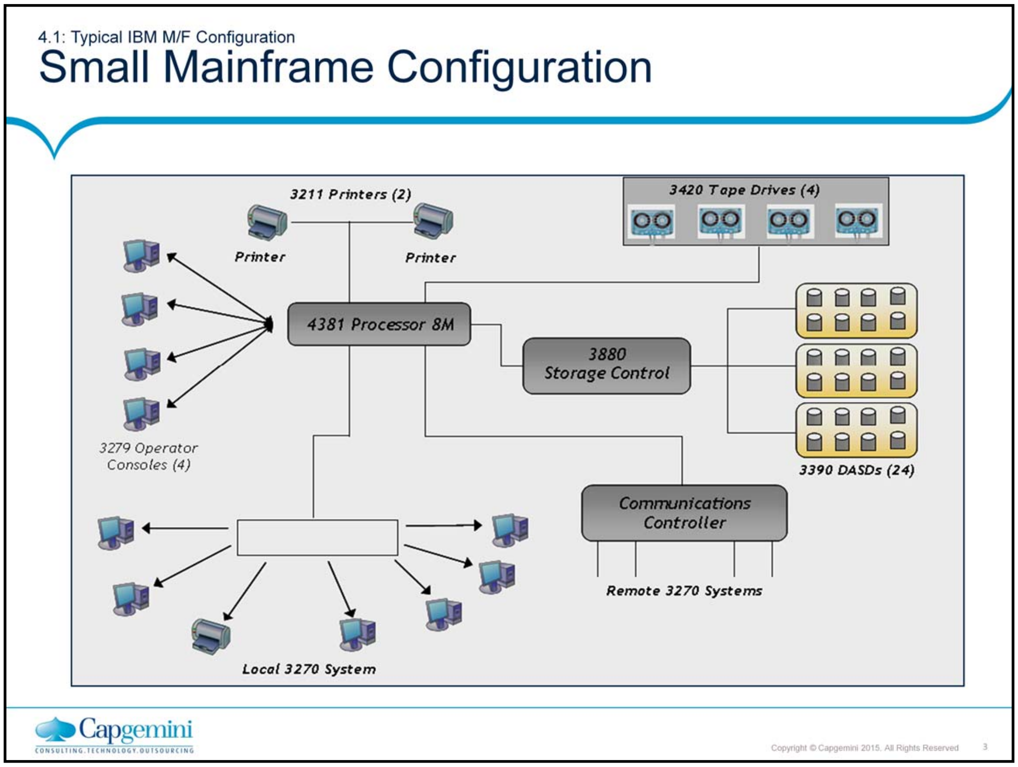
Multiple Virtual Storage

Lesson 4: Typical IBM Mainframe Systems

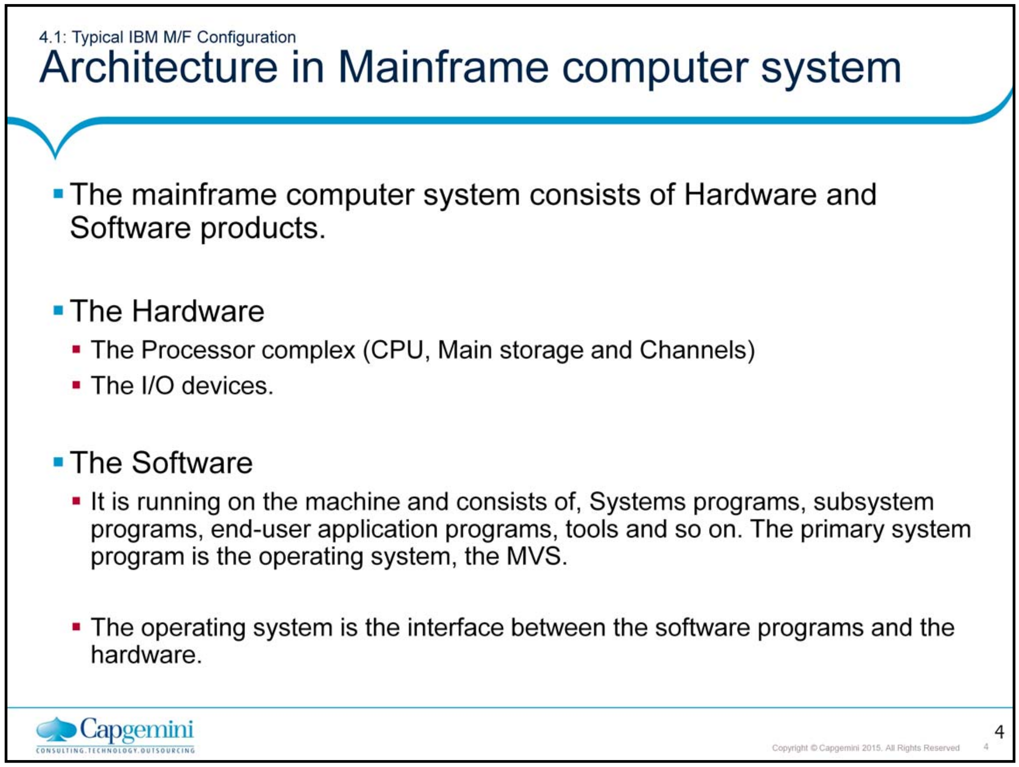
Lesson Objectives

- In this lesson, you will learn the following topics:
  - Typical Mainframe Configuration
  - Various components of IBM Mainframe System
  - Processors
  - Channels Concepts
  - Various I/O Devices used like DASDs and
  - Communication Devices, etc

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



Note: The figure in the above slide shows a small mainframe computer system built around a 4381 processor. It is a Basic system/370 Architecture.



Typical IBM Configuration:

IBM Mainframe Systems:

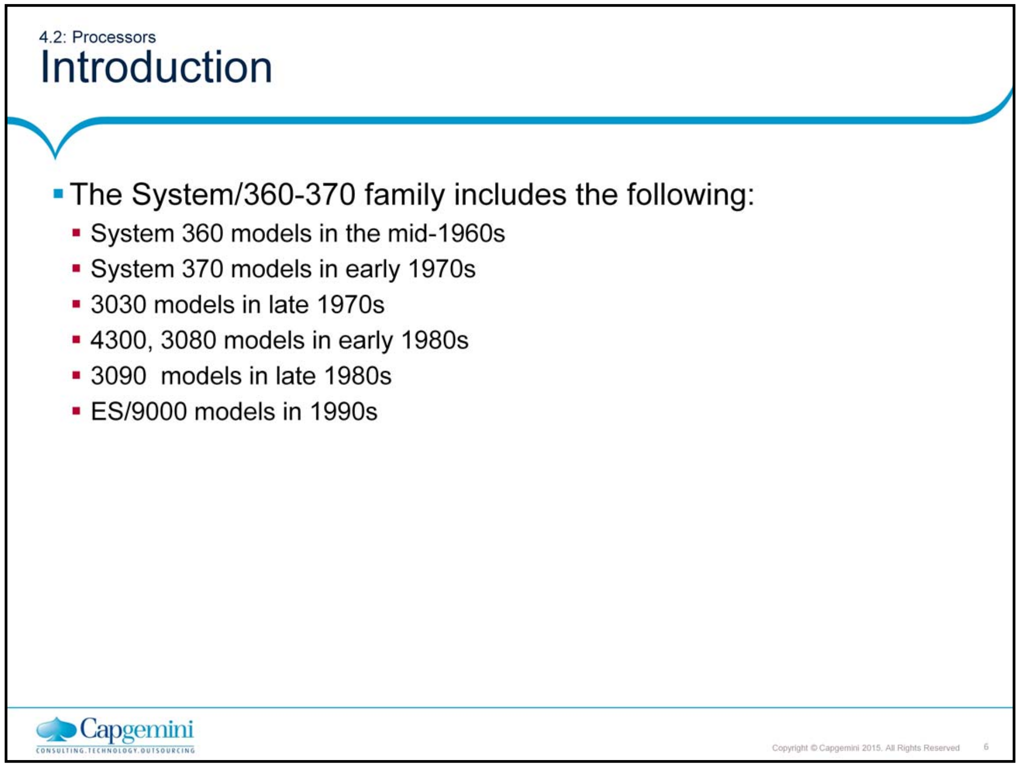
With a variety of IBM Mainframe Processors and I/O devices, we can have a number of Mainframe configurations. So one Mainframe configuration is different from the other.

Let us understand each of these categories.

4.2: Processors

Introduction

- The central components of Mainframe are the processors.
- MVS runs on processors that are members of the System/360-370 family.
- This group of processors has evolved over a period spanning nearly 30 years.



Processors:

Introduction to Processors:

As IBM has developed new models of System 360-370 processors, it has used contemporary technologies to create better, faster, and cheaper machines.

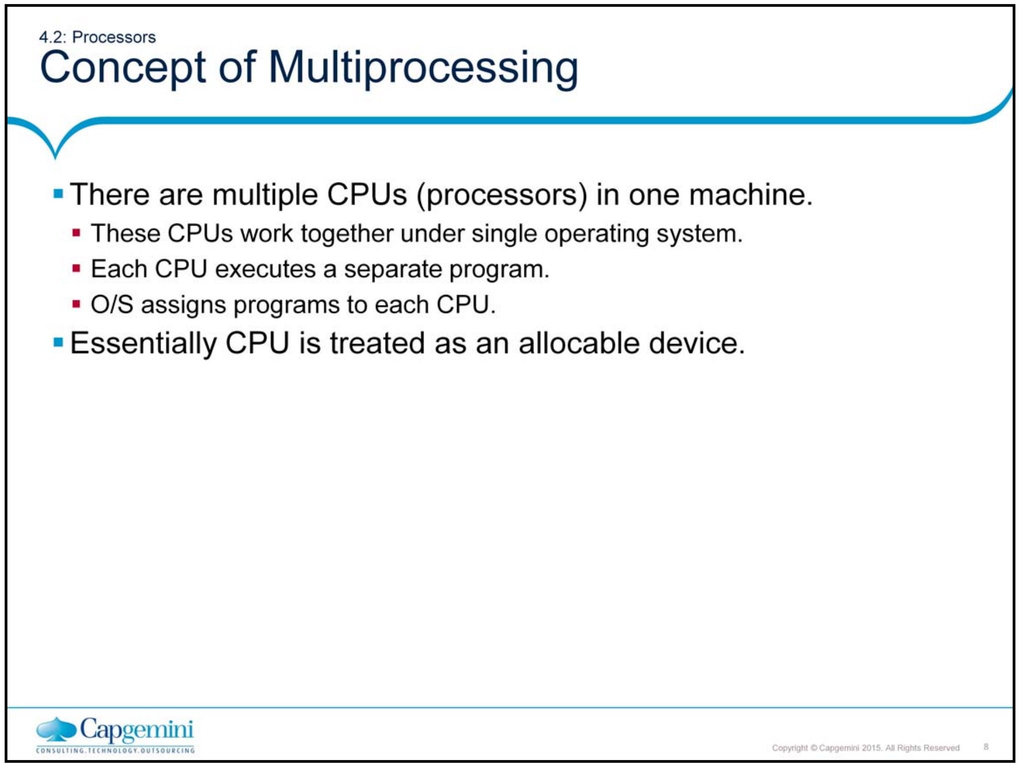
Although the older System/360 and System/370 models are obsolete, the current 4300, 3080, 3090, and ES/9000 processors are still generally called System/370s. This is because IBM has maintained a high degree of compatibility over its 30 year life.

4.2: Processors

PROCESSOR – CPU's – CHANNELS - MEMORY

PROCESSOR CHANNEL	CPU's	MAIN MEM	MAX
4381	1 OR 2	4 – 64MB	18
3084	4	32 – 128MB	48
3090	6	64 – 512MB	128
ES/9000	4 – 8	2048MB	256
S/390 G5/G6	4 – 12	1 – 32GB	256

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



Processors:

Multiprocessing:

In the advanced System/370 family, more than one processor can be included. In multiprocessing system, two or more processors share access to main memory. The operating system determines how each processor is utilized.

Multiprocessing provides two benefits:

> Processing rate is increased because of more number of CPUs will be executing the program instructions.
>
> Systems availability is increased because if one CPU fails the other can take over.

Let us now look at the various I/O devices used in typical mainframe environment.

Firstly let us look at the system/370 architecture.

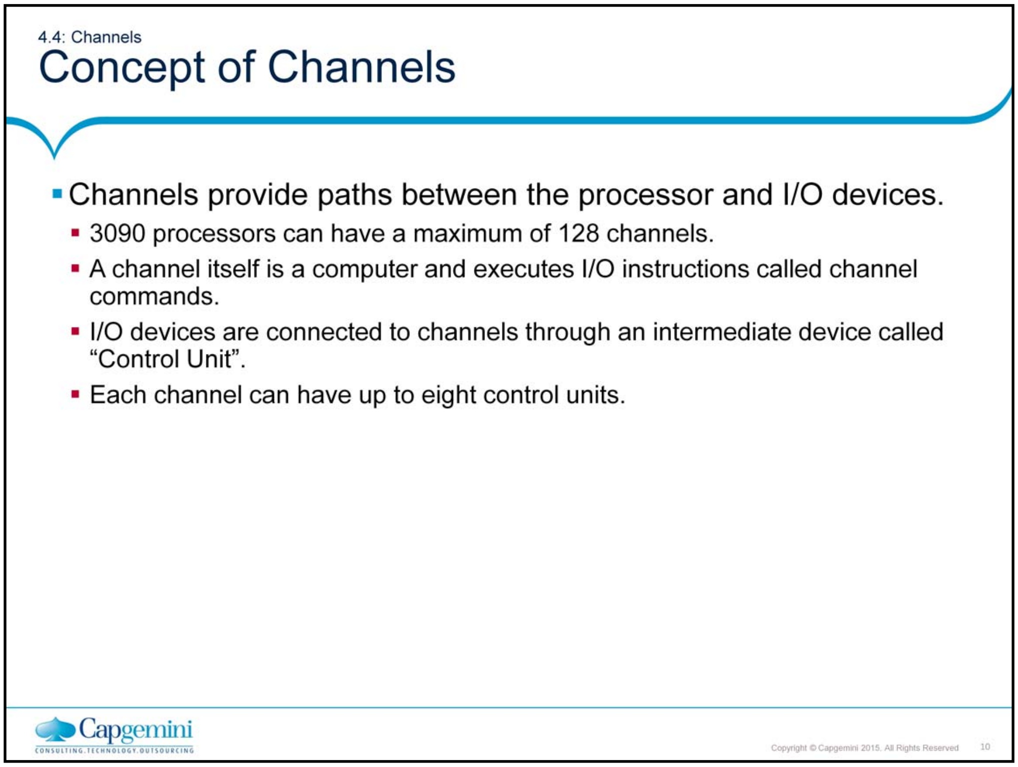
> In multiprocessing, all CPUs share one real storage and channels . The operating system manages the processing ; it assigns work to the first available CPU; in case the CPU failed, work is routed to another CPU.

4.3: Cache Memory

Concept of Cache Memory

- Cache memory refers to high speed memory buffer (faster than main memory).
- It operates between CPU and main memory.
- It is used to store frequently accessed storage locations (instructions).
- It is usually available on all processors.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



Channels:

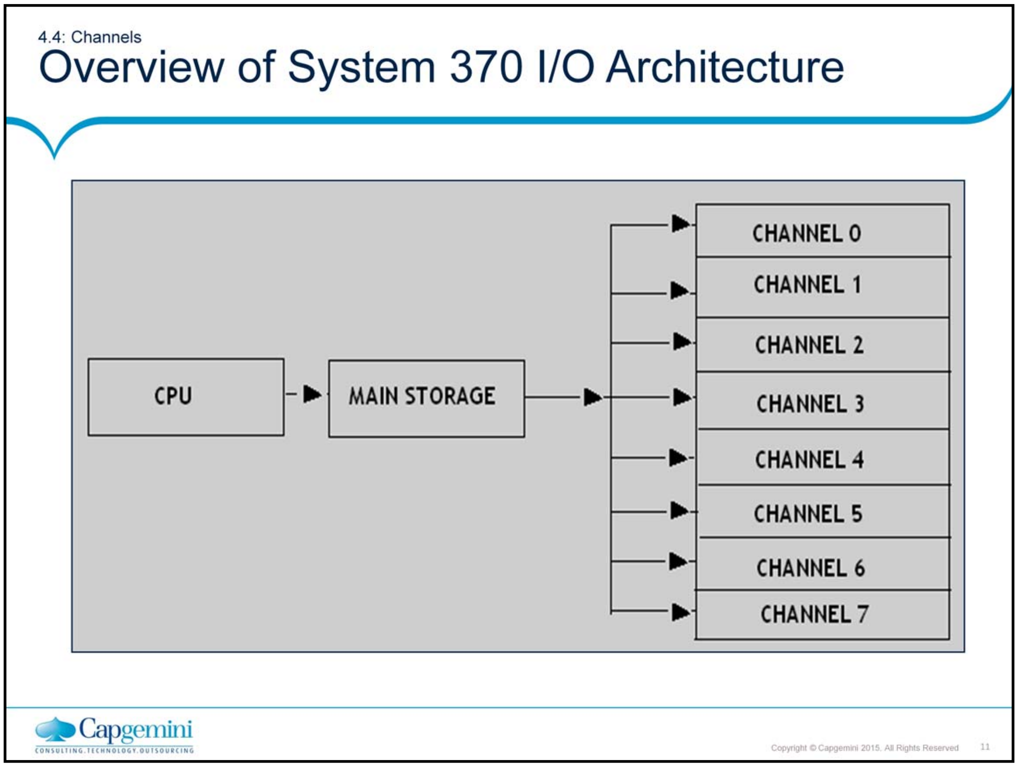
Device Management concept is unique to IBM.

Channels provide access path between CPU and I/O devices (DMA).

> We can connect up to eight control units to one channel.
>
> We can connect up to eight I/O devices to one control unit.

A channel itself is a small computer. It executes I/O instructions called channel commands that control the operations of the I/O devices attached to it. As a result, the channel frees the processor to execute other instructions. Since channel processing overlaps CPU processing, overall system performance is improved.

Let us see an example of a typical I/O architecture for System/370.



Channels:

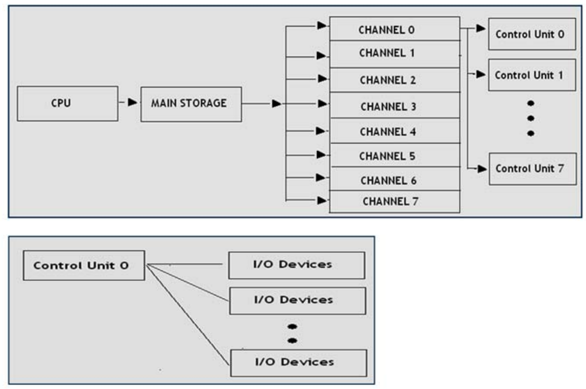
The figure in the above slide shows a typical I/O architecture for system /370.

It consists of CPU attached to main memory and the various channels to which the memory is connected. Channels provide paths between the processor and I/O devices.

4.4: Channels

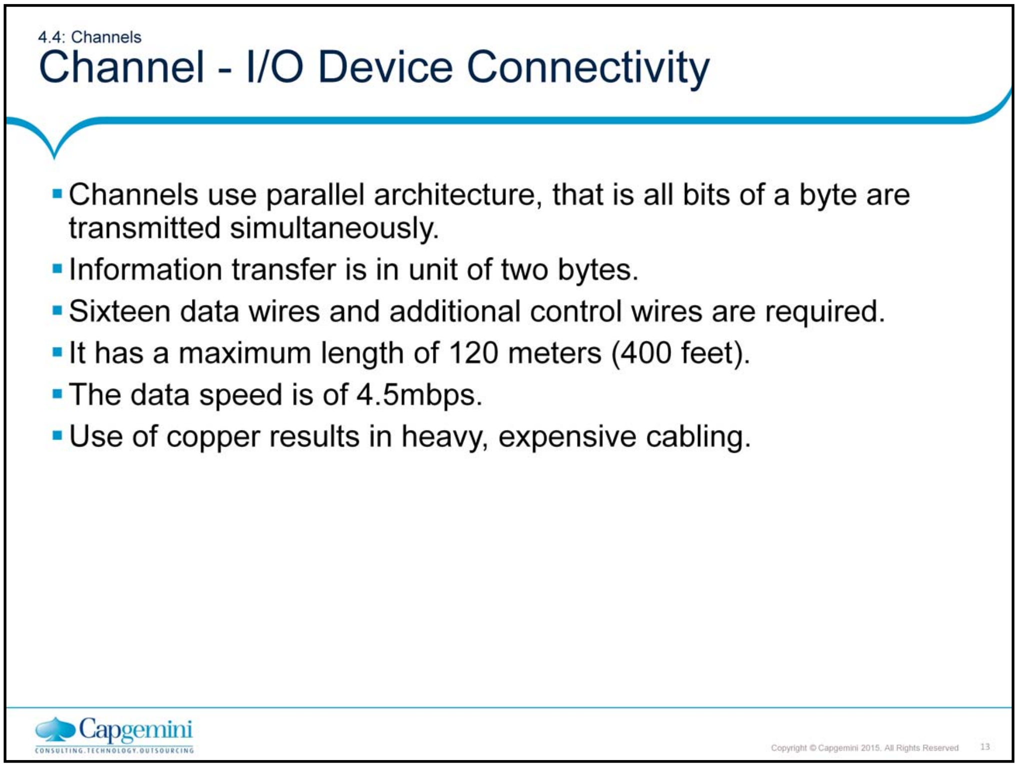
What are Control Units?

- Control units are DASD units. They can be connected to common control unit, called string, which in turn is connected to the “String Controller”.
- String Controller is connected to a channel directly or indirectly. Sometimes a control unit called “Storage Control” connects string controllers to a channel.



 12

Note: DASD is a type of I/O device. To be covered later.



Channels:

Channel – I/O Device Connectivity:

I/O devices are connected to channels using heavy copper cables. They can be of a maximum length of 400 feet.

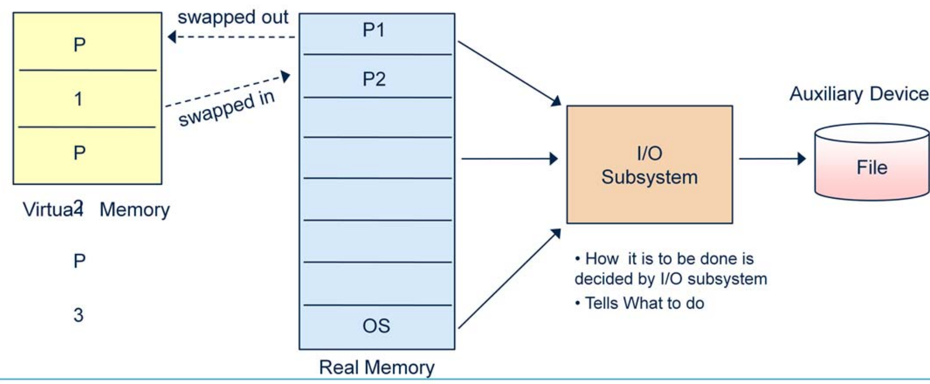
Channels use parallel architecture, that means the cable transmits all bits of a byte simultaneously. To do that, the cable must have a separate wire for each bit – 16 in all (the channel sends two bytes simultaneously) – plus additional wires for control signals.

The parallel channel cable is heavy, bulky, and expensive. A new channel architecture ESCON (Enterprise System Connection), based on fiber optics cable, is 80 times lighter and 50 times less bulky. It extends the 400-feet cable limit to 26 miles. Data transmission is much faster, from 4.5 MB/sec to 17 MB/sec.

4.5: Input-Output Management

Concept of Input-Output Management

- Input-Output Management:
  - **Problem:** Application should not worry about device characteristics. I/O device speed is 100 times slower than CPU.
  - **Solution:** Let all I/O be handled by a specialized system-I/O Subsystem.



4.5: I/O Devices

Types

- We have the following types of I/O devices:
  - Unit record devices
  - Magnetic tape devices
  - Direct Access Storage Devices (disks)
  - Telecommunication devices

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

4.5: I/O Devices

Concept of Unit Record Devices

- Unit record devices consist of:
  - Card Devices (now obsolete): Readers/ Punches/ Reader &Punches
  - Printer
    - Impact Printers - 600 to 2000 LPM
    - Non-Impact Printers - 3800 sub-system, 20,000 LPM
- Each record processed is a single physical unit.
  - For example: card device ~ punch card; printer ~ printed line
- The devices have built-in control units for themselves.
- They are directly attached to the channel.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Unit Record Devices:

Unit Record Devices consist of card devices and printers.

Here, each record processed by the device is a physical unit.

These devices are directly attached to a channel without any control unit, as each unit record device has its built in control units.

In Card devices, each record is a punched card. Example of card devices are Readers, Punches, Reader/Punches which are now obsolete.

In Printers each record is a printed line.

Printers can be of two types:

Impact printers : Impact Printers - 600 to 2000 LPM

Non-impact printers : Laser, 3800 Printing sub-system, 20,000 LPM

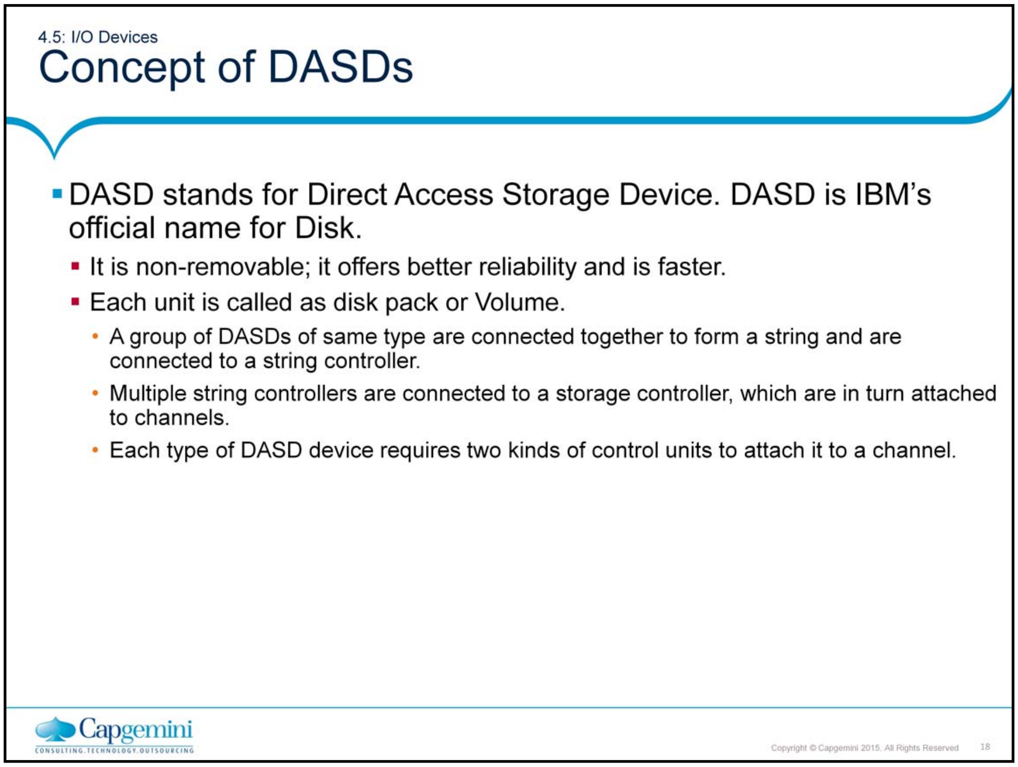
4.5: I/O Devices

Concept of Magnetic Tapes

- Magnetic Tapes have high volume storage.
  - They have sequential processing.
  - They are normally used as back-up devices.
  - They are also used for physical transfer of data.
  - Normally four to eight tape drives are connected to one control unit.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17



Direct Access Storage Device (DASD):

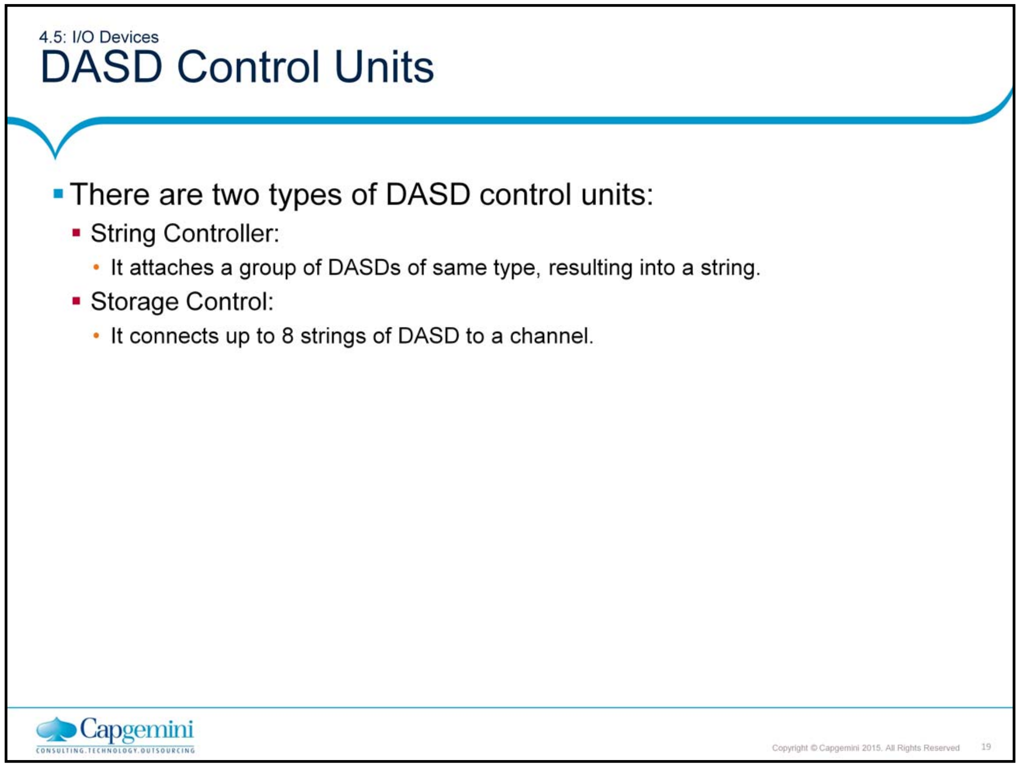
DASD is IBM’s official name for Disk. It is non-removable, offers better reliability, and is faster.

Each unit is called as disk pack or Volume. Volume is a specific unit of storage. For example: Disk pack or DASD, Tape

Each pack has multiple circular surfaces. Each circular surface has multiple concentric tracks. Same track number of all surfaces together constitute a cylinder.

DASD capacity ranges from 100 MB (3330) to 8514MB (3390/9).

A group of DASDs of same type are connected together to form a string and are connected to a string controller. Multiple string controllers are connected to a storage controller. Storage controller is connected to channel.

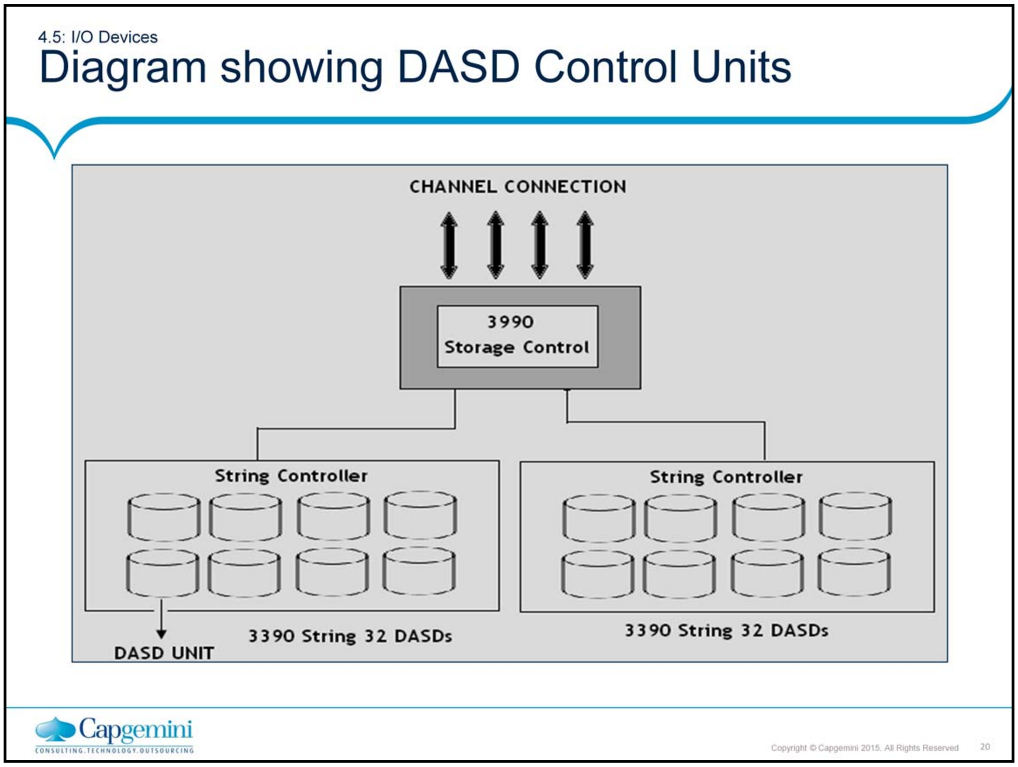


Direct Access Storage Device (DASD):

DASD Control Units:

DASD device requires two kinds of control units attached it to a channel.

> The first, that is string controller, attaches a group of DASDs of same type that results into a string.
>
> The second, that is storage control, connects up to eight strings of DASD to a channel.



Direct Access Storage Device (DASD):

The diagram in the above slide is for a 3390 configuration with two strings attached to a 3990 storage control.

Here, a channel connection is connected to a 3990 storage control (first control unit) which is then connected to the DASD string using the string controller (second controller).

- In 3390 DASD string: up to 32 drives can be connected in one string
- 3990 storage control: attaches two DASD strings
- 3990 storage control supports multiple channel connections to the processor

This enables several simultaneous disk operations to be processed at once.

4.6: Data Communication Network

Concept of Data Communication Network

- Data Communication Network allows local and remote terminals to access the computer systems.
- Components of data communication are as follows:
  - Host Computer
  - Communications Controller
  - Terminal controller
  - Modems and telecommunication lines (telephone line, Satellite Link)
- Data Communication equipment lets an installation create a data communication network (or telecommunication network)
- It lets users at local terminals (terminals at the computer site) and remote terminals (terminals that are not at the computer site) access a computer system.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

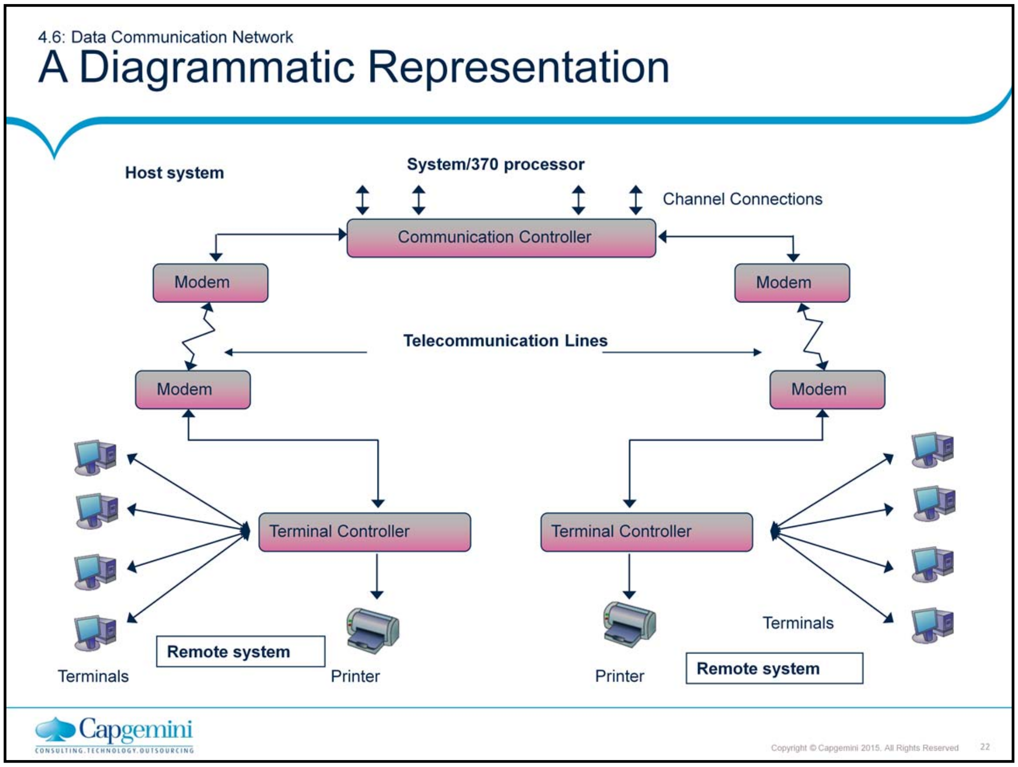
Data Communication Network:

Consider a center of the network in which the host system is a system/370 processor.

The control unit that attaches to the host system’s channels is called a communication controller. It manages the communications functions necessary to connect remote terminal systems via modems and telecommunication lines.

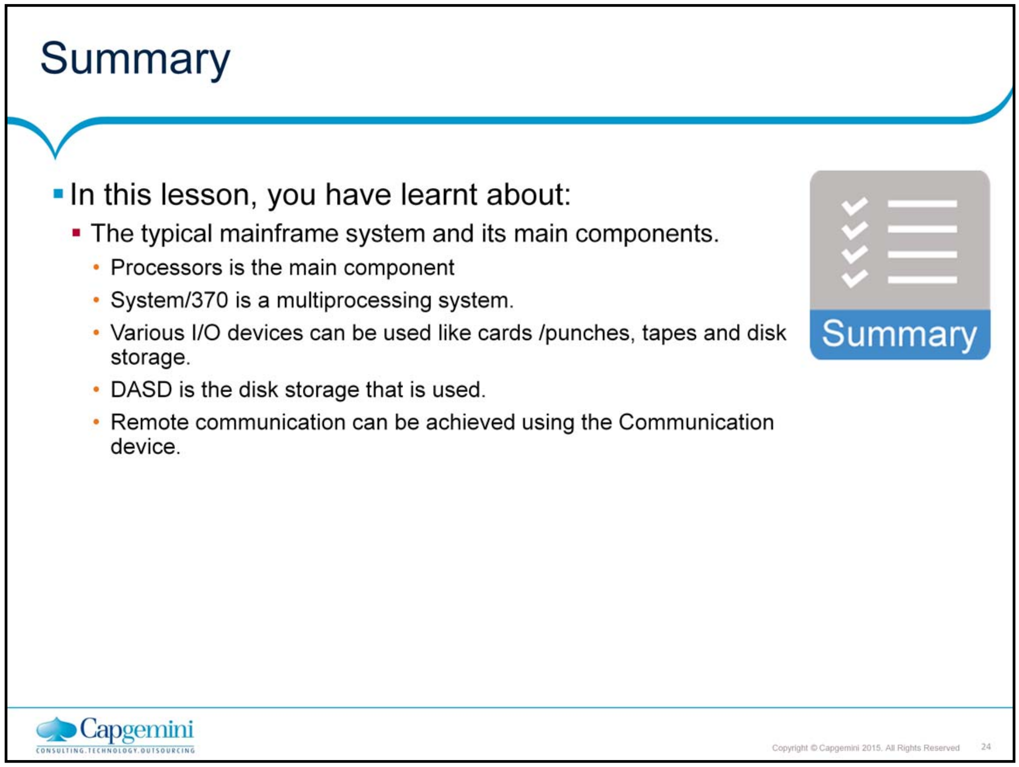
A modem is a device that translates digital signals from the computer equipment at the sending end into audio signal that are transmitted over a telecommunication line. The telecommunication line can be a telephone line, a satellite link, or some other type of connection.

At the receiving end of the line, another modem converts those audio signals back into digital signals.


4.6: Data Communication Network
A Diagrammatic Representation
Host system
System/370 processor
Channel Connections
Communication Controller
Modem
Modem
Telecommunication Lines
Modem
Modem
Terminal Controller
Terminal Controller
Terminals
Remote system
Printer
Printer
Terminals
Remote system
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
22

<u>Data Communication Network</u>:

- The diagram on the earlier slide shows the various components of data communication network.
- Consider the example of a data communication for a system/370 processor. Here the components of data communication will be:
  - Host Computer: a System/370 processor
  - Communications Controller: attached to the system /370 process via channel connection
  - Devices (terminals and printers): connected to the terminal controller (also known as cluster controller)
  - Terminal Controller: connected to communications controller. Terminal Controller managing Local terminals / printers can be connected directly to the channel.
  - Modems and telecommunication lines (telephone line, Satellite Link)
  - Remote terminals / printers: connected to terminal controller (at local site)
- Terminal controller is connected to modem. Modem is connected to telecommunications line. At the receiving end telecommunications line is connected to modem. Modem is connected to communication controller.
- A 3270 Information Display System consists of the following:
  - Sub-system of terminals, printers, and controllers connected to Host computer:
    - Locally through communications controller or directly to channel
    - Remotely through communications controller, modem, and telecommunications line
- A typical 3270 terminal controller (3274) controls up to 32 terminals / printers.
- Emulator programs (Shine Link, Erma Link) allow computers (typically PCs) to mimic 3270 devices.
- These are useful since they allow upload / download of data between MF and PC.


Summary
In this lesson, you have learnt about:
The typical mainframe system and its main components.
Processors is the main component
System/370 is a multiprocessing system.
Various I/O devices can be used like cards /punches, tapes and disk storage.
DASD is the disk storage that is used.
Remote communication can be achieved using the Communication device.
Summary
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
24

Review Question

- Question 1: ___ is a specific unit of storage used in DASDs.
  - Option 1: Volume
  - Option 2: Memory
  - Option 3: Cache
- Question 2: DASD does not require any control unit to connect to processor.
  - True / False
- Question 3: The terminal controllers are attached to the ___ via modems.
  - Option 1: Control Unit
  - Option 2: Communication Controller
  - Option 3: Terminals

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Multiple Virtual Storage

Lesson 5: MVS Concepts and Terminology

Lesson Objectives

- **In this lesson, you will learn:**
  - MVS concepts
  - The concepts of
    - Address Space
    - Paging
    - Swapping

5.1: MVS Concepts

Overview

- **Two main component of MVS are:**
  - Virtual Storage
  - Multiprogramming
- **In MVS, the concepts of Virtual storage and Multiprogramming are closely related.**
- **In a way, they refer to the same functionality in MVS.**
- **Let us see each of these components and understand what is MVS.**

5.2: Virtual Storage

Concept of Virtual Storage

- **Virtual storage is a facility that simulates a large amount of main storage by treating DASDs storage as an extension of real storage.**
- **In other words, when virtual storage is used, the processor appears to have more storage than it actually does.**

5.3: Addresses Spaces

Concept of Address Space

- To search information in a location, an address is required that indicates storage location.
  - An address space is a complete range of addresses that can be accessed by the computer.
  - The number of digits allowed to represent an address limits the maximum size of a computer's address space.
  - For example: Suppose a computer records its addresses using six decimal digits. Then such a computer can access storage with addresses from 0 to 999,999.
- Main storage consists of millions of individual storage locations, each of which can store one character or byte of information.
  - To refer to a particular location, we can use an address, which indicates the storage location's offset from the beginning of memory.
  - The first byte of storage is at address 0, the second byte of storage is at address 1, and so on.
  - Each successive byte of main storage has an address that is one greater than the previous byte of storage.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 5

An address space is simply the complete range of addresses - and as a result, the number of storage locations - that can be accessed by the computer.

5.3: Addresses Spaces

Concept of Address Space (contd.)

- The original System/370 processors used 24-bit binary numbers to represent an address.
  - Since the largest number that can be represented in 24 bits is about 16 million, an address space on a System/370 cannot contain more than 16M bytes of storage.
- 370-XA processors, in XA mode, operate using 31-bit addresses.
  - So the largest address space that can be used is 2 GB.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

5.3: Addresses Spaces

Concept of Address Space(contd.)

- One way to think of Virtual Storage is that it lets the computer push its address space to the maximum capacity allowed by the address format, even if the amount of real storage installed on the processor is less than the maximum capacity of the address format.
- So, in 370-mode, virtual storage can simulate a 16 MB address space, even if only 4 MB or 8 MB of real storage is actually installed.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

5.4: MVS

MVS Address Space

- In MVS, the concept of virtual storage is taken one step further.
  - MVS not only simulates more storage, but it also uses real storage to simulate several address spaces, each of which is independent of the other.
  - Hence the name Multiple Virtual Storage (MVS).
  - MVS uses real storage and areas of DASD storage, called page data sets, in combination to simulate several virtual storage address spaces.
  - When multiple virtual storages are used, the total amount of virtual storage that can be simulated is almost limitless. This is because MVS can create an almost unlimited number of address spaces.
  - However, the size of an address (24 bits or 31 bits) still limits the size of each individual address space to 16 MB or 2 GB.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 8

5.4: MVS

MVS Address Space

- Furthermore, various factors limit the number of address spaces that can be simulated. Some of them are:
  - The speed of the processor
  - The amount of real storage installed effectively
- Although an MVS system can support more than one address space at a time, the CPU can access only one address space at a time.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

5.4: MVS

MVS Address Space

- When CPU is accessing instructions and data from a particular address space, that address space is said to be in control of the CPU. So the program in that address space will continue to execute until MVS intervenes and places the CPU in control of another address space.
- Multiple virtual storage is how MVS implements multiprogramming.
- Each background job or time sharing user is given its own address space.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

5.4: MVS

MVS Address Space

- So each job or user can access up to 16MB or 2GB of virtual storage independently of any other job or user on the system at the same time.
- To pass control from one job or user to another, MVS transfers control of the CPU to the other job's or user's address space.
- Then the CPU can access instructions and data in that address space until MVS is ready to pass control to a job or user in yet another address space.

5.5: Paging

Concept of Paging

- The total amount of Virtual Storage that can be used under MVS is almost unlimited. As a result, the amount of real storage present on a particular machine is nearly always less than the amount of Virtual Storage being used.
- To provide for a larger virtual storage, MVS treats DASD as an extension of real storage.
- MVS divides virtual storage into 4K sections called pages.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

The total amount of virtual storage that can be used under MVS is almost unlimited. As a result, the amount of real storage present on a particular machine is nearly always less than the amount of virtual storage being used.

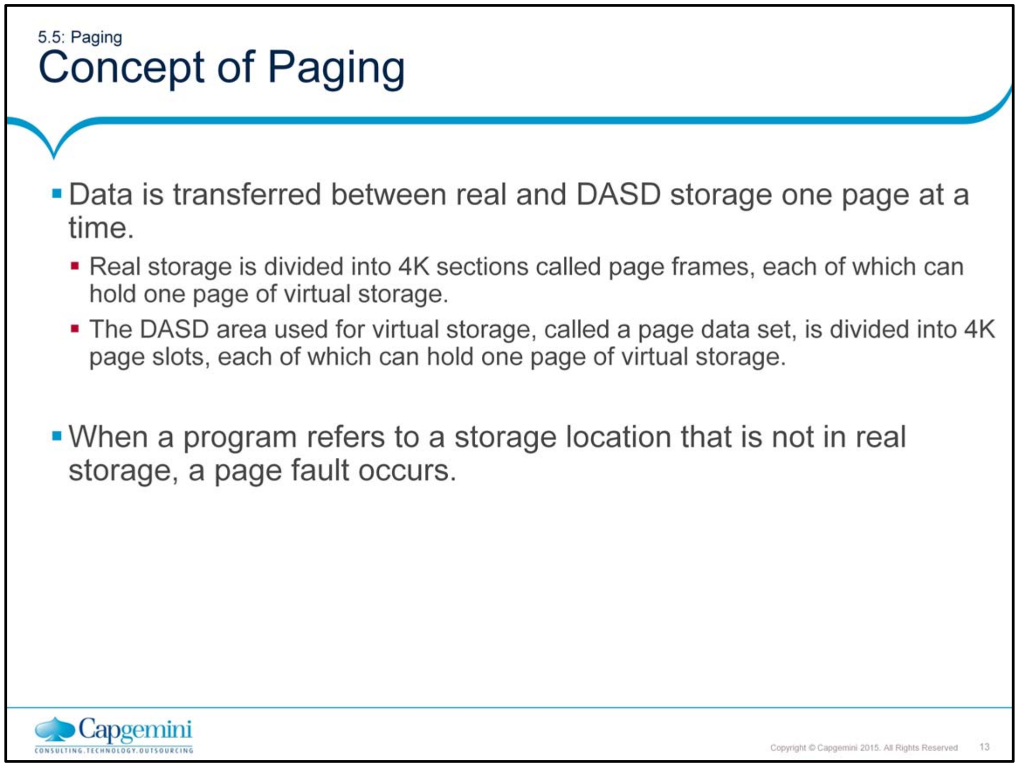
To provide for the larger amount of virtual storage MVS treats DASD storage as an extension of real storage.

MVS divides virtual storage address space into contiguous 4K (4096 bytes) blocks; each block called a PAGE.

Data is physically transferred from virtual storage to central storage in PAGES; the process is termed PAGING.

In the central storage the 4Kb page block is termed FRAME and the expanded storage it is EFRAME.

The DASD area used for virtual storage (referred to as auxiliary storage), holds many PAGES of virtual storage.



When a program refers to a storage location that is not in central storage, a page fault occurs.

Then a page-in is said to occur, whereby MVS locates the page that contains the needed data either on expanded storage or on auxiliary storage and transfers it into central storage.

The new page may overlay data in the central storage page frame or, the data in a page frame would be moved to auxiliary storage in order to produce space for the new page. This is called page-out.

MVS keeps track of what pages are in what page frames by maintaining tables that reflects the status of real storage and of each address space. The real storage frames that contain those tables cannot be paged out; they must always remain in real storage as long as their associated address spaces are active.

5.5: Paging

Concept of Paging

- When a page fault occurs:
  - MVS locates the page that contains the needed data on DASD and transfers it into real storage. This operation is called a “page-in”.
    - In some cases, the new page can overlay data in a real storage page frame.
  - Sometimes, data in a page frame has to be moved to a page data set to make room for the new page. That is called a “page-out”.

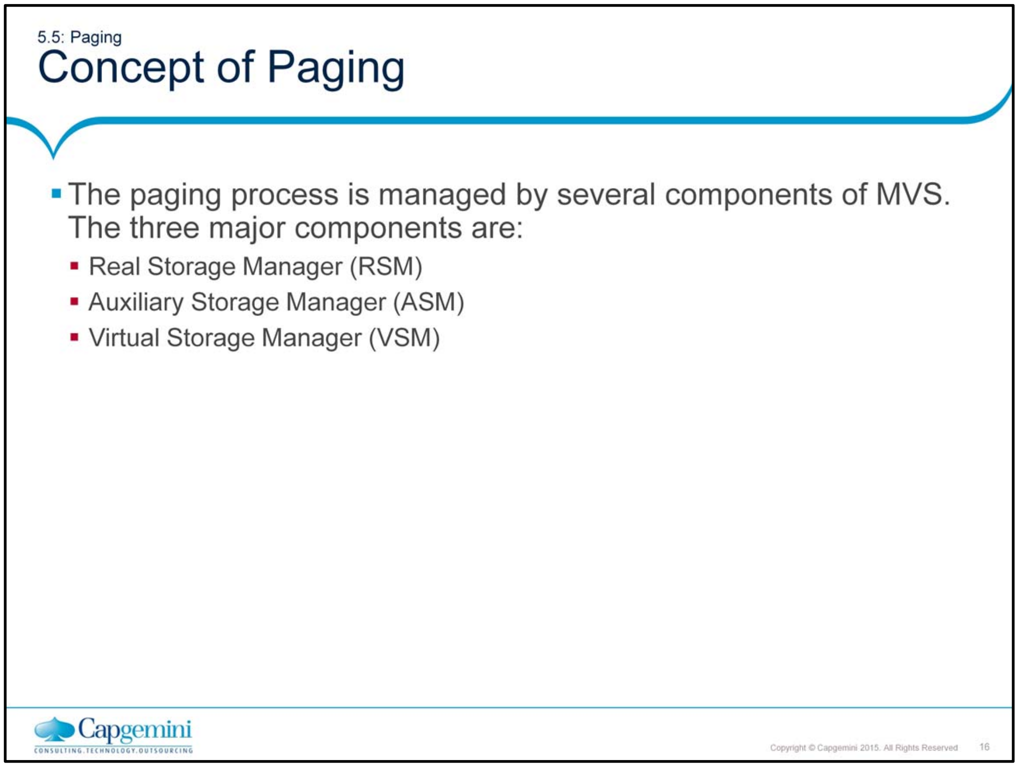
5.5: Paging

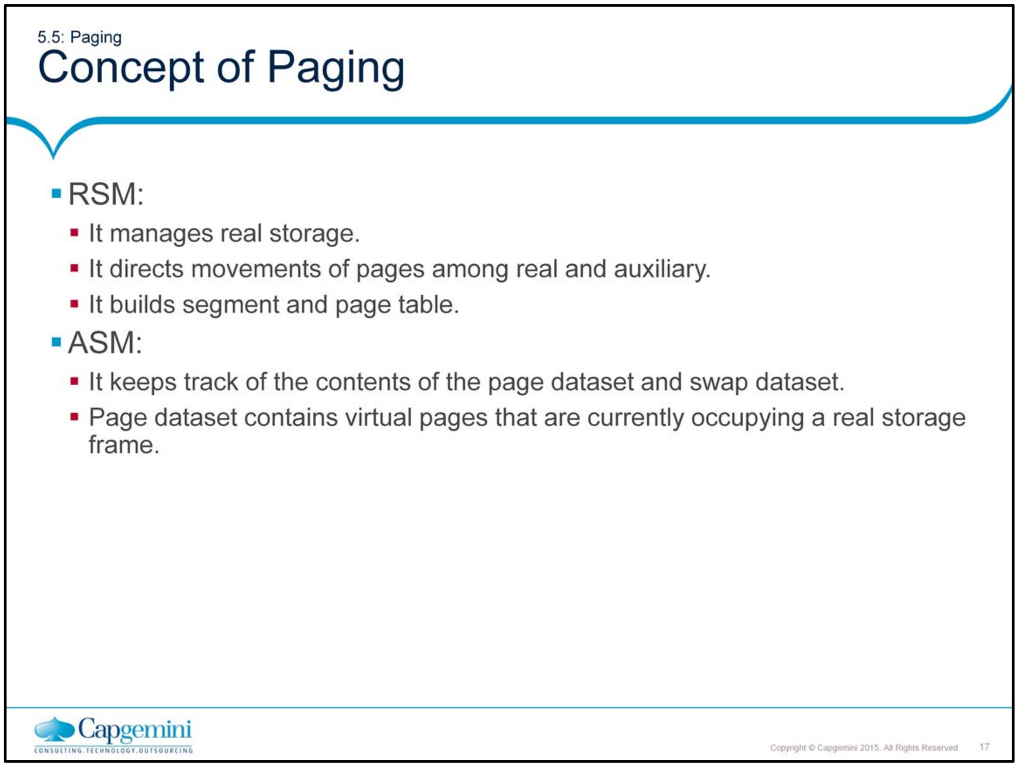
Concept of Paging

- The real storage frames that contain those tables cannot be paged out. They must always remain in real storage as long as their associated address spaces are active.
- Either way, “the process of bringing a new page into real storage is called paging”.
- At any given moment, page frames in real storage contain pages from more than one address space.
- MVS keeps track of the pages that are in particular page frames by maintaining tables that reflects the current status of real storage and of each address space.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

MVS keeps track of what pages are in what page frames by maintaining tables that reflects the current status of real storage and of each address space.


5.5: Paging
Concept of Paging
The paging process is managed by several components of MVS. The three major components are:
Real Storage Manager (RSM)
Auxiliary Storage Manager (ASM)
Virtual Storage Manager (VSM)
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
16

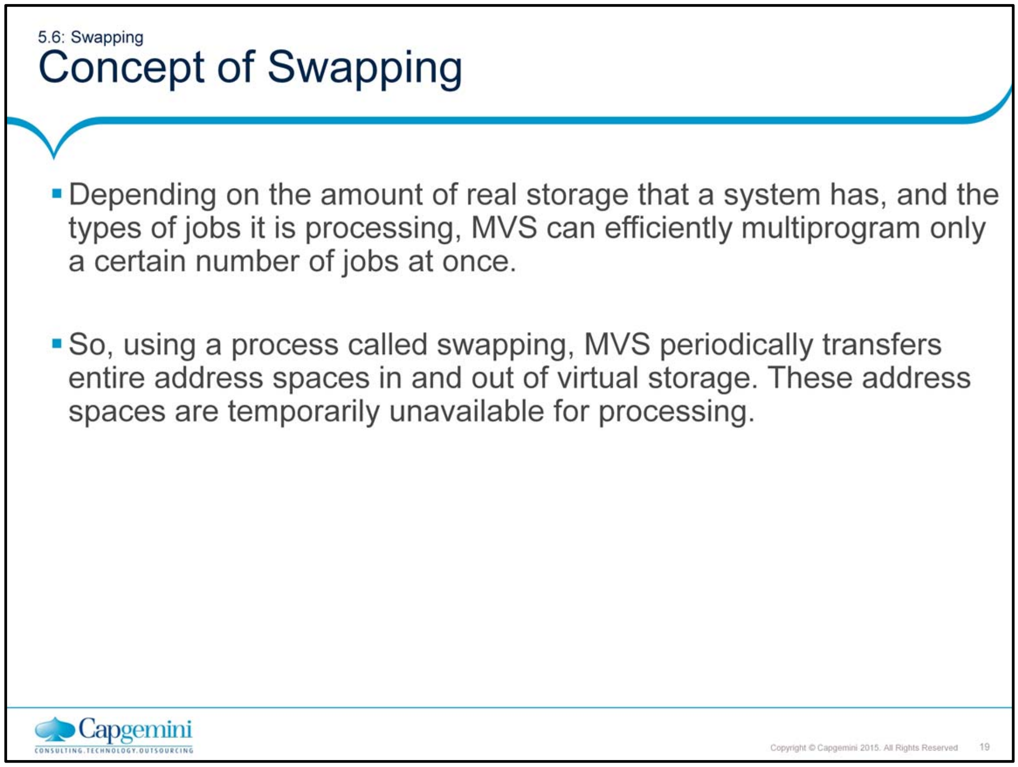

5.5: Paging
Concept of Paging
RSM:
It manages real storage.
It directs movements of pages among real and auxiliary.
It builds segment and page table.
ASM:
It keeps track of the contents of the page dataset and swap dataset.
Page dataset contains virtual pages that are currently occupying a real storage frame.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
17

5.5: Paging

Concept of Paging

- VSM:
  - It controls allocation / de-allocation of virtual storage.
  - It maintains storage use information for Storage Management Facility (SMF).

 18


5.6: Swapping
Concept of Swapping
Depending on the amount of real storage that a system has, and the types of jobs it is processing, MVS can efficiently multiprogram only a certain number of jobs at once.
So, using a process called swapping, MVS periodically transfers entire address spaces in and out of virtual storage. These address spaces are temporarily unavailable for processing.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
19

5.6: Swapping

Concept of Swapping (contd.)

- When an address space is swapped out, its critical pages – the ones that contain the tables that keep track of the location of each virtual page for the address space – are written to a special data set called a swap data set.
- Later, when the system can accommodate the job again, the address space is swapped in so that it can be processed again.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 20

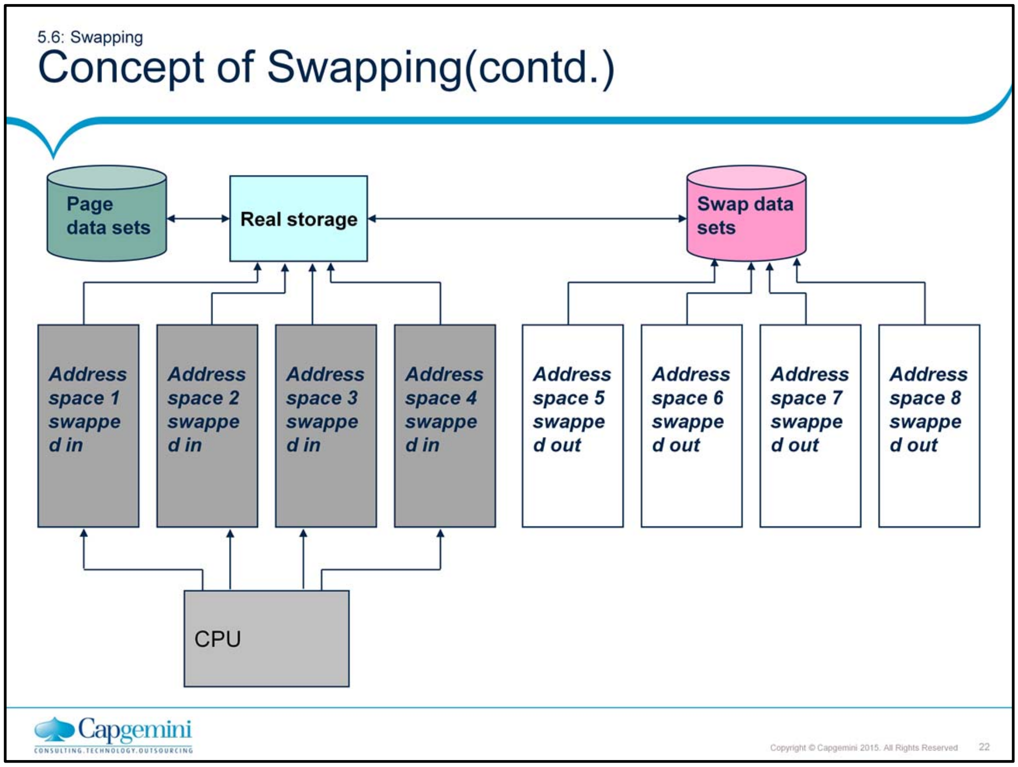
5.6: Swapping

Concept of Swapping(contd.)

- The figure on the following slide depicts the following:
  - Four address spaces are currently swapped in. The gray color indicates the address space that is currently in control.
  - Four additional address spaces are swapped out. They cannot compete for virtual storage or the CPU until they are swapped in.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 21



Note: In the figure on the above slide, four address spaces are currently swapped in. The gray color indicates the address space that is currently in control.
Four additional address spaces are swapped out. They cannot compete for virtual storage or the CPU until they are swapped in.

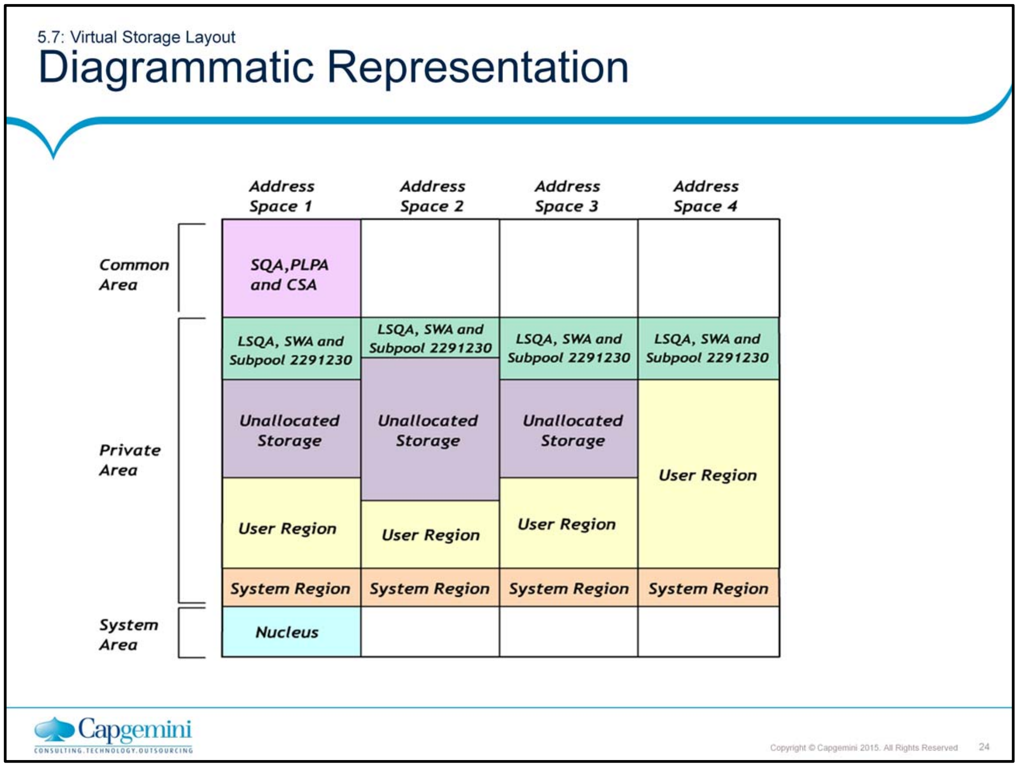
5.6: Swapping

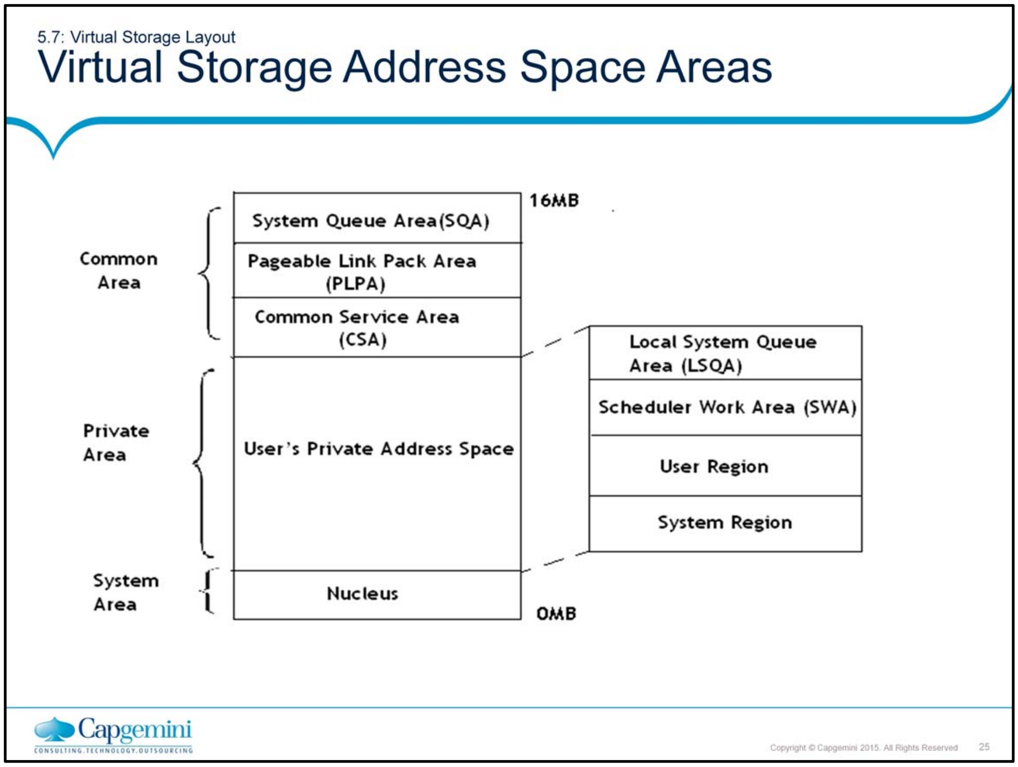
Concept of Swapping (contd.)

- Swapping is same thing as paging, only at a higher level.
- Rather than moving small 4K pieces of virtual storage in and out of real storage, swapping effectively moves entire address spaces in and out of virtual storage.
- Since paging occurs only for address spaces that are currently in virtual storage, paging does not occur for address spaces that are swapped out.

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

 23


5.7: Virtual Storage Layout
Diagrammatic Representation
Address Space 1
Address Space 2
Address Space 3
Address Space 4
Common Area
SQA,PLPA and CSA
Private Area
LSQA, SWA and Subpool 2291230
LSQA, SWA and Subpool 2291230
LSQA, SWA and Subpool 2291230
LSQA, SWA and Subpool 2291230
Unallocated Storage
Unallocated Storage
Unallocated Storage
User Region
User Region
User Region
User Region
System Region
System Region
System Region
System Region
System Area
Nucleus
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
24


5.7: Virtual Storage Layout
Virtual Storage Address Space Areas
16MB
System Queue Area(SQA)
Common Area
Pageable Link Pack Area (PLPA)
Common Service Area (CSA)
Local System Queue Area (LSQA)
Private Area
User's Private Address Space
Scheduler Work Area (SWA)
User Region
System Region
System Area
Nucleus
0MB
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
25

5.7: Virtual Storage Layout

Virtual Storage Address Space Areas

- System Area:
  - It contains the nucleus load module, page frame table entries, data blocks for system libraries, and so many other things
  - It is always resident in the memory.
  - Contains operating system programs and data
  - These areas are shared by all address space on the system
  - Resides at the low end of the address space
  - Contains the MVS nucleus which among other things controls the operations of virtual storage paging and swapping

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

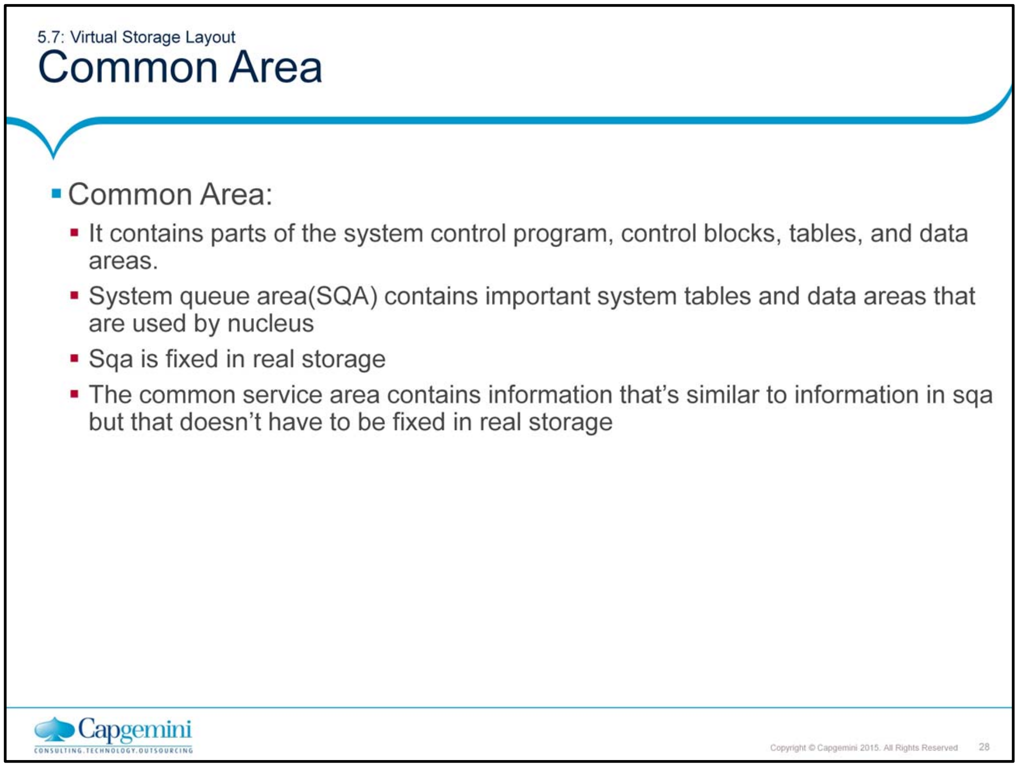
 26

5.7: Virtual Storage Layout

System Area

- The entire system area must be resident at all times so it operates in real mode
- It can't be paged or swapped

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING


5.7: Virtual Storage Layout
Common Area
Common Area:
It contains parts of the system control program, control blocks, tables, and data areas.
System queue area(SQA) contains important system tables and data areas that are used by nucleus
Sqa is fixed in real storage
The common service area contains information that’s similar to information in sqa but that doesn’t have to be fixed in real storage
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
28

5.7: Virtual Storage Layout

Common Area –Pageable Link Pack Area

- Contains operating system programs that don’t have to be fixed in real storage in the nucleus.
- Not fixed in real storage.

 29

5.7: Virtual Storage Layout

Virtual Storage Address Space Areas

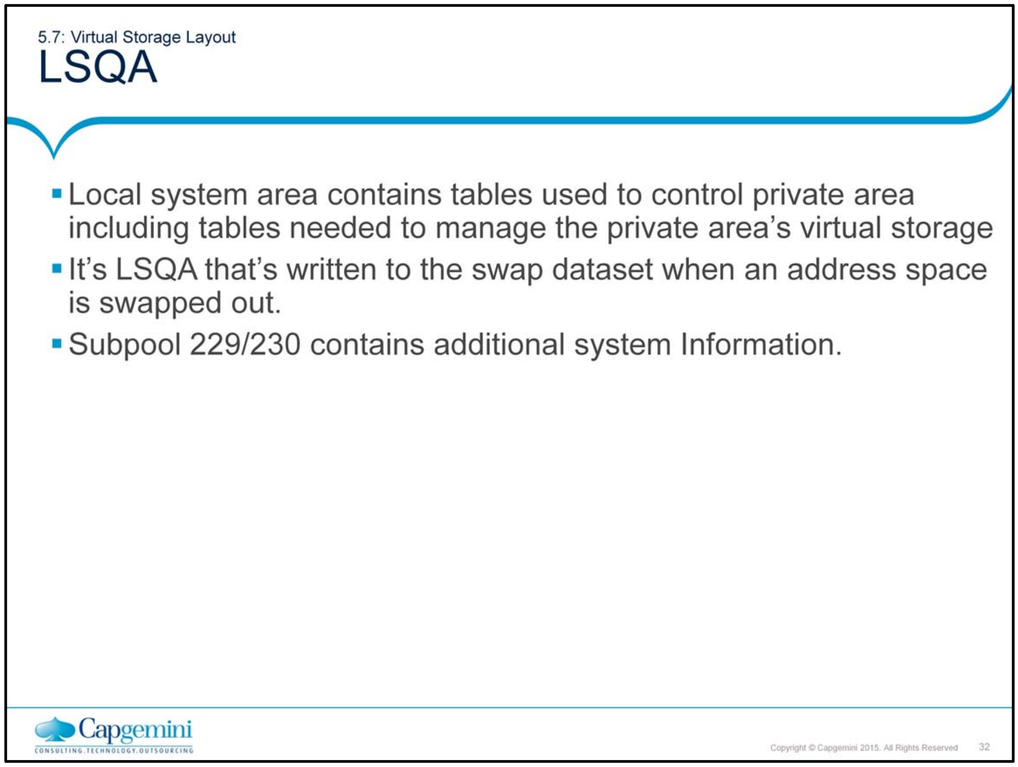
- Private Area:
  - The Private Area is made up of :
    - System Region
    - User Region
    - Scheduler Work Area (SWA)
    - Local System Queue Area (LSQA)

5.7: Virtual Storage Layout

Private Area

- Is the portion of address space that contains data that's unique for each address space
- Within each job's or user's private area there are 3 basic areas
- At the bottom of the private area is system region an area of storage used by operating systems program that provide services for users program running in private area

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



At the top of the private area are three local system areas that contain information that applies only to the private area of a particular address space

5.7: Virtual Storage Layout

Virtual Storage Address Space Areas

- User Region:
  - It is the space within Private Area that is available for running the user’s program.
- Scheduler Work Area (SWA):
  - SWA contains control blocks that exist from task initiation to task termination.
  - The information in SWA is created when a job is interpreted and used during job initiation and execution.
  - It is pageable and swappable.
  - Contains tables used to manage the execution of jobs and program within private area

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 33

The rest of the private area ,which comprises most of the address space is either unallocated or allocated to a user region. It’s in the user region that your program or programs actually execute

The size of the user region varies depending on the amount of storage required by the program being executed. If necessary, the user region can allocate all of the storage between the system region and the LSQA,SWA and subpool 229/230/On most MVS systems that amounts to about 10MB to 12 MB

5.7: Virtual Storage Layout

Demo

- Demo on:
  - The TSO Mainframe Environment
  - ISPF menu

Summary

- In this lesson, you have learnt about:
  - Various MVS concepts:
    - Multiprogramming and Virtual Storage are two main feature in MVS.
    - Paging and Swapping are necessary to ensure that pages needed for execution are there in the main memory.
    - Address space in which the pages are stored in datasets.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

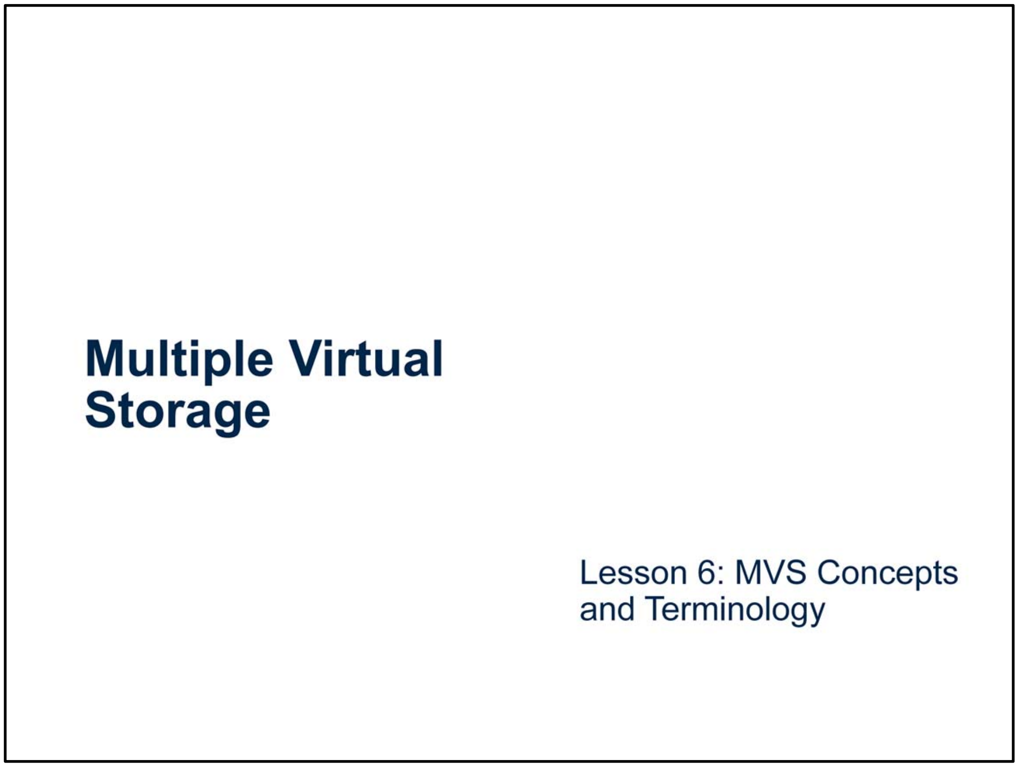
 35

Review Question: Match the Following

1. Address Space	a. Page frames
2. Multiprogramming	b. Address space
3. Virtual Storage	c. Jobs
4. Paging	d. Pages
5. Swapping	e. page datasets

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 36


Multiple Virtual Storage
Lesson 6: MVS Concepts and Terminology

Lesson Objectives

- In this lesson, you will learn the following topics:
  - Various types of data used in MVS
  - Dataset organization
    - VSAM and Non-VSAM
  - MVS Datasets
  - Data Management
  - Accessing Datasets

6.1: Types of Data

MVS Data Management

- Anything that needs to be stored and accessed on user request is data for MVS.
- Various types of data used in MVS are listed below:
  - Business Data
  - Application Components
  - MVS (System Data)
  - Temporary Data

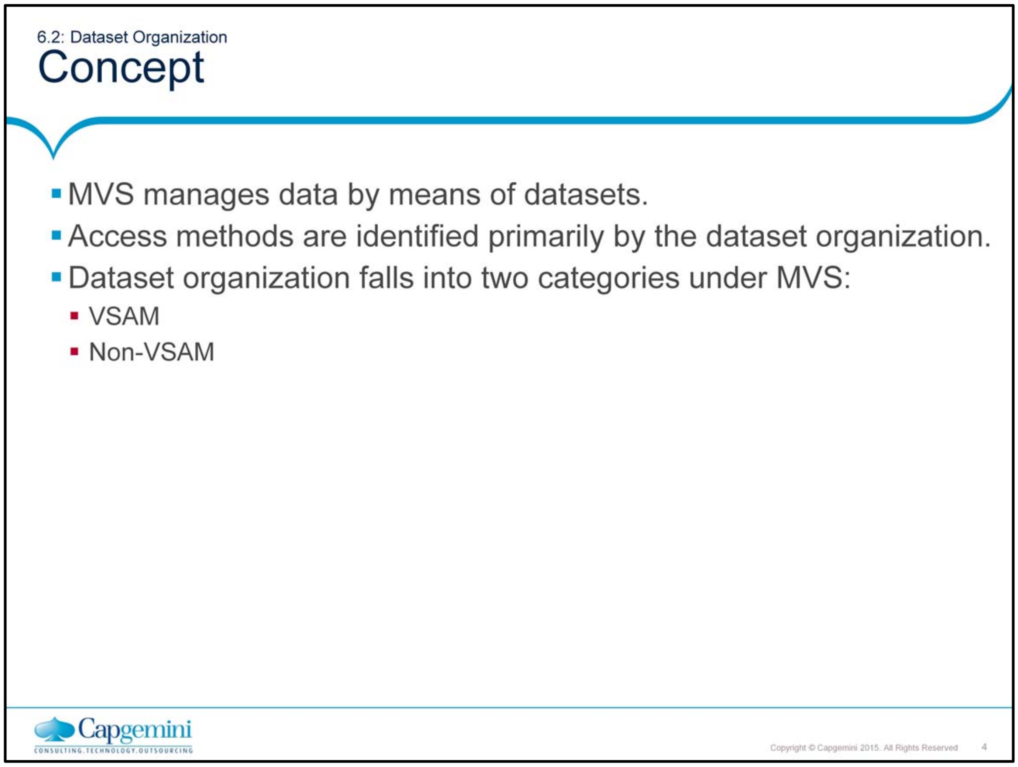
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Types of Data:
Various types of data used in MVS are listed below:
Business Data:
- Database
- Indexed Files
- Flat Files

Application Components:
- Source Programs
- Executable Programs
- Screen Definitions
- Record Layout Definitions
- Command File Scripts

MVS (System Data):
- O/S program
- User Information (ID, Password, Profile)
- Access Permissions

Temporary Data:
- O/S Built Data (for example: task queues, segment table, page table)
- Spooled Output
- Work Files for Sort



Dataset Organization:
Dataset refers to a file that contains one or more records. The record is the basic unit of information used by a program running on MVS OS.
Any named group of records is called a dataset. Datasets can hold information, such as medical records or insurance records, to be used by a program running on the system.
All types of datasets can be stored on DASD (only sequential datasets can be stored on magnetic tape).
An access method defines the technique that is used to store and retrieve data.
Access methods have their own dataset structures to organize data.
Dataset organization is categorized as VSAM and Non-VSAM:

> Virtual Storage Access Method (VSAM) : It applies to both a dataset type and the access method used to manage various user datatypes.

6.2: Dataset Organization

Concept of Non-VSAM Organization

- Non-VSAM provides four basic methods of organizing data stored in datasets:
  - Physical Sequential
  - Indexed Sequential
  - Direct
  - Partitioned

Non-VSAM:

Let us see the non-VSAM methods of organizing data:

Physical Sequential:

Records are stored one after another in consecutive sequence.

Data can reside on just any type of I/O devices.

This method of organizing data is appropriate when records from files do not have to be retrieved at random.

Indexed Sequential:

This method of organizing data includes an index, which relates key field values to the location of their corresponding data records.

Direct Organization:

This method permits random access of records.

It does not use an index.

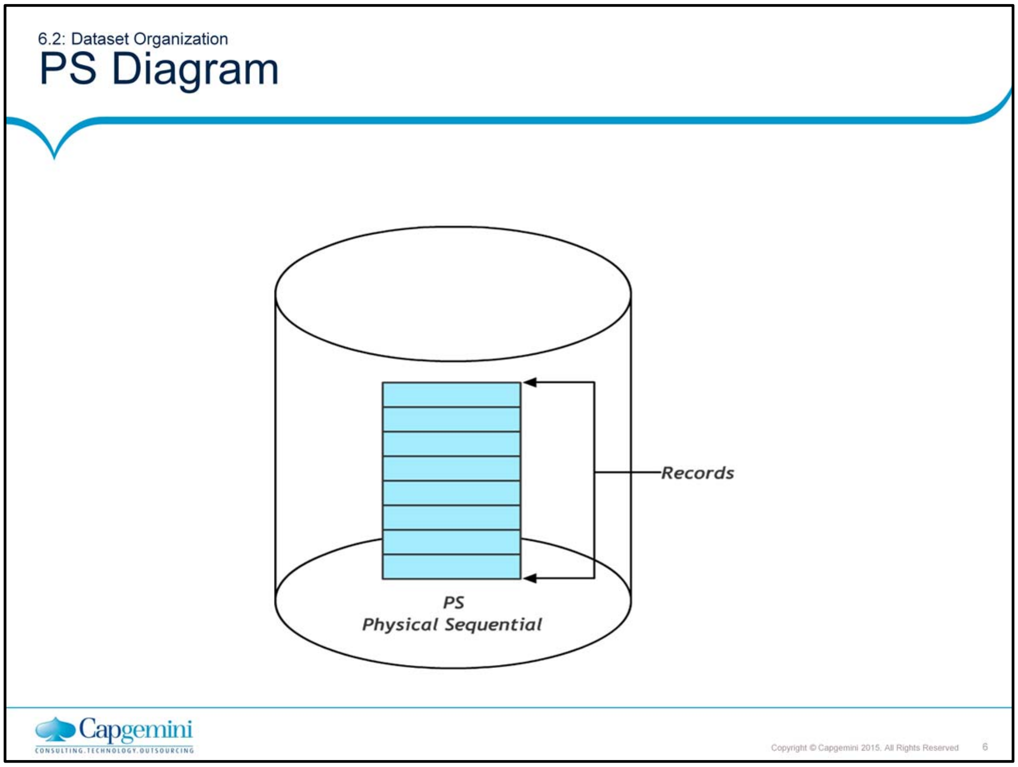
To access records, the disk location address (By hashing) of that record has to be specified.

Partitioned Organization:

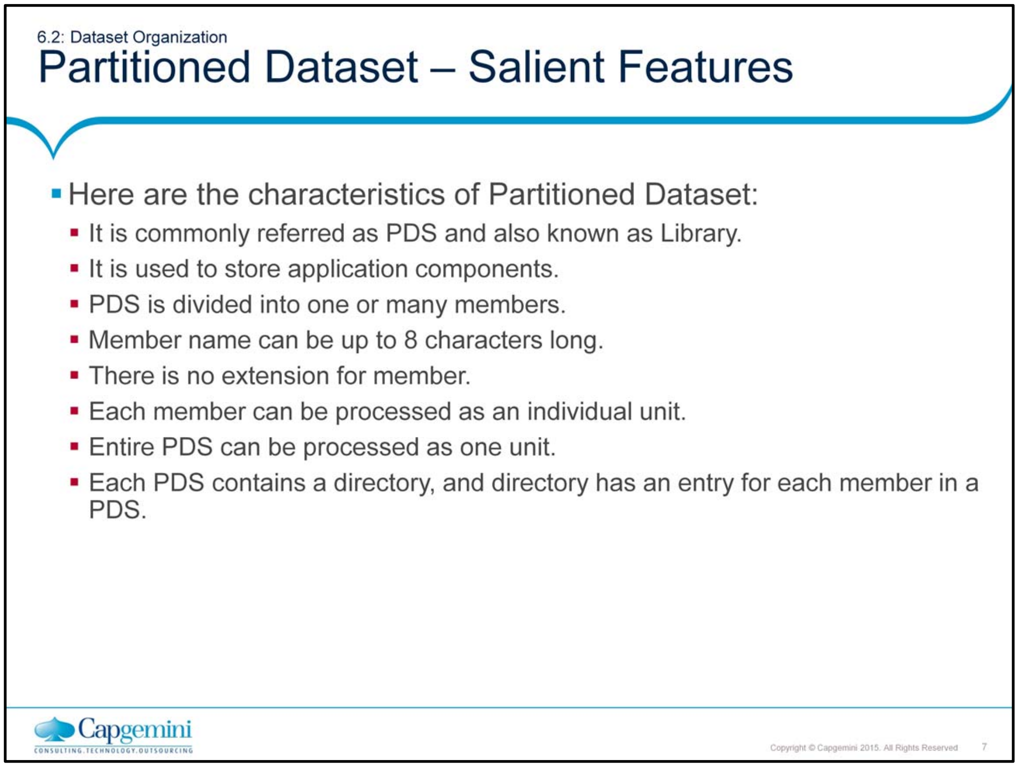
This method consists of one or more members.

Each of these members can be processed, as if it were a separate physical sequential file.

Names of members in a Partitioned dataset (PDS) is stored in a directory.



Note: Diagram denotes the PS file in DASD.

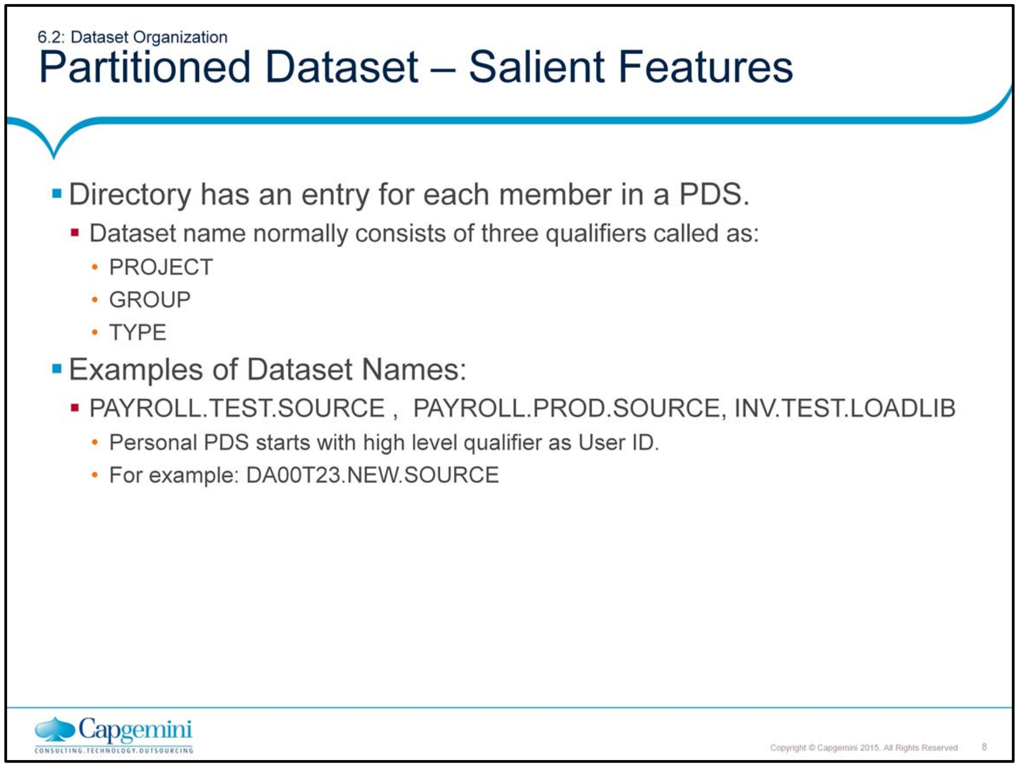


Non-VSAM:

Partitioned Dataset – Salient Features:

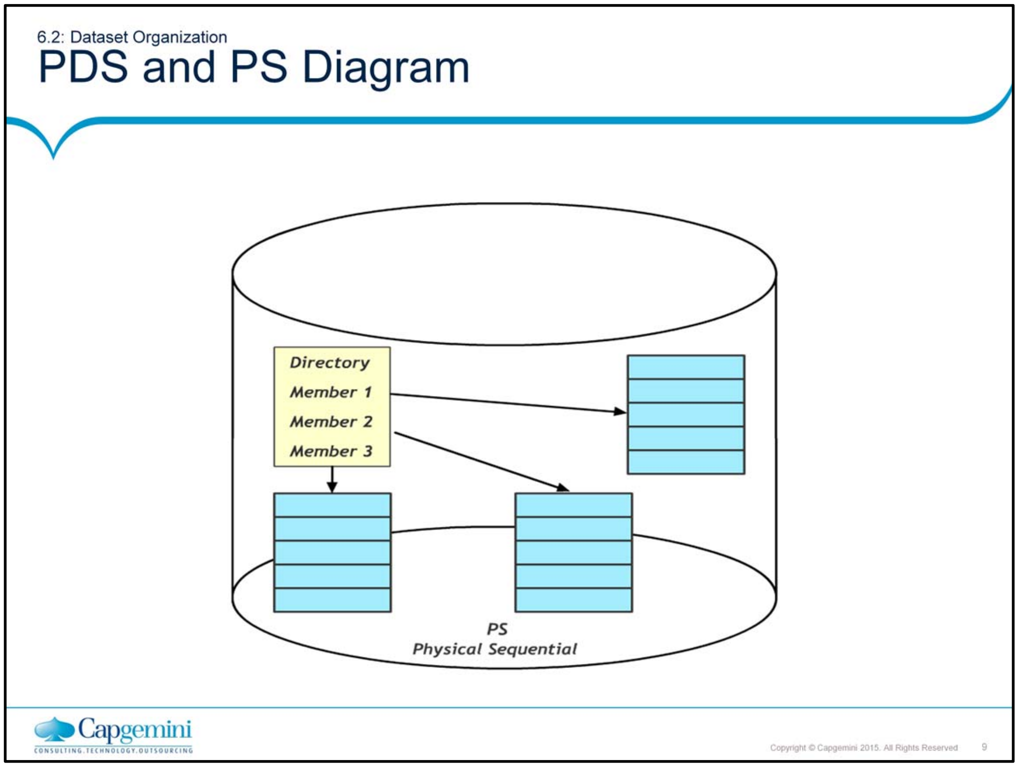
Additional Functions for PDS are given below:

- Compress
- Member Management
- Create, Modify, Delete, Copy, Rename

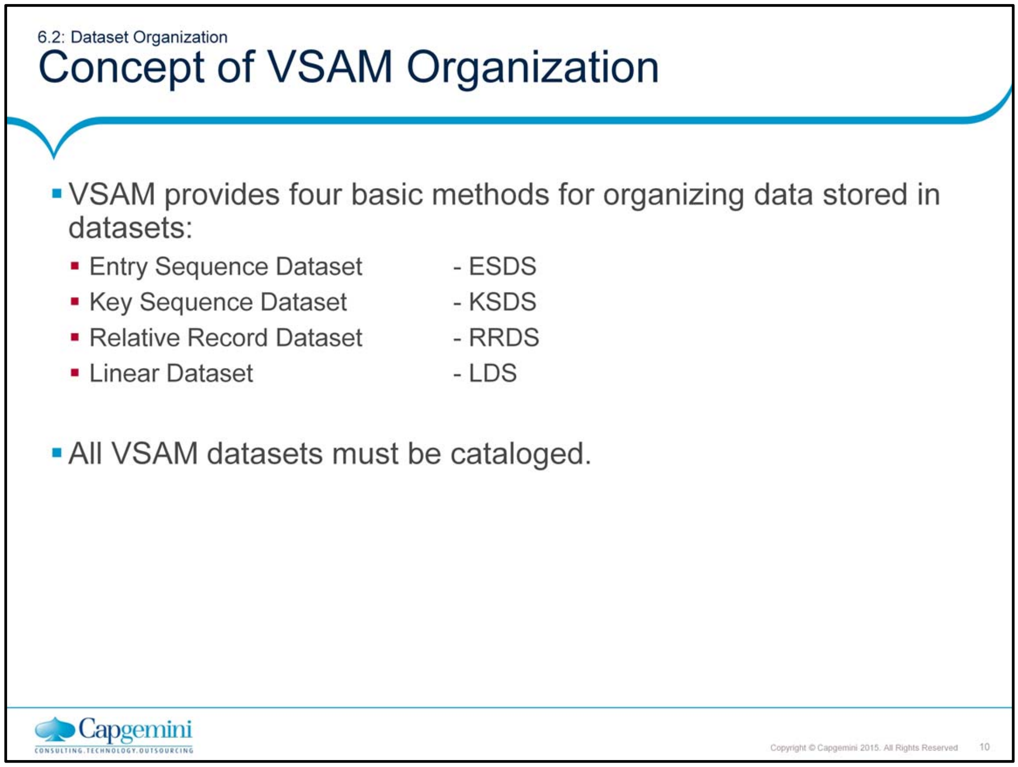


Non-VSAM:
Partitioned Dataset – Salient Features:
Some Examples of (PS) Member Name are given below:
PAB0017, PAB0105, PAC0021

Note: Usually, the application component type cannot be identified from the member name. Hence naming conventions are used for PDS.



Note: The diagram in the above slide shows the Directory (that is PDS) and PS (that is Member in a PDS).



VSAM:

VSAM is used to organize records into four types of datasets: key-sequenced, entry-sequenced, linear, or relative record.

The primary difference among these types of datasets is the way in which their records are stored and accessed.

VSAM (Virtual Sequential Access Method) is used for more complex applications. VSAM arranges records by an index key, relative record number, or relative byte addressing. VSAM is used for direct or sequential processing of fixed-length and variable-length records on DASD. Data that is organized by VSAM is cataloged for easy retrieval.

The VSAM methods of organizing data are given below:

- ESDS:
  - It can only reside on DASD.
  - It is functionally equivalent to Physical Sequential File.
- KSDS:
  - It is functionally equivalent to Indexed Sequential File.
- RRDS:
  - It lets you retrieve the record by specifying the location relative to the start of the file.
- LDS:
  - Linear is the only form of a byte-stream dataset in traditional MVAS files.

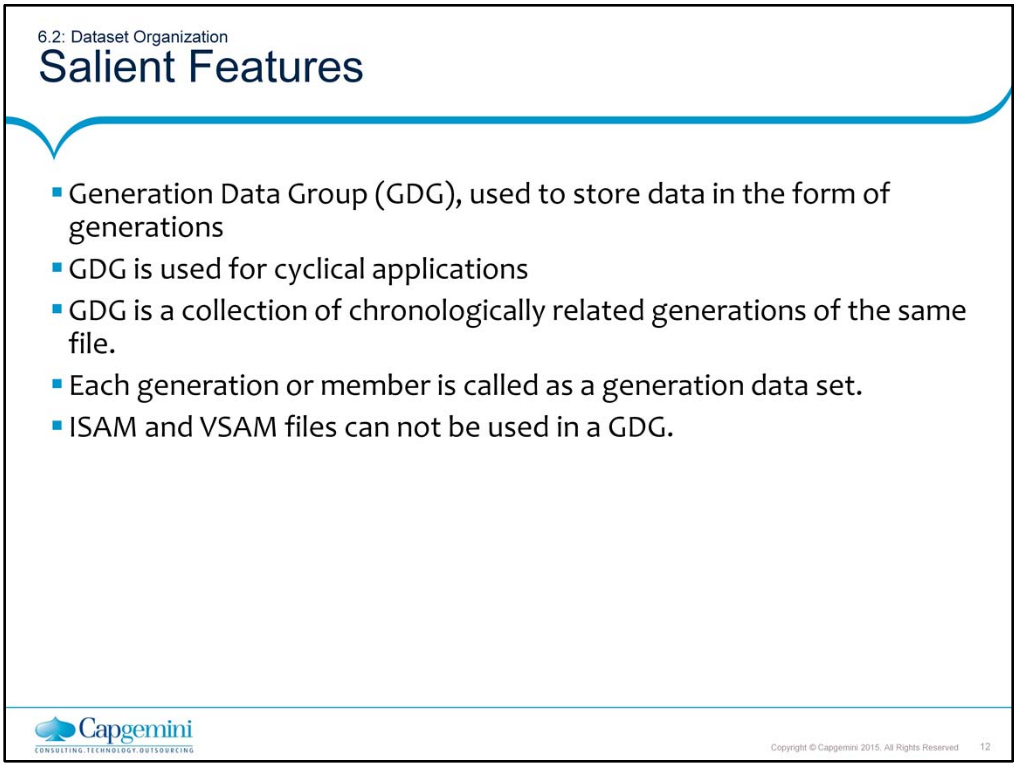
6.2: Dataset Organization

Salient Features

- Non-VSAM was developed in mid 1960s.
- VSAM (Virtual Storage Access Method) was introduced in early 1970s.
- VSAM was expected to replace Non-VSAM Data Organization Functions.
  - Today, most sites use both VSAM and Non-VSAM Data Organization.
- VSAM is the primary data organization for user data.
- VSAM is also called as “native” file management system of IBM.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11



Generation data group (GDG) is used to store data in the form of Generations. Used for cyclical applications.

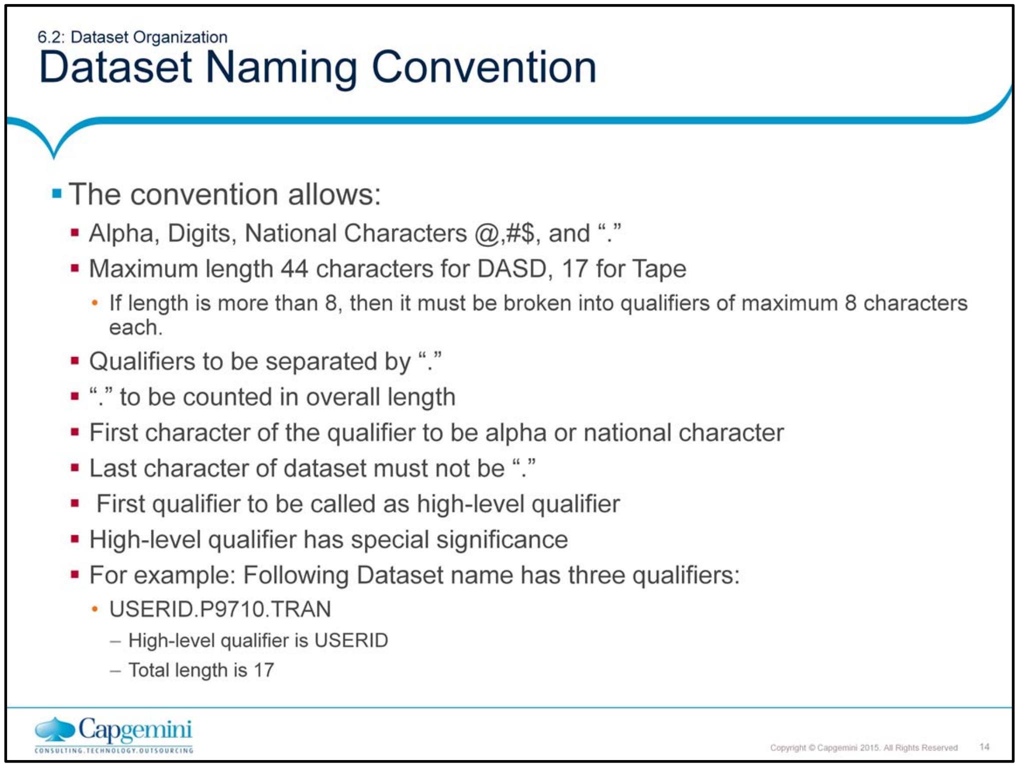
- It is a collection of chronologically related generations of the same file. Each generation or member is called as a generation data set.
- ISAM and VSAM files can not be used in a GDG.
- Each processing cycle creates a new generation.
- GDGs must be catalogued so that MVS can use the catalog entries to keep track of the relative generation numbers.
- When the GDG catalogue entry is made, we specify the number of generations that we wish to create.

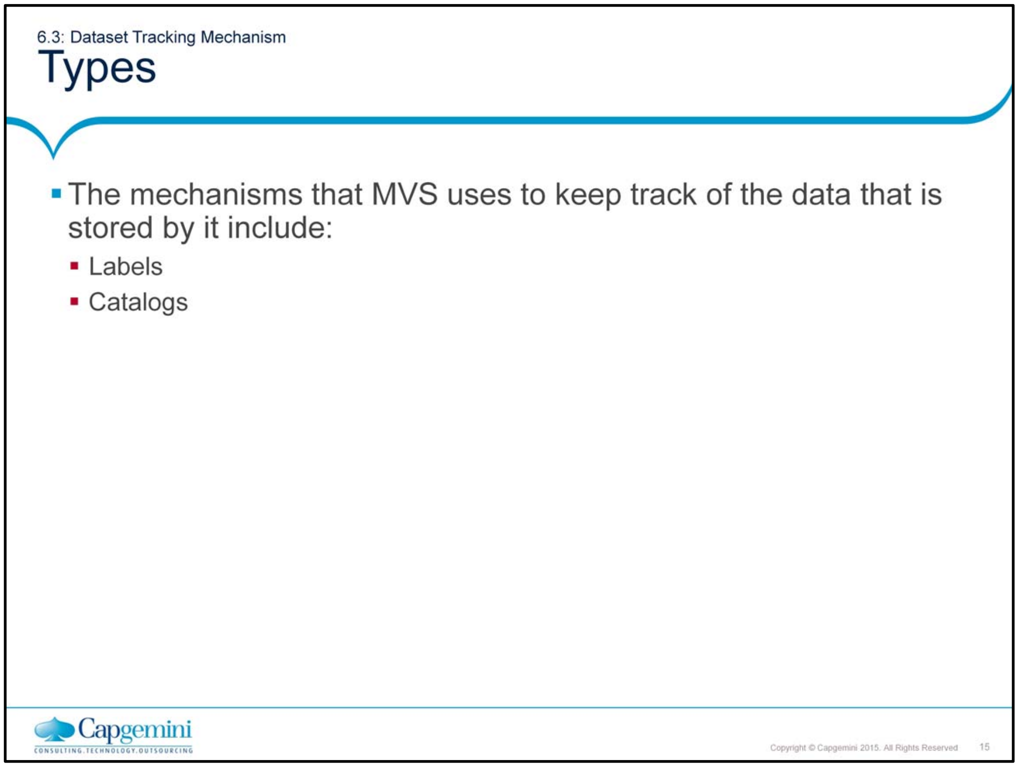
6.2: Dataset Organization

Salient Features

- Most of the DBMS running under MVS use VSAM as underlying Data Organization (for example: DB2, IDMS).
- Physical Sequential Data Organization is used for “flat” files.
- Index Sequential and Direct Data Organization are not very popular now-a-days (these functions are handled better by VSAM).
- Partitioned Data Sets (PDS) are also used by MVS to store O/S programs.

 13


6.2: Dataset Organization
Dataset Naming Convention
The convention allows:
Alpha, Digits, National Characters @,#$, and "."
Maximum length 44 characters for DASD, 17 for Tape
If length is more than 8, then it must be broken into qualifiers of maximum 8 characters each.
Qualifiers to be separated by "."
"." to be counted in overall length
First character of the qualifier to be alpha or national character
Last character of dataset must not be "."
First qualifier to be called as high-level qualifier
High-level qualifier has special significance
For example: Following Dataset name has three qualifiers:
USERID.P9710.TRAN
– High-level qualifier is USERID
– Total length is 17
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
14


6.3: Dataset Tracking Mechanism
Types
The mechanisms that MVS uses to keep track of the data that is stored by it include:
Labels
Catalogs
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
15

6.3: Dataset Tracking Mechanism

Label Processing

- When a dataset is stored on disk or tape, MVS identifies it with special records called ‘labels’.
- There are two types of DASD labels:
  - Volume Label or Vol1 Label or DASD Label
  - File Label or Dataset Label

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

6.3: Dataset Tracking Mechanism

Label Processing - Volume Label

- Volume Label Processing:
  - Each DASD is labeled. It is also called as Volume Label (VOL1 label).
  - VOL1 label is stored on a disk volume at third record of track 0 in cylinder 0.
  - VOL1 label (Volume label) has two important functions:
    - It identifies the volume by providing a volume serial no. : Vol-ser. Every DASD volume must have a unique six-characters vol-ser.
    - It contains the disk address of the VTOC.
    - VTOC (Volume Table of Contents) is a special file that contains the file labels for all the datasets on the volume.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17

Dataset Tracking Mechanism:
Volume Serial Number:
Each DASD is identified by a unique number, Volume Serial Number, that is vol-ser. Vol Ser must be specified for accessing the Dataset (which is not cataloged). Every DASD volume must have a unique six-characters vol-ser.
VTOC:
Volume Table Of Contents that is VTOC is a special file for each DASD. The VTOC contains the file labels for all the datasets on the volume. These labels are called Data Set Control Block (DSCB) have several formats called Format-1, Format-2, and so on.

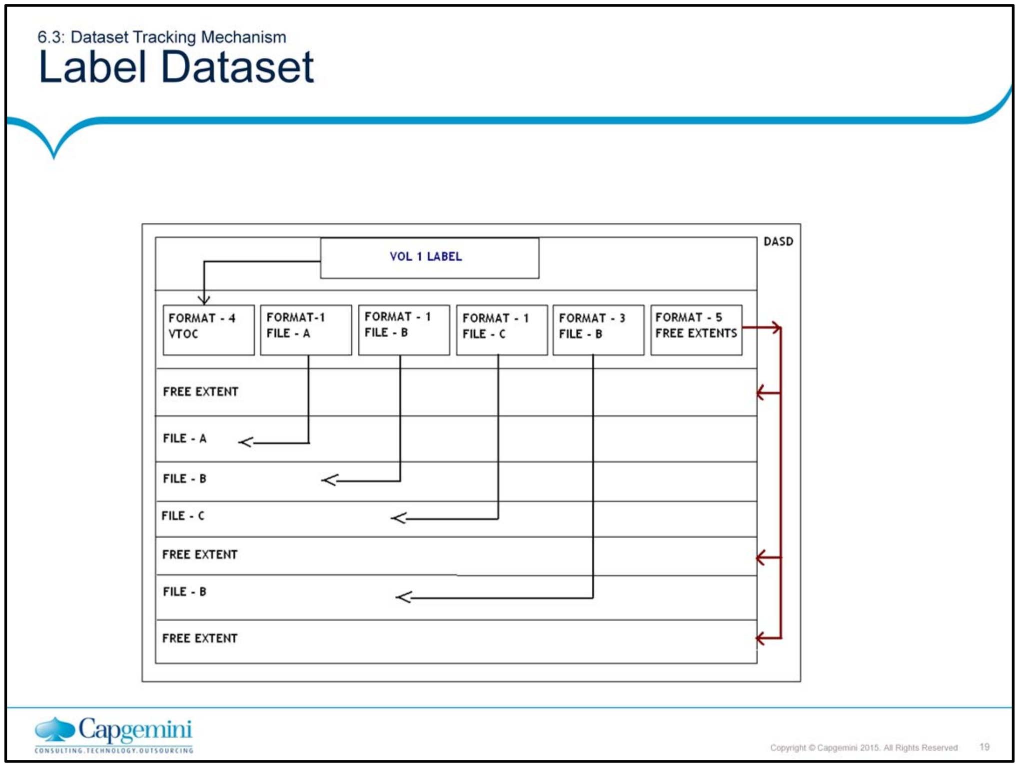
6.3: Dataset Tracking Mechanism

Label Processing - Dataset Label

- Dataset Label Processing:
  - Each dataset is a label record called File label or Data Set Control Block (DSCB).
  - DSCB describes dataset's name, it's DASD location, and other details.
  - DSCBs have several formats, called Format-1, Format-2, and so on.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 18



Label Processing
When a dataset is stored on disk or tape, MVS identifies it with special records called ‘labels’.
• There are 2 types of DASD labels : Volume, File Label
• All DASD volumes must contain a volume label, often called a VOL1 label. This label is always in the same place on a disk volume : the 3rd record of track zero in cylinder zero.
• Volume label has 2 important functions
It identifies the volume by providing a volume serial no. : Vol-ser. Every DASD volume must have a unique six-characters vol-ser.
It contains the disk address of the VTOC.
• The VTOC (Volume Table of Contents) is a special file that contains the file labels for the datasets on the volume.
• These labels are called Data Set Control Block (DSCB) have several formats called Format-1, Format-2 and so on.

Format-4-dscb : describes VTOC itself
Format-1-dscb : describes a dataset by supplying dataset name, DASD location & other characteristics [space is allocated to DASD file in area called extents. Each extent consists of one or more adjacent tracks]
[has room to define 3 extents for a file (1 primary, 2 secondary)
Format-3-dscb : if file requires more than 3 extents, this dscb is created
It contains room for 13 additional secondary extents [As a result file can contain up to 16 extents]
Format-5-dscb : contain information about free extents that aren’t allocated to files each can define up to 26 free extents

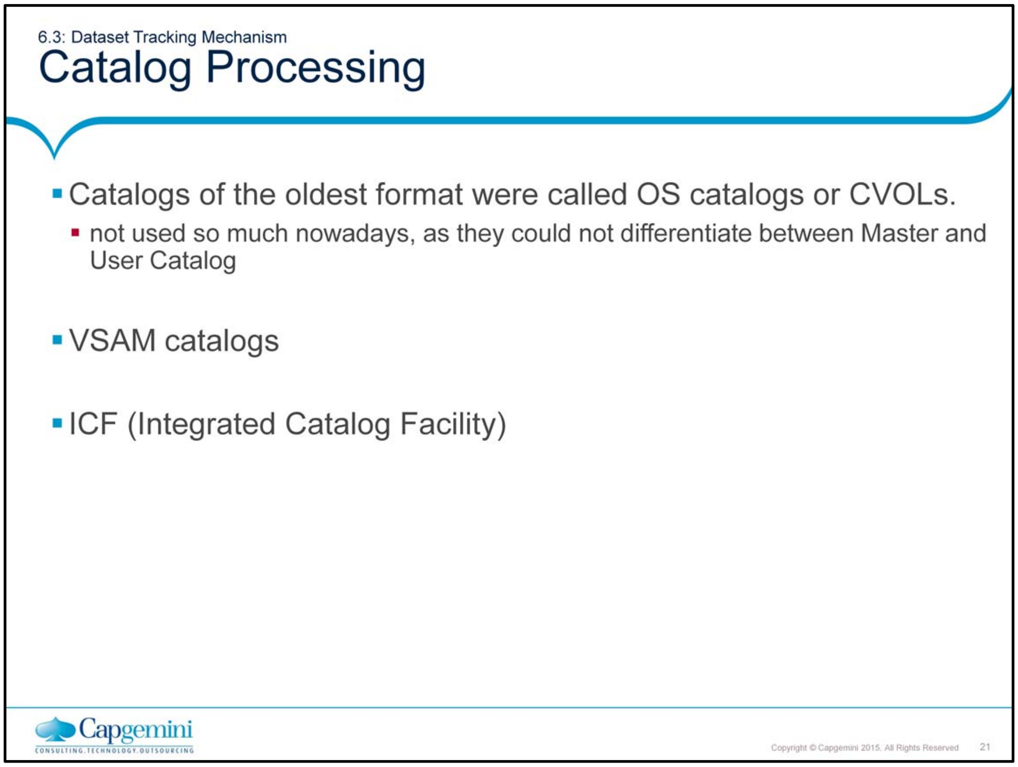
6.3: Dataset Tracking Mechanism

Catalog Processing

- MVS provides a comprehensive catalog facility.
  - It records the location of the files
  - Under MVS, there are two types of catalogs
    - MASTER Catalog
    - USER Catalog
- MASTER Catalog
  - In each MVS system, there is only one Master catalog
  - It contains entries that identify system datasets and an entry for each User catalog created in the Mainframe.
- USER Catalogs
  - There can be a number of User catalogs
  - User Catalogs contain entries that identify User datasets.

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

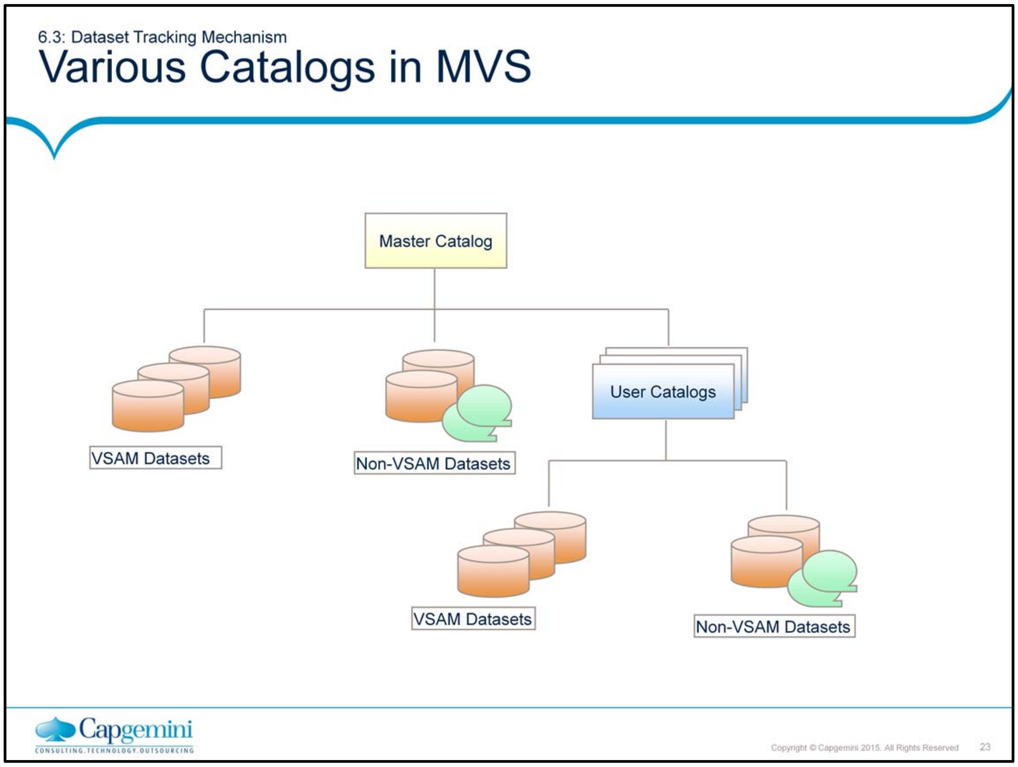
 20


6.3: Dataset Tracking Mechanism
Catalog Processing
Catalogs of the oldest format were called OS catalogs or CVOLs.
not used so much nowadays, as they could not differentiate between Master and User Catalog
VSAM catalogs
ICF (Integrated Catalog Facility)
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
21

6.3: Dataset Tracking Mechanism

Catalog Processing

- Catalog Features:
  - Each MVS has only one Master Catalog which also functions as system catalog.
  - Master Catalog is used by MVS for system datasets.
  - User Catalog is used for user datasets.
  - There can be multiple User Catalogs.



- Master Catalog contains one entry of each User Catalog:
  - VSAM datasets must be cataloged.
  - Non-VSAM datasets may or may not be cataloged.
  - An Alias can be created for a catalog.
- Usually, the high-level qualifier of a dataset is same as the catalog name or catalog alias name.
- Multiple datasets can be cataloged in single user catalog.
- Alias helps to catalog datasets with different high-level qualifiers to be cataloged in a single user catalog.

6.4: Sysplex

Sysplex

- Sysplex
  - A system complex, commonly called a Sysplex, allows multiple processors to be joined into a single unit, sharing the same Sysplex name and Couple Data Sets.
  - Put another way, a Sysplex is a single logical system running on one or more physical systems.
  - Sysplexes are often isolated within a single system, but Parallel Sysplex technology allows multiple mainframes to act as one.
  - Used for disaster recovery, Parallel Sysplex combines data sharing and parallel computing to allow a cluster of up to 32 systems to share a workload for high performance and high availability.

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

 24

6.5: Dataset Management

Functions of Dataset

- Data Management Functions for Non-PDS:
  - Allocate
  - Process
    - Add Records
    - Modify Records
  - Delete Records
  - De-allocate (delete)
  - Copy
  - Rename
  - Catalog/Uncatalog

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 25

6.5: Dataset Management

Functions of Dataset

- Functions for PDS:
  - Compress
  - Member Management
  - Create, Modify, Delete, Copy, Rename

6.5: Dataset Management

Functions of Dataset

- How Data Management is achieved?
  - Interactively using MVS Commands
  - Executing MVS Utility Programs (batch mode)
  - Through Application Programs
  - On-line Processing
  - Batch Processing

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 27

6.6: Dataset Processing

Accessing Datasets in MVS

- How are datasets processed in MVS?
- An application program that is part of a user job goes through three phases as it processes a dataset:

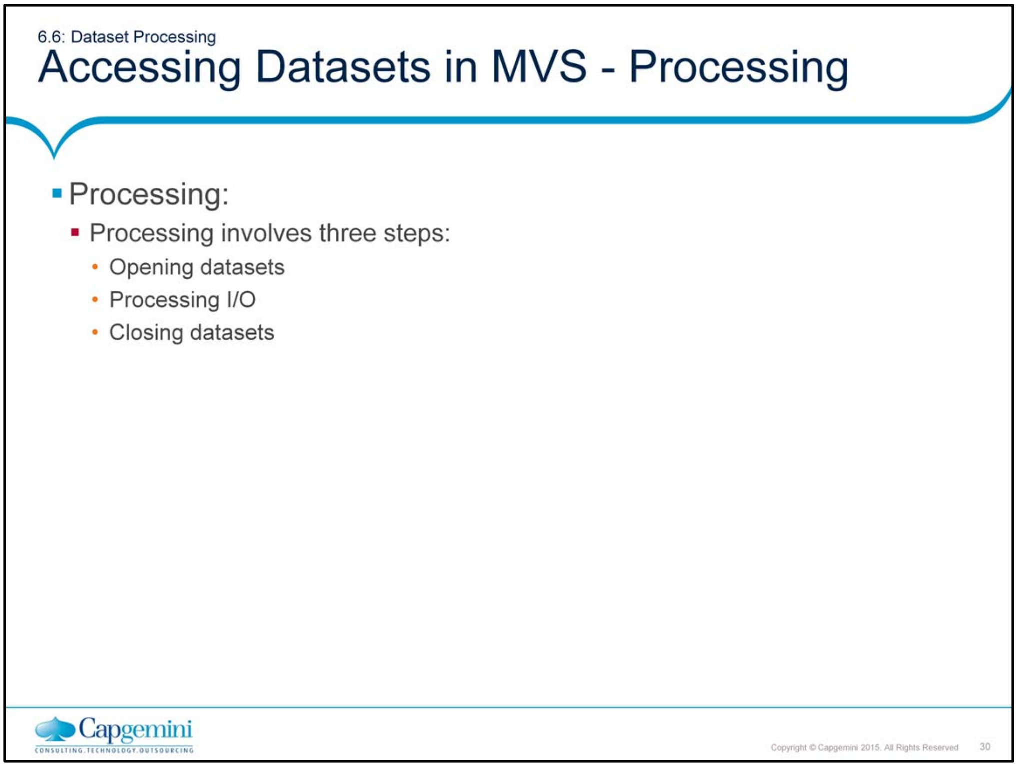
1. Allocation
2. Processing
3. De-allocation

6.6: Dataset Processing

Accessing Datasets in MVS - Allocation

- Allocation:
  - The process of locating an existing dataset or space for a new dataset and preparing the system control block needed to use the dataset is called “Allocation”.
  - Allocation occurs at three levels:
    - Unit is selected and allocated, for example: SYSALLDA-DASD, TAPE.
    - Volume is allocated.
    - Dataset on that volume is allocated.

 29



Dataset Processing:

Accessing Datasets in MVS – Processing:

An access method is an interface between an application program and the physical operations of storage devices.

Physical Sequential - BSAM, QSAM :-like sequential file disadvantage is that records have to be processed one at a time from the beginning as if the file reside on tape

Indexed Sequential - BISAM, QISAM (obsolete) records can be accessed sequentially and randomly depending on processing requirements

Direct - BDAM
Partitioned - BPAM
ESDS, KSDS, RRDS - VSAM
Entry sequenced data set –reside on dasd. Key sequenced dataset same as
indexed sequential file.
Relative record dataset:-it lets you retrieve the record by specifying the location of relative to the start of file.
LdS linear is only form of a byte-stream dataset in traditional MVS files.

6.6: Dataset Processing

Accessing Datasets in MVS - Processing

- When you code an i/o instruction in an application program and you actually invoke an access method which in turn issues proper I/o instructions to access the I/o device
- Basic sequential access method provides low level support for sequential datasets
- Basic indexed sequential file for indexed
- Basic direct access method for direct files
- Queued access methods provide a higher level of support for sequential and indexed sequential files

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 31

6.6: Dataset Processing

Accessing Datasets in MVS – De-allocation

- De-allocation:
  - Each file is automatically de-allocated when job is finished with it.
  - While de-allocating, disposition of dataset can be decided, whether you want to retain the file or should it be deleted.
  - Disposition indicates what MVS does with a non-Vsam file when its deallocated.
  - Disposition of temporary files indicates whether the file should be retained until the end of the job or deleted immediately

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 32

6.6: Dataset Processing

Accessing Datasets in MVS – De-allocation

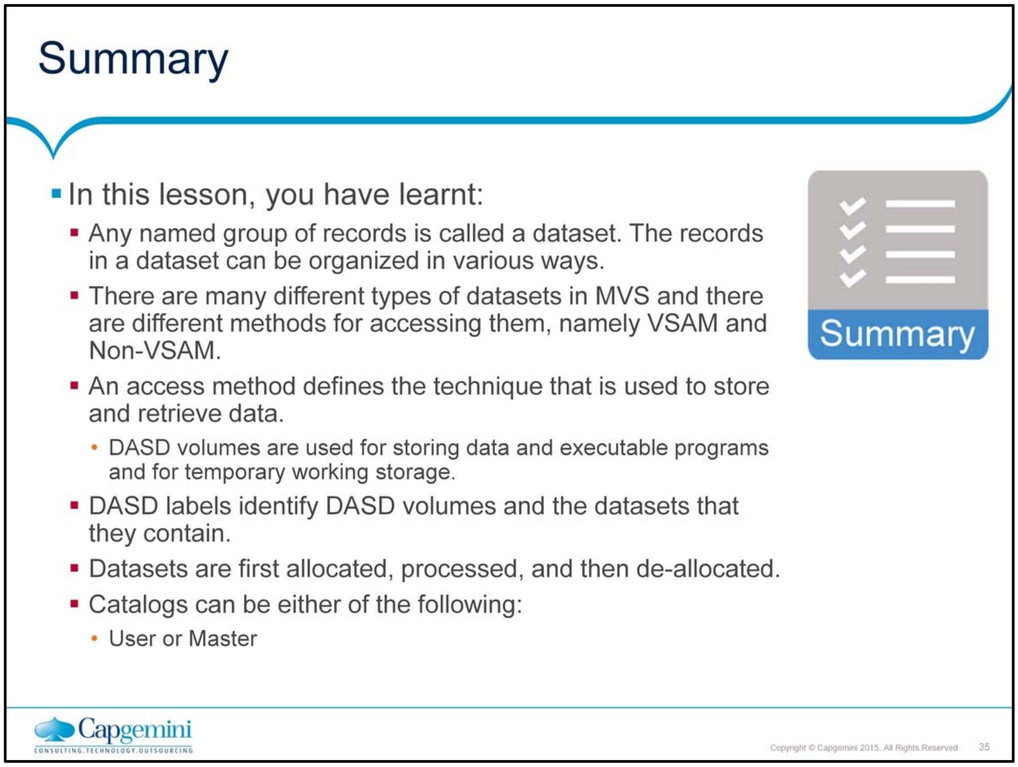
- For permanent file, disposition indicates whether a file should be kept or deleted
- In addition permanent file disposition's indicates whether an entry for the file should be maintained in master catalog or user catalog.

6.6: Dataset Processing

Demo

- Demo on:
  - Creation of PDS and PS

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING


Summary
In this lesson, you have learnt:
Any named group of records is called a dataset. The records in a dataset can be organized in various ways.
There are many different types of datasets in MVS and there are different methods for accessing them, namely VSAM and Non-VSAM.
An access method defines the technique that is used to store and retrieve data.
DASD volumes are used for storing data and executable programs and for temporary working storage.
DASD labels identify DASD volumes and the datasets that they contain.
Datasets are first allocated, processed, and then de-allocated.
Catalogs can be either of the following:
User or Master
Summary
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
35

Review Question

- Question 1: VSAM datasets are always ___.
  - Option 1: Cataloged
  - Option 2: Un-Cataloged
  - Option 3: Master Catalog
- Question 2: MVS has many different types of datasets.
  - True / False
- Question 3: ___ applies to both a dataset type and the access method used to manage various user data types.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 36

Multiple Virtual Storage (MVS)

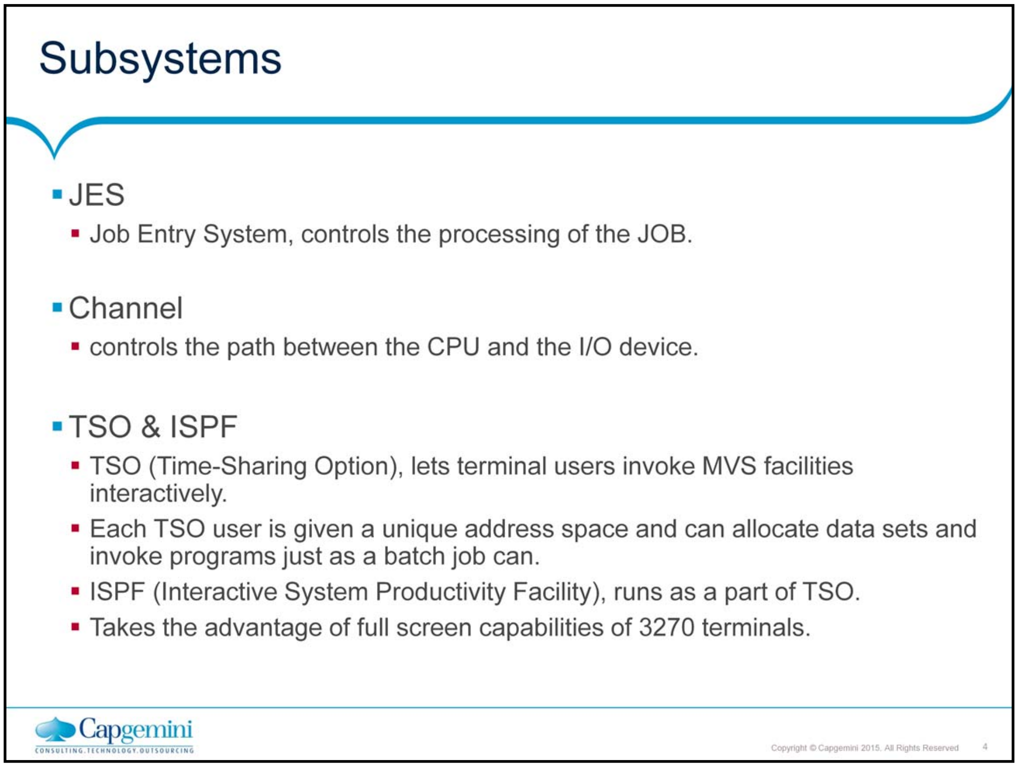
Lesson 7: MVS Tools Overview

Lesson Objectives

- In this lesson, you will learn the following topics:
  - Subsystems
    - JES
    - Channels
    - TSO/ISPF
    - CICS
    - DB2
    - IMS
    - RACF
  - Working with TSO/ISPF

Subsystems & other facilities

- Some of the facilities available under MVS are also considered to be subsystems.
- SUBSYSTEM
  - A software product that operates in its own address space under the control of MVS.
  - May provide services that duplicate services provided by the operating system.

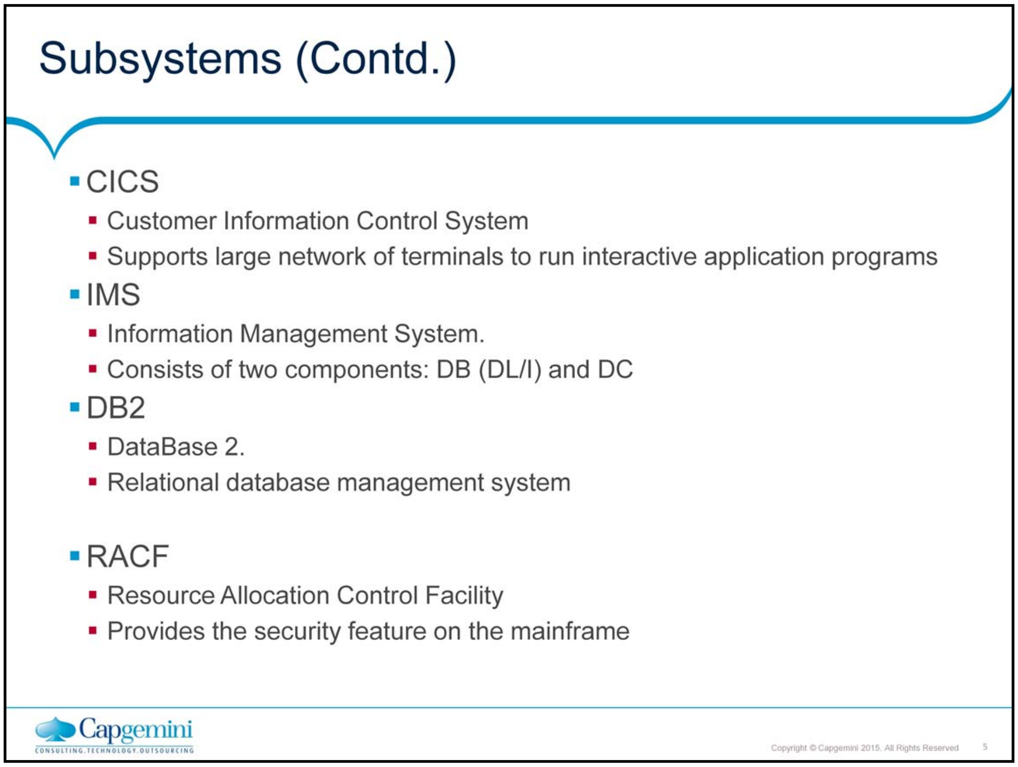


SMF:

- SMF stands for System Management Facility.
- It monitors jobs as they execute and record information such as the amount of CPU time used, the amount of DASD I/O that was performed, the number of print records that were created, and so on. This information is recorded in special datasets so that it can be used as the basis for billing.

VTAM:

- Any system supporting terminal devices, be it local or remote, should include Telecommunications Access Method.
- VTAM, that is Virtual Terminal Access Method, is one of the powerful Telecommunication (TC) Access Method.
- It is part of **System Network Architecture (SNA)**, which is a comprehensive telecommunication product.
- It is a subsystem that runs in its own address space which allows VTAM to provide centralized control over all terminals attached to the system.
- Each VTAM terminal device is allocated to VTAM address space.



CICS:

- CICS stands for Customer Information Control System.
- CICS works with VTAM applications to support large networks of terminals that can run interactive application programs.
- CICS implements multi-programming within itself. In CICS, multiple programs which are a part of the same application are executed within CICS own address space. CICS then selects one program at a time for execution. CICS itself is multi-programmed by MVS along with other programs.

DB2:

- DB2 – Database 2 – is a database management system.
- DB2 access and manages relational databases using the SQL (structured query language).
- DB2 does not provide any online environment, but online DB2 programs are written to run under CICS.

RACF:

- RACF stands for Resource Access Control Facility.
- RACF is a comprehensive security package. It provides security to restrict unauthorized users to access data in MVS.
- Both users and resources, like datasets, are identified by RACF. Whenever user tries to access a resource, the security is checked by RACF, thus restricting unauthorized user access.
- RACF is not a subsystem. Rather it is a set of routines that is stored in the PLPA, which are invoked by users address space as and when required.

Use of TSO

- Time Sharing Option (TSO) is an interactive processing tool.
- It is used by the terminal user to interactively invoke MVS facilities.
- TSO internally treats each terminal user as a Job.
- Various TSO commands are available, thus providing a variety of functions.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Time Sharing Option (TSO):
TSO is a subsystem, in which terminal users invoke MVS facilities interactively. Time Sharing option, release first in 1969. MVS handles each TSO user as it handles batch jobs. Each TSO user has an unique address space for running programs When TSO user logs on, TSO MVS initiates a terminal session, which is unique to the user, and keeps running till the TSO user logs off. The user specific batch job that starts up handles
What datasets are available
What terminal monitor program is to be used
What procedure to auto execute at logon
TSO internally treats each terminal user as a Job. A **Job Stream** is created when a terminal user logs in, and each terminal user is given a separate address space in which datasets can be allocated and programs can be invoked.

TSO is used for the following tasks:

1. Time Sharing
2. Resource sharing
3. Handling each TSO user as batch jobs would be handled
4. The user specific batch job that starts up, handles the following:
   a. The datasets that are available
   b. The terminal monitor program that is to be used
   c. The procedure that has to auto execute at logon

Different TSO Commands:

There are about 26 commands providing a variety of functions that can be used. TSO provides function such as given below:

1. Dataset Management functions
2. Program Development functions
3. Batch job functions
4. Other functions like Help, Broadcast, CList, and REXX

These commands are issued at the READY prompt or TSO command prompt.

Some Dataset Management functions are used for the following tasks:

1. Allocate Datasets dynamically
2. List Datasets
3. Print Datasets
4. Copy Datasets
5. Delete Datasets
6. Rename Datasets
7. List Catalog Entries
8. List VTOC Entries
9. Use AMS Services

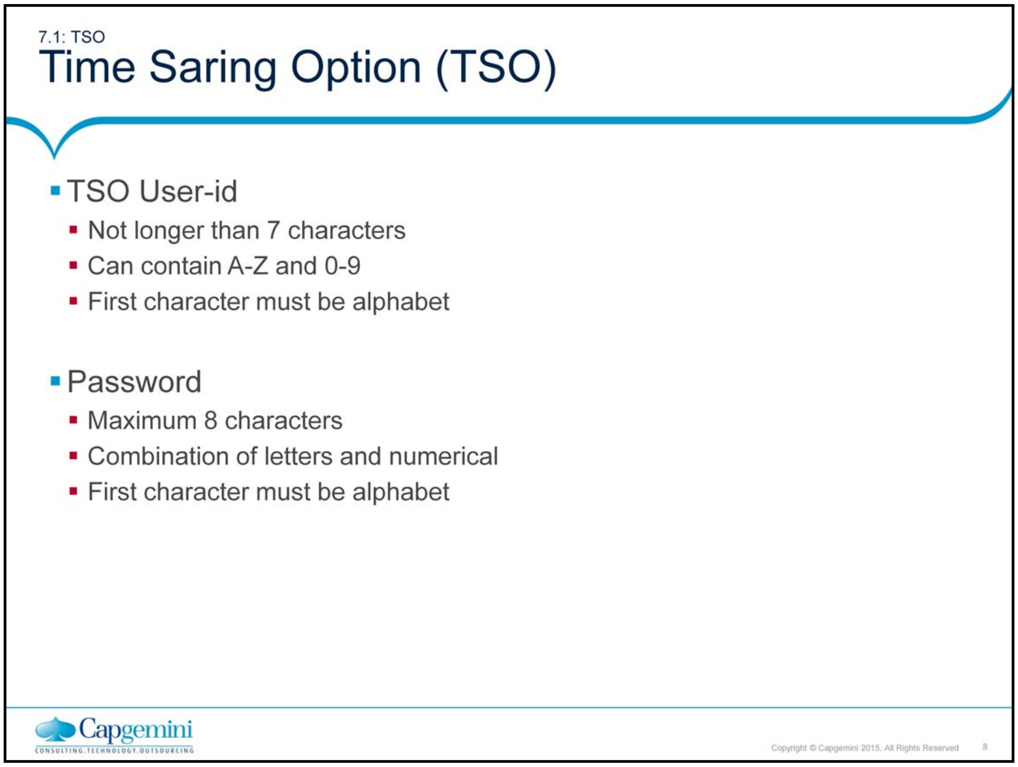
Program Development functions are used for the following tasks:

1. Create program
2. Edit program
3. Compile program
4. Link edit a program
5. View output
6. Route output to a printer

Batch Job functions include the following functions:

1. Submit jobs
2. Monitor job
3. View output
4. Route output

Help Functions on TSO commands can be obtained by typing, “HELP” at the “READY” prompt.


7.1: TSO
Time Saring Option (TSO)
TSO User-id
Not longer than 7 characters
Can contain A-Z and 0-9
First character must be alphabet
Password
Maximum 8 characters
Combination of letters and numerical
First character must be alphabet
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
8

7.1: TSO

Time Sharing Option (TSO) Contd…

- TSO FUNCTIONS
  - Commands that provides a variety of functions can be used.
  - Allow Dataset Management functions
  - Program Development functions.
  - Batch job functions.
  - Other functions like Help, Broadcast, CLIST and REXX.
  - You can issue these at the READY prompt or TSO command.

Demo

- Logon to Mainframe
- Usage of demo READY prompt on TSO region
- Logoff from Mainframe

7.2: ISPF

Use of ISPF

- ISPF stands for Interactive System Productivity Facility.
- ISPF runs as part of TSO.
- Following key functions are implemented using ISPF:
  - **Editor:** Program Sources, Job Commands
  - **Data Management:** PDS and Physical Sequential Data Set Management
  - **Job Processing:** Initiate Job, Check job log
  - Miscellaneous
  - **PDF:** Program Development Facility is part of ISPF.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ISPF:

ISPF takes advantage of full screen (24 x 80) capability of 3270 terminals as it runs part of TSO.

Panels are provided for terminal users for issuing commands.

PDF, which is part of ISPF, is used by the user/programmer to create, develop and submit new jobs.

Access to ISPF is gained by Keying ISPF at the READY prompt

This is done as default in the auto executed CLIST at startup.

Menu driven, full screen interface.

Scrollable Display

Panels

Split Screen capabilities

Capability to execute TSO commands

Submit facility

Spool / System Display Search Facility (SDSF)

Online tutorial

When this is entered you get the Primary Options Menu.

7.2: ISPF

Data Management Functions

- Dataset Management functions
  - Allocate Datasets dynamically
  - List Datasets
  - Print Datasets
  - Copy Datasets
  - Delete Datasets
  - Rename Datasets
  - List Catalog Entries
  - List VTOC Entries
  - Use AMS Services

7.2: ISPF

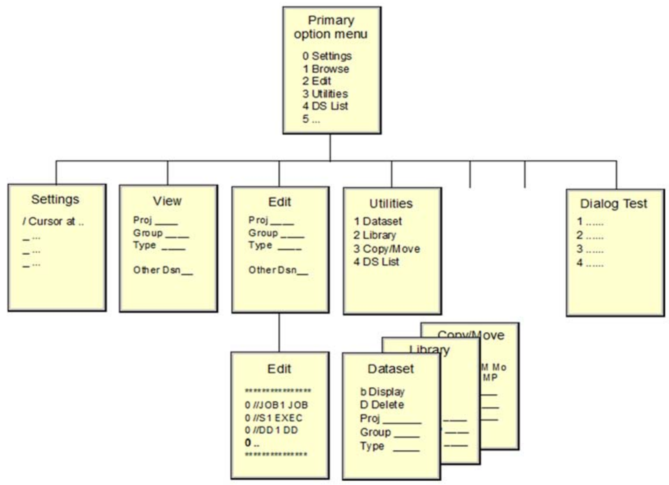
PDF & Batch Functions

- Program Development functions
  - Create program
  - Edit program
  - Compile program
  - Linkedit a program
  - View output
  - Route output to a printer
- Batch job functions
  - Submit Jobs for background processing
  - Monitor the progress of a background job
  - View output
  - Route output

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

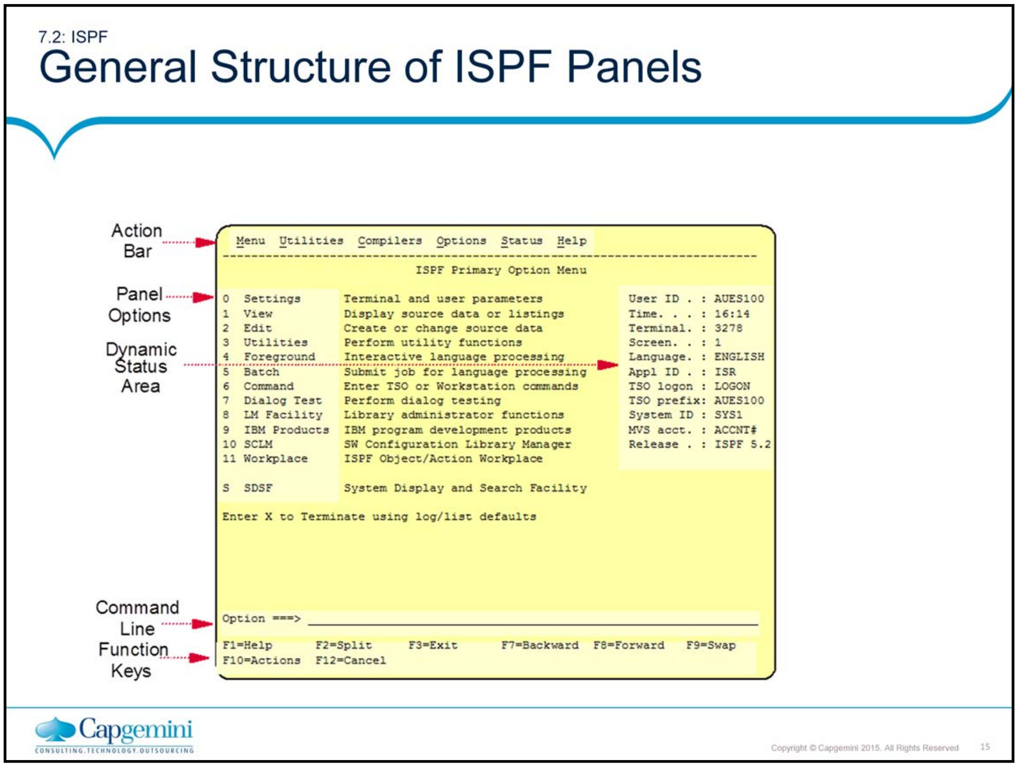
7.2: ISPF

ISPF Menu Structure



Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 14


7.2: ISPF
General Structure of ISPF Panels
Action Bar
Panel Options
Dynamic Status Area
Command Line
Function Keys
Menu Utilities Compilers Options Status Help
ISPF Primary Option Menu
0 Settings Terminal and user parameters
1 View Display source data or listings
2 Edit Create or change source data
3 Utilities Perform utility functions
4 Foreground Interactive language processing
5 Batch Submit job for language processing
6 Command Enter TSO or Workstation commands
7 Dialog Test Perform dialog testing
8 LM Facility Library administrator functions
9 IBM Products IBM program development products
10 SCLM SW Configuration Library Manager
11 Workplace ISPF Object/Action Workplace
S SDSF System Display and Search Facility
User ID . : AUES100
Time. . . : 16:14
Terminal. : 3278
Screen. . : 1
Language. : ENGLISH
Appl ID . : ISR
TSO logon : LOGON
TSO prefix: AUES100
System ID : SYS1
MVS acct. : ACCNT#
Release . : ISPF 5.2
Enter X to Terminate using log/list defaults
Option ===>
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel
Copyright © Capgemini 2015. All Rights Reserved
15

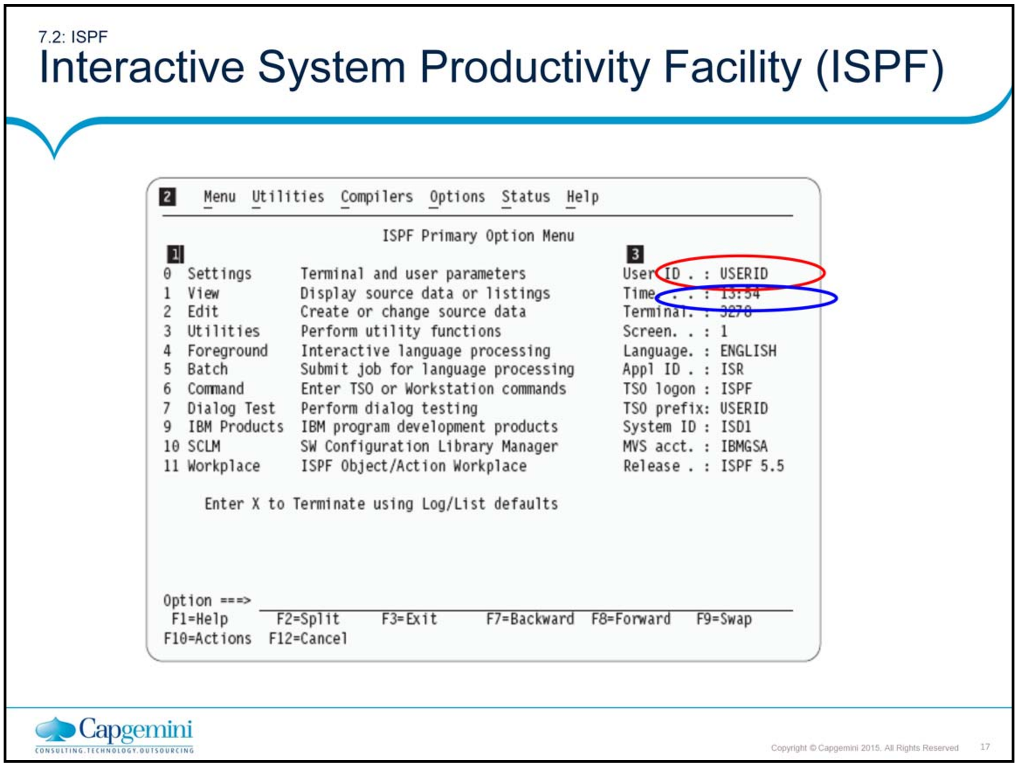
7.2: ISPF

List of Function Keys & Identifier Keys

- PA/PF Key Map

```
PF1  ===> HELP          Enter the Tutorial
PF2  ===> SPLIT         Enter Split Screen Mode
PF3  ===> END           Terminate the current operation
PF4  ===> RETURN        Return to primary options menu
PF5  ===> RFIND   Repeat find
PF6  ===> RCHANGE       Repeat Change
PF7  ===> UP            Move screen window up
PF8  ===> DOWN   Move screen window down
PF9  ===> SWAP          Activate the other logical screen in split screen
mode
PF10 ===> LEFT          Scroll screen left
PF11 ===> RIGHT  Scroll screen right
PF12 ===> RETRIEVE      Retrieve last command
PA1  ===> ATTENTION     Interrupt Current operation
PA2 ==> RESHOW Redisplay the current screen
```

PF1 - PF12 Keys may be duplicated from PF13 to PF24 in 24 key mode.


7.2: ISPF
Interactive System Productivity Facility (ISPF)
2 Menu Utilities Compilers Options Status Help
ISPF Primary Option Menu
1
3
0 Settings Terminal and user parameters
1 View Display source data or listings
2 Edit Create or change source data
3 Utilities Perform utility functions
4 Foreground Interactive language processing
5 Batch Submit job for language processing
6 Command Enter TSO or Workstation commands
7 Dialog Test Perform dialog testing
9 IBM Products IBM program development products
10 SCLM SW Configuration Library Manager
11 Workplace ISPF Object/Action Workplace
User ID . : USERID
Time. . . : 13:54
Terminal. : 3278
Screen. . : 1
Language. : ENGLISH
Appl ID . : ISR
TSO logon : ISPF
TSO prefix: USERID
System ID : ISD1
MVS acct. : IBMGSA
Release . : ISPF 5.5
Enter X to Terminate using Log/List defaults
Option ===>
F1=Help F2=Split F3=Exit F7=Backward F8=Forward F9=Swap
F10=Actions F12=Cancel
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
17

7.2: ISPF

Interactive System Productivity Facility (ISPF) Contd…

- Exiting from ISPF
  - To terminate ISPF you can
  - type =x at the command line
  - or use the PF3 key to exit
  - If you haven't specified default dispositions for your List and log datasets then the termination panel is displayed

7.2: ISPF

Interactive System Productivity Facility (ISPF) Contd…

- Editing Datasets (Option 2)
  - The Primary Editor entry is similar to that for Browse as regards concatenating datasets and dataset selection.
  - Labels can be defined as in browse but may be entered as line commands.
  - Error messages may be removed by typing RESET on the command line.

- Scroll Amounts
  - HALF Move the screen windows half a page (11 lines or 40 columns)
  - PAGE Move the screen windows one page (22 lines or 80 columns)
  - N Move the screen windows n lines or columns
  - CSR Move the screen windows at the cursor position to top, bottom, left, right
  - DATA Move the screen windows one line or one column less than a full page

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 19

7.2: ISPF

Line Commands

- Standard Line editing commands
  - C copy this line
  - Cn Copy n lines starting with this line
  - CC Copy a block of lines
  - A Place the copied lines after this line
  - An Repeat the copied lines n times after this line
  - B Place the copied lines before this line
  - Bn Repeat the copied lines n times before this line

  - D Delete line
  - Dn Deletes n lines starting with this line
  - DD Deletes the block of lines beginning with the first DD commands and ending with the second DD command

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

7.2: ISPF

Line Commands

- X Exclude this line
- Xn Exclude n lines starting with this line
- XX Exclude a block of lines

- S Show one line of the excluded text
- Sn Show n lines

- F Show the first line of the excluded text
- Fn Show the first n lines

- L Show the last line of the excluded text
- Ln Show the last n lines

7.2: Tools

Line Commands

- I Insert a single line following this line
- In Insert n lines following this line

- M Move this line
- Mn Move n lines starting with this line
- MM Move a block of lines
- A Place the moved lines after this line
- An Repeat the moved lines n times after this line

- B Place the moved lines before this line
- Bn Repeat the moved lines n times before this line.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 22

7.2: ISPF

Line Commands

- R Repeat this line
- Rn Repeat this line n times
- RR Repeat a block of lines
- RRn Repeat a block of lines n times

- Shifting text source

Data shift	Column shift	Meaning
<	(	Shift this line left 2 position
<n	(n	Shift this line n position left
<<	((	Shift a block of lines left
< <n	((n	Shift a block of lines left
>	)	Shift this line right 2 position
>n	)n	Shift this line n position right
>>	))	Shift a block of lines right
>>n	))n	Shift a block of lines right

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 23

7.2: ISPF

Line Commands

- TE Text Edit
- TF Text Flow
- TS Text Split
- LC Lower Case
- UC Upper Case

- Other Line Commands
  - COLS Display column line
  - TABS Setting up a tab positions
  - MASKS Display a mask line. Used for repetitive insertion of information
  - BOUNDS Display boundary line

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 24

7.2: ISPF

Primary Commands

- Primary Commands/Command line commands
  - CANCEL
  - CAPS ON / OFF
  - LOCATE — To locate a dataset
  - TSO SUBMIT — To execute
  - SORT — Sorts the dataset list based on the fields shown on the next transparency
  - FIND — Finds occurrence of a string with the list of datasets
  - SAVE DSN — Saves the current dataset
  - COPY — COPY member [AFTER / BEFORE]
  - CREATE — CREATE [member]
  - DELETE — DELETE ALL
  - EDIT — EDIT [member Name]
  - CHANGE — CHANGE str1 str2 [range] ALL

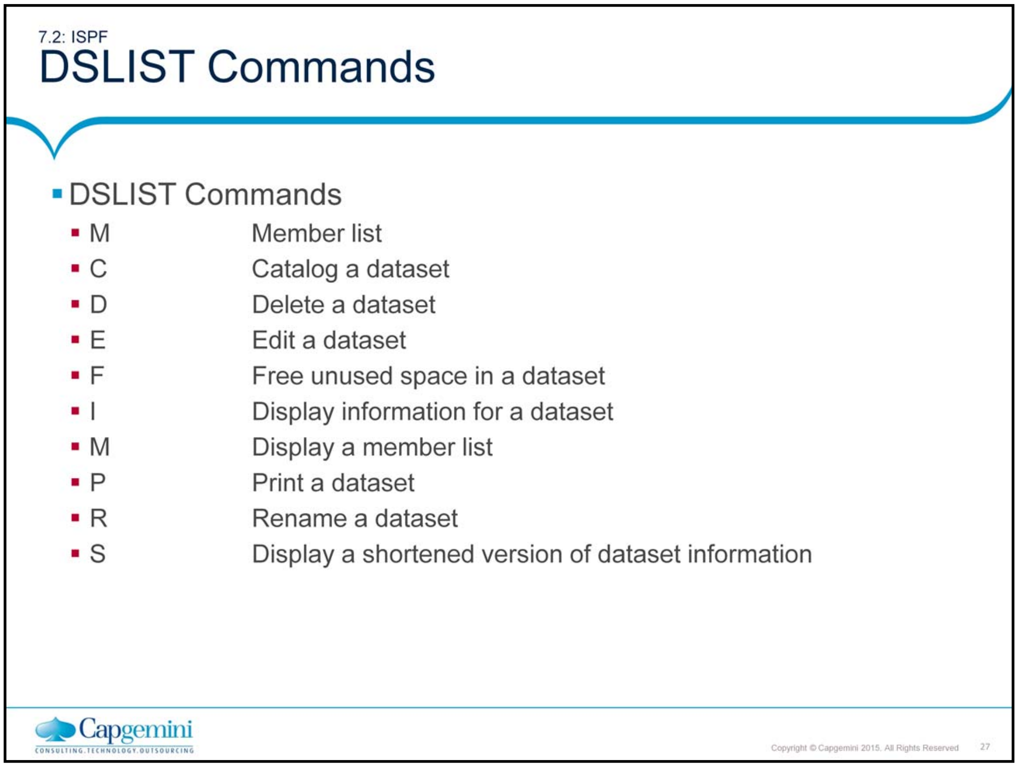
 25

7.2: ISPF

Primary Commands

- Primary Commands
  - FIND FIND str1
  - HEX HEX ON / OFF
  - LOCATE LOCATE [NUMBER]
  - SAVE
  - UNDO
  - UNNUM
  - RENUM
  - RESET
  - REPLACE REPLACE [memebr Name]

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING


7.2: ISPF
DSLIST Commands
DSLIST Commands
M Member list
C Catalog a dataset
D Delete a dataset
E Edit a dataset
F Free unused space in a dataset
I Display information for a dataset
M Display a member list
P Print a dataset
R Rename a dataset
S Display a shortened version of dataset information
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 27

7.2: ISPF

DLIST Commands

- DSLIST Commands
  - U Uncatalog a dataset
  - X Print a dataset indexed listing
  - Z Compress a dataset
  - = Repeat the last command
- MEMBER SELECTION LIST Commands
  - S Select Member
  - D Delete Member
  - B Browse Member
  - E Edit Member

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

7.2: ISPF

ISPF Main Menu

- TSO / ISPF MAIN Menu

```
----------------------- ISPF/PDF PRIMARY OPTION MENU ------------------------

OPTION ===> pfshow
                                                                       USERID   - iGTRN01
   0  ISPF PARMS     - Specify terminal and user parameters           TIME     - 06:58
   1  BROWSE         - Display source data or output listings         TERMINAL - 3278
   2  EDIT           - Create or change source data                   PF KEYS  - 12
   3  UTILITIES      - Perform utility functions
   4  FOREGROUND     - Invoke language processors in foreground
   5  BATCH          - Submit job for language processing
   6  COMMAND        - Enter TSO Command, CLIST, or REXX exec
   7  DIALOG TEST    - Perform dialog testing
   8  LM UTILITIES   - Perform library administrator utility functions
   9  IBM PRODUCTS   - Additional IBM program development products
  10  SCLM           - Software Configuration and Library Manager
   C  CHANGES        - Display summary of changes for this release
   T  TUTORIAL       - Display information about ISPF/PDF
   X  EXIT           - Terminate ISPF using log and list defaults
   D  DATACENTER     - Perform Datacenter Defined Functions
   S  SDSF           - Spool Display and Search Facility
   U  USER           - Perform User Defined Functions
 F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=RFIND    F6=RCHANGE
 F7=UP      F8=DOWN     F9=SWAP   F10=LEFT     F11=RIGHT   F12=RETRIEVE
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

7.2: ISPF

ISPF Utilities Menu

- ISPF Option 3

```
------------------------  UTILITY SELECTION MENU  ----------------------------
 OPTION  ===>
   1  LIBRARY          -  Compress or print data set.  Print index listing.
                          Print, rename, delete, browse, or edit members
   2  DATASET          - Allocate, rename, delete, catalog, uncatalog, or
                          display information of an entire data set
   3  MOVE/COPY  - Move, copy, or promote members or data sets
   4  DSLIST           - Print or display (to process) list of data set names
                         Print or display VTOC information
   5  RESET            - Reset statistics for members of ISPF library
   6  HARDCOPY   - Initiate hardcopy output
   8  OUTLIST          -     Display, delete, or print held job output
   9  COMMANDS         - Create/change an application command table
  10  CONVERT          - Convert old format menus/messages to new format
  11  FORMAT           - Format definition for formatted data Edit/Browse
  12  SUPERC           - Compare data sets  (Standard Dialog)
  13  SUPERCE          - Compare data sets and Search-for strings (Extended Dialog)
  14  SEARCH-FOR - Search data sets for strings of data (Standard Dialog)
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 30

7.2: ISPF

ISPF Utilities Menu

- ISPF Library Utility (Option 3.1)

```
---------------------------- LIBRARY UTILITY --------------------------------
OPTION  ===>
   blank - Display member list              B - Browse member
   C - Compress data set             P - Print member
   X - Print index listing           R - Rename member
   L - Print entire data set         D - Delete member
   I - Data set information          E - Edit member
                                      S - Data set information (short)
ISPF LIBRARY:
   PROJECT ===> DA0034T
   GROUP   ===> TRG      ===>       ===>       ===>
   TYPE    ===> JCL
   MEMBER  ===>                (If  "P", "R", "D", "B", "E" or blank selected)
   NEWNAME ===>                (If  "R" selected)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:

   DATA SET NAME  ===>
   VOLUME SERIAL  ===>        (If not cataloged)
DATA SET PASSWORD ===>        (If password protected)
```

7.2: ISPF

ISPF Utilities Menu

- ISPF Dataset Utility (Option 3.2)

```
----------------------------  DATA SET UTILITY  -------------------------------
 OPTION  ===> A

   A - Allocate new data set                C - Catalog data set
   R - Rename entire data set                 U - Uncatalog data set
   D - Delete entire data set                 S - Data set information (short)
   blank - Data set information             M - Enhanced data set allocation

 ISPF LIBRARY:
   PROJECT ===> DA0034T
   GROUP   ===> TRG
   TYPE    ===> JCL

 OTHER PARTITIONED OR SEQUENTIAL DATA SET:

   DATA SET NAME  ===>
   VOLUME SERIAL  ===>        (If not cataloged, required for option "C")
 DATA SET PASSWORD ===>         (If password protected)
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 32

7.2: ISPF

ISPF Utilities Menu

- New dataset allocation (option 3.2.A)

```
------------------------ ALLOCATE NEW DATA SET ------------------------------
 COMMAND ===>
 DATA SET NAME: DA0034T.TRG.JCLS
   VOLUME SERIAL          ===> BS3008     (Blank for authorized default volume)
   GENERIC UNIT           ===>            (Generic group name or unit address)
   SPACE UNITS            ===> BLOCK      (BLKS, TRKS, or CYLS)
   PRIMARY QUANTITY       ===> 26         (In above units)
   SECONDARY QUANTITY     ===> 12         (In above units)
   DIRECTORY BLOCKS       ===> 0          (Zero for sequential data set)
   RECORD FORMAT          ===> FB
   RECORD LENGTH          ===> 150
   BLOCK SIZE             ===> 1500
   EXPIRATION DATE        ===>          (YY/MM/DD, YYYY/MM/DD
              YY.DDD, YYYY.DDD in Julian form  DDDD for retention period in days or blank)
                     ( * Only one of these fields may be specified)
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

7.2: ISPF

ISPF Utilities Menu

- Renaming Dataset (Option 3.2.R)

```
---------------------------- RENAME DATA SET --------------------------------
COMMAND ===>

DATA SET NAME: DA0034T.TRG.JCL
VOLUME:        BS3008
ENTER NEW NAME BELOW:     (The data set will be recataloged.)

ISPF LIBRARY:
   PROJECT ===> DA0034T
   GROUP   ===> TRG
   TYPE    ===> JCL

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
   DATA SET NAME  ===>
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 34

7.2: ISPF

ISPF Utilities Menu

- Dataset information (Option 3.2.s)

```
-------------------------- DATA SET INFORMATION -----------------------------
COMMAND ===>
DATA SET NAME: DA0034T.TRG.JCL
GENERAL DATA:                                   CURRENT ALLOCATION:
  Management class:                      Allocated blocks:          26
  Storage class:                         Allocated extents:          1
    Volume:             BS3008     Maximum dir. blocks:           1
    Device type:        3390
  Data class:
    Organization:       PO                CURRENT UTILIZATION:
    Record format:      FB                      Used blocks:                  11
    Record length:      150                     Used extents:                  1
    Block size:        1500                     Used dir. blocks:              1
    1st extent blocks:    26                    Number of members:             5
    Secondary blocks:     12
    Data set name type:  PDS
    Creation date:      1996/08/08
    Expiration date:    ***NONE***
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 35

7.2: ISPF

ISPF Utilities Menu

- Allocate datasets managed by SMS

```
          ------------------------  ALLOCATE NEW DATA SET  ------------------------------
COMMAND ===>
DATA SET NAME: DA0034T.TRG.JCL
  MANAGEMENT CLASS       ===> MCSTANDS      (Blank for default management class)
  STORAGE CLASS          ===> SCNORM        (Blank for default storage class)
  VOLUME SERIAL          ===>                      (Blank for authorized default volume)
  DATA CLASS             ===>                      (Blank for default data class)
  SPACE UNITS            ===> BLOCK         (BLKS, TRKS, CYLS, KB, MB or BYTES)
  PRIMARY QUANTITY       ===> 26                   (In above units)
  SECONDARY QUANTITY     ===> 12                   (In above units)
  DIRECTORY BLOCKS       ===> 1                    (Zero for sequential data set) *
  RECORD FORMAT          ===> FB
  RECORD LENGTH          ===> 150
  BLOCK SIZE             ===> 1500
  DATA SET NAME TYPE     ===> PDS                  (LIBRARY, PDS, or blank)     *
  EXPIRATION DATE        ===>                      (YY/MM/DD, YYYY/MM/DD
                                            YY.DDD, YYYY.DDD in Julian form
                                            DDDD for retention period in days
                                            or blank)
   (* Specifying LIBRARY may override zero directory block)
```

 36

7.2: ISPF

ISPF Utilities Menu

- Move / Copy (Option 3.3)

```
---------------------------  MOVE/COPY UTILITY  -------------------------------
OPTION  ===>
   C - Copy data set or member(s)                  CP - Copy and print
   M - Move data set or member(s)                  MP - Move and print
   L - Copy and LMF lock member(s)                 LP - Copy, LMF lock, and print
   P - LMF Promote data set or member(s)           PP - LMF Promote and print

SPECIFY "FROM" DATA SET BELOW, THEN PRESS ENTER KEY

FROM ISPF LIBRARY:        ------ Options C, CP, L, and LP only -------
   PROJECT   ===> DA0034T
   GROUP     ===> TRG      ===>       ===>       ===>
   TYPE      ===> JCL
   MEMBER    ===>             (Blank or pattern for member selection list,
                          '*' for all members)
FROM OTHER PARTITIONED OR SEQUENTIAL DATA SET:
   DATA SET NAME      ===>
   VOLUME SERIAL      ===>        (If not cataloged)
DATA SET PASSWORD ===>            (If password protected)
```

7.2: ISPF

ISPF Utilities Menu

- Move / Copy (Option 3.3)

```
COPY --- FROM DA0034T.TRG.JCL ------------------------------------------------
 COMMAND ===>

 SPECIFY "TO" DATA SET BELOW.

 TO ISPF LIBRARY:
    PROJECT ===> DA0034T
    GROUP     ===> TRG
    TYPE        ===> JCL

 TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:
    DATA SET NAME  ===>
    VOLUME SERIAL  ===>         (If not cataloged)
 DATA SET PASSWORD ===>         (If password protected)

 "TO" DATA SET OPTIONS:
    IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS ===> YES    (YES or NO)
    IF SEQUENTIAL, "TO" DATA SET DISPOSITION          ===> OLD    (OLD or MOD)
    SPECIFY PACK OPTION FOR "TO" DATA SET              ===>   (YES, NO or blank)
```

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

 38

7.2: ISPF

ISPF Utilities Menu

- DSLIST Utility (Option 3.4)

```
--------------------------- DATA SET LIST UTILITY -----------------------------

OPTION  ===>

  blank - Display data set list *                 P   - Print data set list
  V     - Display VTOC information only    PV - Print VTOC information only

Enter one or both of the parameters below:
  DSNAME LEVEL  ===> DA0034T.TRG.*
  VOLUME        ===>

  INITIAL DISPLAY VIEW     ===> VOLUME   (VOLUME,SPACE,ATTRIB,TOTAL)
  CONFIRM DELETE REQUEST   ===> YES      (YES or NO)
* The following line commands will be available when the list is displayed:

B - Browse data set       C - Catalog data set       F - Free unused space
E - Edit data set         U - Uncatalog data set     = - Repeat last command
D - Delete data set       P - Print data set
R - Rename data set       X - Print index listing
I - Data set information  M - Display member list
S - Information (short)   Z - Compress data set     TSO cmd, CLIST or REXX exec
```

 39

7.2: ISPF

ISPF Utilities Menu

- DSLIST Dataset Selection

```
DSLIST - DATA SETS BEGINNING WITH DA0034T.TRG.* ----------------- ROW 1 OF 23

COMMAND ===>                                   SCROLL ===> PAGE

COMMAND   NAME                          MESSAGE      VOLUME
-------------------------------------------------------------------------------
          DA0034T.TRG.ACCOUNT           MIGRAT
          DA0034T.TRG.BADCOBOL          MIGRAT
          DA0034T.TRG.COBOL             MIGRAT
          DA0034T.TRG.COBOL1            MIGRAT
m         DA0034T.TRG..JC               BS3008
          DA0034T.TRG.LNK               MIGRAT
```

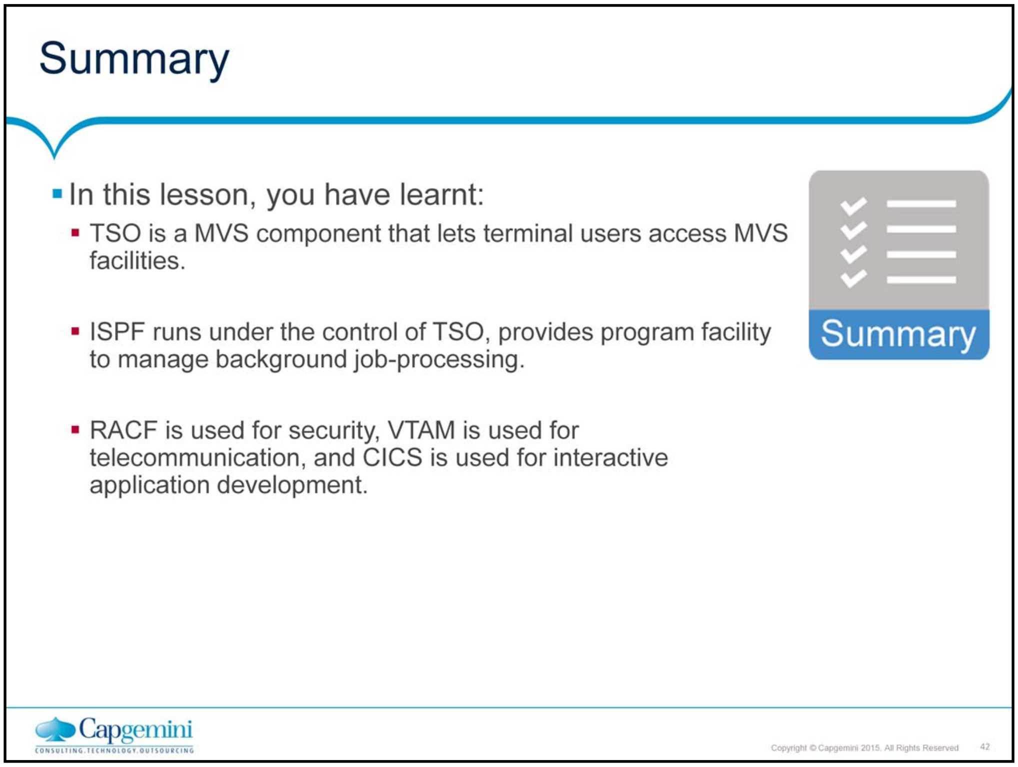
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 40

Demo

- Working with TSO and ISPF

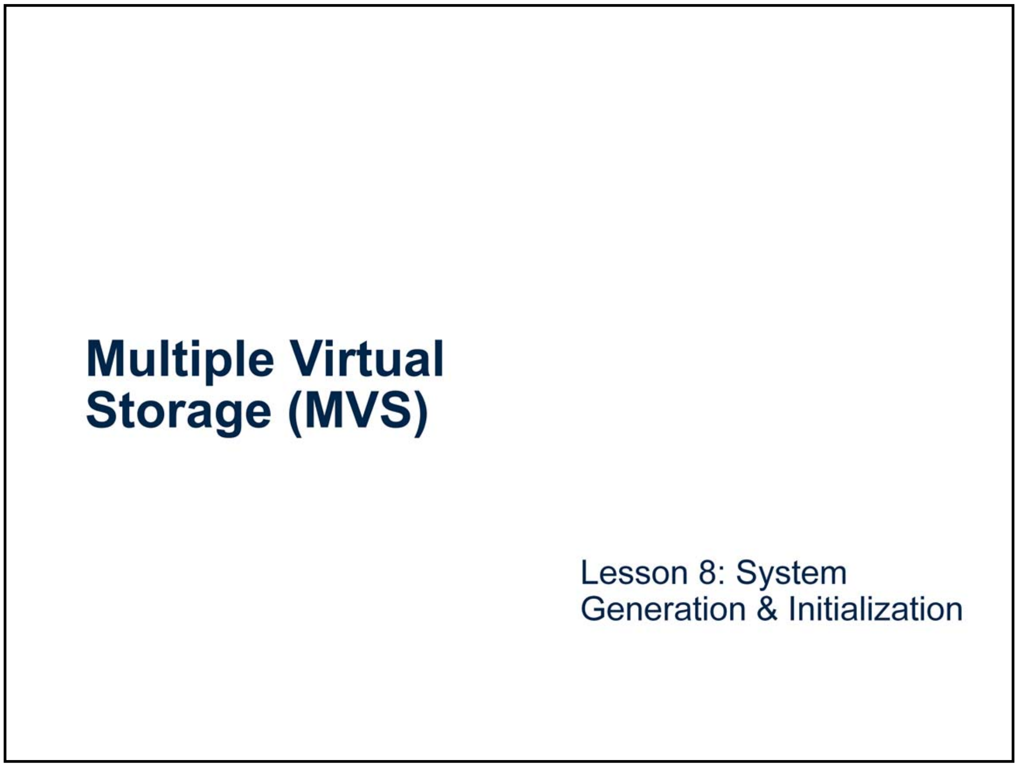
 41


Summary
In this lesson, you have learnt:
TSO is a MVS component that lets terminal users access MVS facilities.
ISPF runs under the control of TSO, provides program facility to manage background job-processing.
RACF is used for security, VTAM is used for telecommunication, and CICS is used for interactive application development.
Summary
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
42

Review Question: Match the Following

1. VTAM	a. TSO
2. CICS	b. Security Package
3. RACF	c. Utility program
4. ISPF	d. TC Access Method
5. SORT	e. Interactive Applications

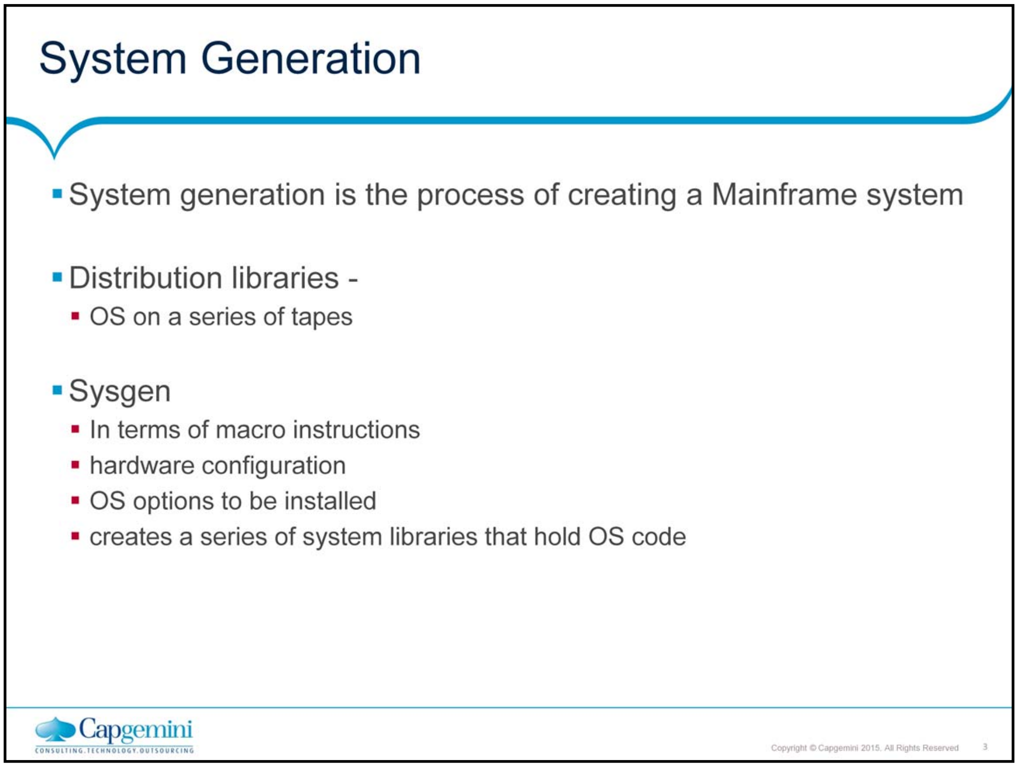
Capgemini
CONSULTING. TECHNOLOGY. OUTSOURCING


Multiple Virtual
Storage (MVS)
Lesson 8: System
Generation & Initialization

Lesson Objectives

- In this lesson, you will learn the following topics:
  - System Generation & Initialization
  - System Datasets
  - ISRDDN diagnostic utility

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



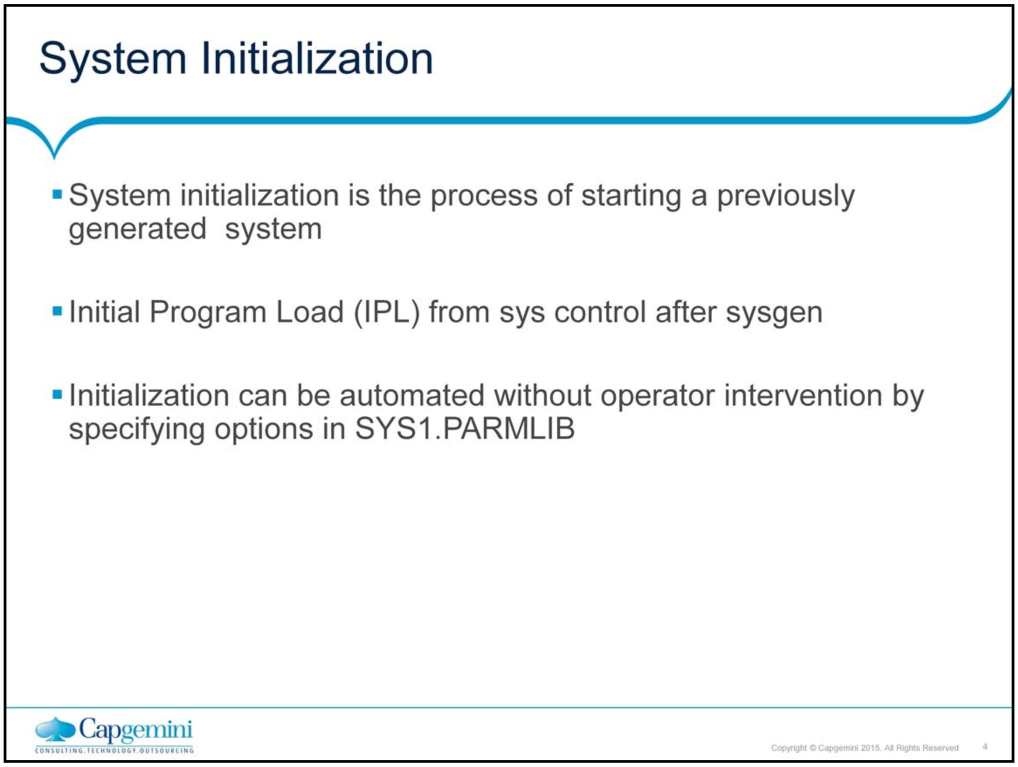
The basic components that make up the OS are on a series of tapes, called distribution libraries. System generation selects and assembles the various distribution libraries. To control system generation, normally known as sysgen, a system's programmer codes special macroinstructions that specify how the OS components from the distribution libraries have to be put together.

An installation must have a working OS before it can create a new one since a working OS is required to execute the macroinstructions. Sysgen is usually used to upgrade to a newer version or to make changes to the current version. For installations which does not have already a working OS, a small, limited functional OS is setup that can execute the Sysgen for the complete full functional Mainframe system.

The macroinstructions that Sysgen uses fall under two categories:

The first category of macros defines the system hardware configuration. They are needed because the OS must know about every I/O device that is attached to the system. As a result for every I/O added, the system must be generated again. (Actually smaller, less time-consuming type of Sysgen called an **iogen** can be used to change the device configuration).

The second category of macroinstructions in a sysgen indicates which options of the operating system should be included. They indicate whether JES1 or JES2 is used, what optional access methods are installed, and so on. The output from a sysgen is a set of system libraries that among other things contains the executable code that makes up the operating system.



System initialization is a process by which the OS system code is loaded from selected system libraries into central storage. This process defines the Mainframe system.

The system requires initializing:

- after generating a new system.
- after changes have been made to the system.
- after a system failure.

System Datasets

- SYS1.NUCLEUS
  - PDS containing nucleus program
  - One of the members contains a pointer to the Master Catalog
- SYS1.PARMLIB
  - Options for initialization
  - Contains about 30 members that specify various options which is used during System initialization.
- SYS1.LINKLIB
  - Contains mostly OS executables
  - Contains executable programs that are written by users in COBOL.
  - These are not read into storage until they are required

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

System Datasets (Contd..)

- SYS1.LPALIB
  - Contains executable programs that are part of the operating system.
  - These are always available in storage for any program that needs them.
- SYS1.PROCLIB :
  - PDS containing JCL procedures
  - These can be used by any valid JCL user.
- SYS1.CMDLIB
  - Commands for TSO mode

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

ISRDDN diagnostic utility

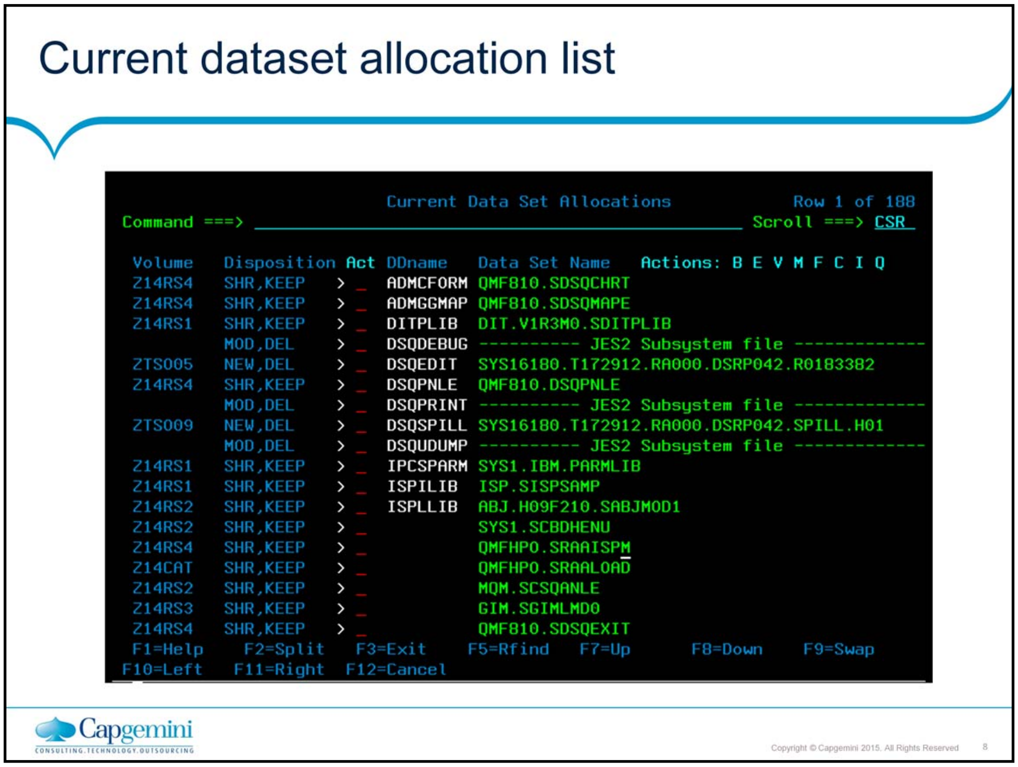
- ISRDDN is a utility, that provides a list of allocated DD names, a list of system ENQs, a list of data set causing ENQ contention, and a means of viewing storage within a TSO user's address space.
- ISRDDN is also used to provide some facilities to gather information about the environment, that we work on.
- To invoke the ISRDDN program, type TSO ISRDDN on any of the ISPF command line.

The allocated DD name list shows you all of the DD names allocated to your TSO session. From the list you can perform functions such as Edit or Compress against individual data sets, DD names, or sets of DD names. You can also perform actions against the entire list of displayed DD names.

The ENQ list, which is available by typing ENQ on the allocation list command line, shows you ENQs on your system. You can limit the size of the list by specifying the QNAME, RNAME, job, user or address space name, and system name.

The ENQ contention list, available by typing CON on the allocation list command line, shows you ENQ contentions on your system for data sets (QNAME SYSDSN).

You can Browse storage using the BROWSE primary command from the allocation list. You can only browse storage which an unauthorized program can see (private and common).



When you start ISRDDN, you will see the Current Data Set Allocations list. On the right side of the display is a list of DD names and their associated data sets. The list of data sets may also contain indicators of DUMMY allocations, subsystem files or allocations to the terminal. The DD name is shown in white unless the first data set in the concatenation is scrolled off the top of the screen. If the first data set in a concatenation is not on the screen, the DD name will be yellow.

In the center of the display is a column of 1 character input fields preceded by greater-than signs (>). These input fields are used for line commands such as **E** for edit or **I** for information.

The left side of the display contains columns of information about individual data sets. When you scroll right or left, the left side of the screen will change.

Initially the left side of the screen will contain the volume name and disposition. If the disposition is red, then there are other jobs waiting to use this data set as shown. Use the Q line command to see what jobs are waiting. Also, you can view VTOC information for a volume by placing the cursor on the volume name and pressing the Enter key.

Data set attributes in ISRRDN

```
                          Current Data Set Allocations                  Row 1 of 188
Command ===> _                                                       Scroll ===> CSR

Blksz Lrecl RCFM Org  Act DDname   Data Set Name    Actions: B E V M F C I Q
27600   400 FB   PO   > _ ADMCFORM QMF810.SDSQCHRT
27600   400 FB   PO   > _ ADMGGMAP QMF810.SDSQMAPE
 8800    80 FB   PO   > _ DITPLIB  DIT.V1R3M0.SDITPLIB
                      > _ DSQDEBUG ---------- JES2 Subsystem file -------------
 4029    79 FBA       > _ DSQEDIT  SYS16180.T172912.RA000.DSRP042.R0183382
    0     0      VS   > _ DSQPNLE  QMF810.DSQPNLE
                      > _ DSQPRINT ---------- JES2 Subsystem file -------------
 4096  4096 F         > _ DSQSPILL SYS16180.T172912.RA000.DSRP042.SPILL.H01
                      > _ DSQUDUMP ---------- JES2 Subsystem file -------------
 6160    80 FB   PO   > _ IPCSPARM SYS1.IBM.PARMLIB
 8800    80 FB   PO   > _ ISPILIB  ISP.SISPSAMP
32760    ** U    PO   > _ ISPLLIB  ABJ.H09F210.SABJMOD1
32760    ** U    PO   > _          SYS1.SCBDHENU
27920    80 FB   PO   > _          QMFHPO.SRAAISPM
32760    ** U    PO   > _          QMFHPO.SRAALOAD
32760    ** U    PO   > _          MQM.SCSQANLE
32760    ** U    PO   > _          GIM.SGIMLMD0
32760    ** U    PO   > _          QMF810.SDSQEXIT
 F1=Help     F2=Split    F3=Exit     F5=Rfind    F7=Up       F8=Down     F9=Swap
F10=Left    F11=Right   F12=Cancel
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

If you scroll right once, you will see the attributes of each dataset. For some types of allocations, such as subsystem allocations, you may see different information.

Allocation List Line Commands:

Allocation list line commands are entered next to a DD name or data set. By default the allocation list is in *short* format. This means that for concatenations, the DD name is next to the first data set name in the concatenation.

E - Edit

The E line command edits a data set or concatenation. It can be used on any data set or any DD name allocated to a data set (real or VIO). You might want to use the E line command for editing temporary files such as JCL which was created by file tailoring and written to the ISPCTLn DD name.

B - Browse

The B line command browses a data set or concatenation. It can be used on any data set or any DD name allocated to a data set (real or VIO). You can use the B line command for browsing allocated files. For example, the compress option in the PDF utilities, option 3.1, creates a listing data set which is sometimes allocated to the ISPCTL1. When you press the HELP key after compressing a data set in option 3.1, you may see that the listing was saved in a temporary data set. The B line command in ISRDDN is an easy way to browse that data set.

V - View

Use the V command to view a data set or concatenation. This is similar to E (edit) but there is no SAVE command. Use this when you want to view a data set and modify it for easier viewing without risking changes to the data set.

M - Member list

The M command displays an enhanced member list for a data set or concatenation. This allows you greater flexibility in working with allocated data sets. You might use this command when you have several different operations to perform on members.

F - Free

Use the FREE command to free an allocation. The Free command must be specified next to a DD name, although F commands next to data sets in a concatenation with an F next to the DD name are ignored because those data sets are removed from the list before the F commands are processed.

C or Z - Compress

Use the Compress command to compress partitioned data sets. The compress command can be used with data sets that are allocated as shared and can be used next to data set names or DD names.

I - Information

The I command attempts to invoke the PDF data set information utility to display information about a data set. It can be used next to any real data set name. VIO data sets are not supported. This command can provide information such as the number of allocated directory blocks or a data set's SMS management class, or other information that is not shown by scrolling the Current Data Set Allocations list left or right.

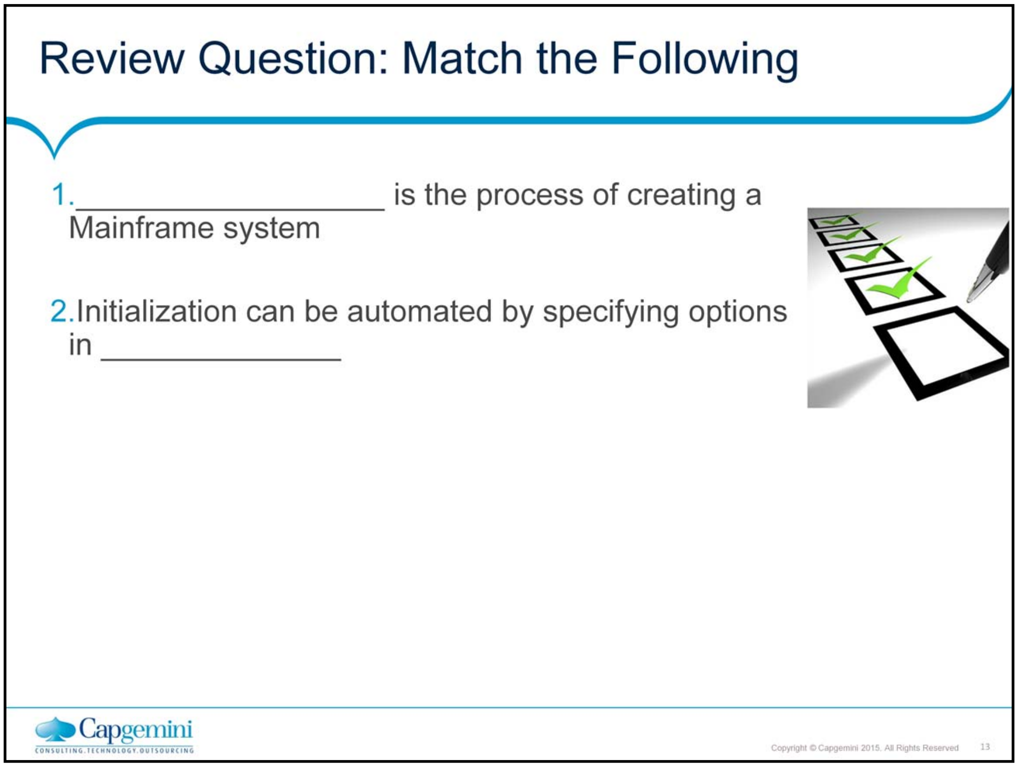
Q - Query ENQs

The Q command will show all SYSDSN and SPFEDIT ENQs that exist for a data set. This command is useful when you want to see what other users or jobs are using a data set you have allocated. Using the Q command provides the same information as using the ENQ primary command and selecting an RNAME of the data set name.

Summary

- In this lesson, you have learnt:
  - System generation & initialization
  - System datasets

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



1. System Generation
2. SYS1.PARMLIB

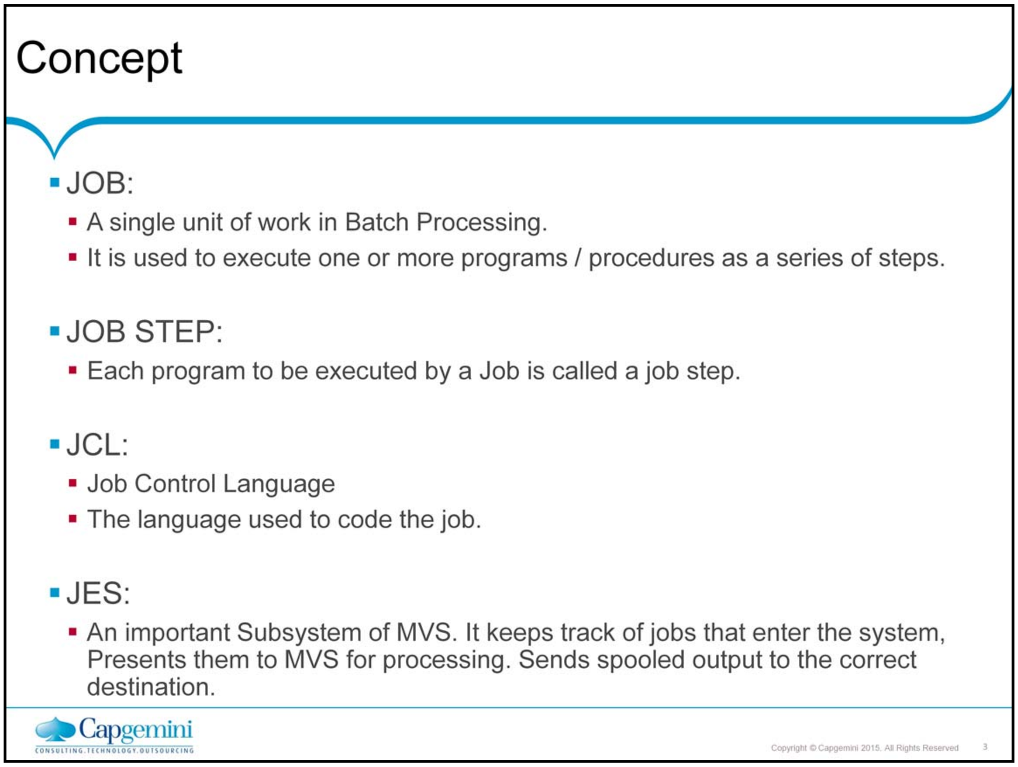
Multiple Virtual Storage

Lesson 9 : Job Management Overview

Lesson Objectives

- In this lesson, you will learn the following topics:
  - The basic concept and functions of Job
  - Various stages of a Job
  - Job Management
  - Job Scheduling

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



What is a Job?

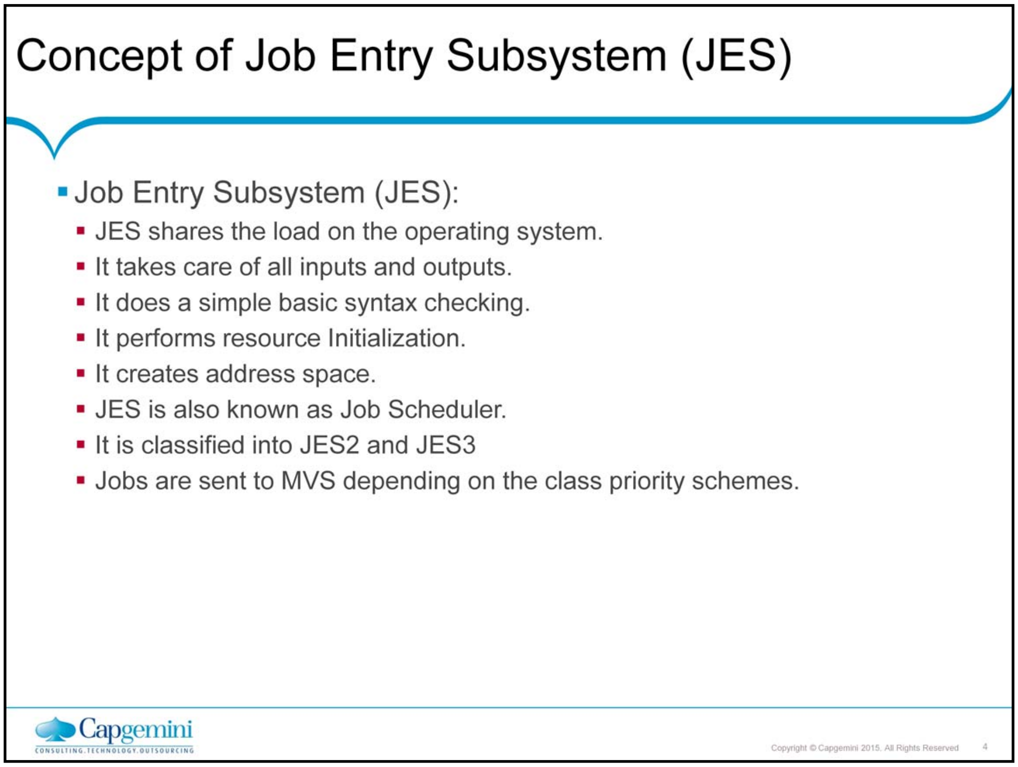
- JCL is the most dreaded word for a newcomer to IBM world.
- Job is execution of one or more related programs in pre-defined sequence.
  - Example: A job of creating an executable module (load module) from a source program consists of executing Compiler program and Linker program in sequence.

Why JCL?

- Since the job is executed in the background, without user interaction, *all information required for the execution must be supplied in advance.*
- JCL is used to specify this control information.
- The most common information supplied through JCL is as follows:
  - ➢ To whom the job belongs (which user id)?
  - ➢ What is the program / utility that is to be executed?
  - ➢ Where (in which library / PDS) do you find the load module of the program or utility?
  - ➢ Where (which DASD volume / catalog, what data set name) do you find the input data files for the program / utility?
  - ➢ Where should (which DASD volume, what data set name) the output files be created?
  - ➢ The printer output should be directed to which printer?

What is JCL?

- JCL stands for Job Control Language.
- Connotation is set for job commands stored as MEMBER in a PDS, For example: JCL to execute a batch program, JCL to compile and link a COBOL program, JCL to allocate a VSAM data set, JCL to SORT and MERGE two Physical Sequential Data Sets.
- Thus JCL is nothing but a set of commands.
- User keys-in commands using a editor, and saves as PDS Member For example: PAYROLL.TEST.JCL(PROG1JCL)
- Good grasp of JCL is a must to be a versatile IBM programmer.

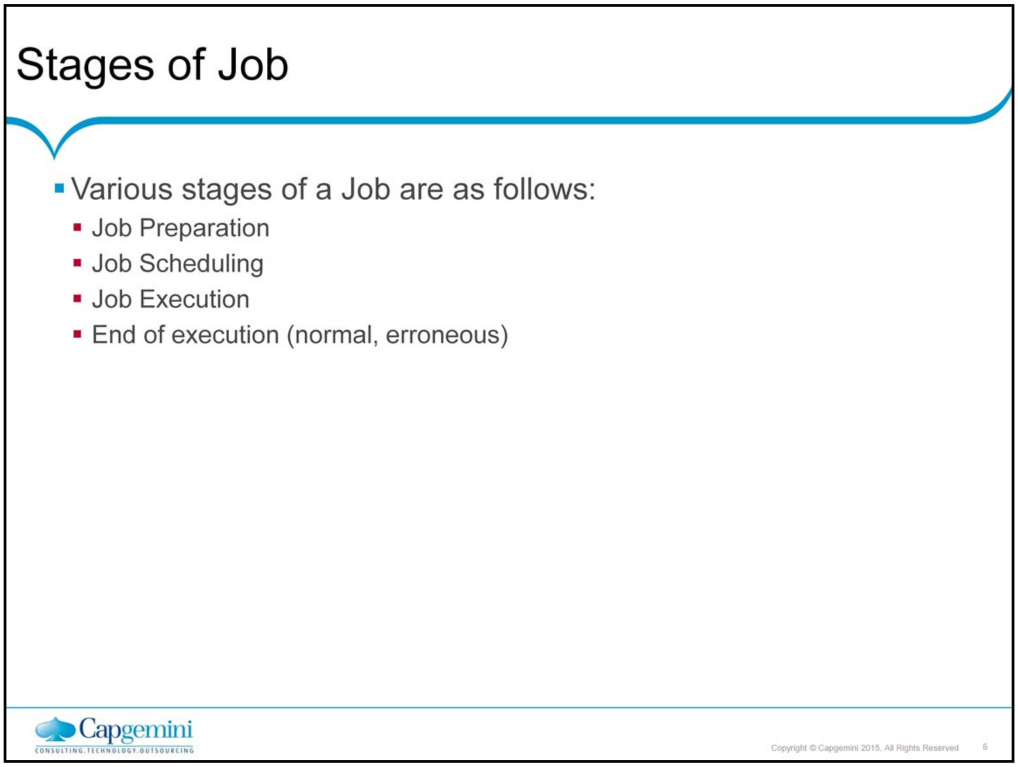


Job Management:
Job entry Subsystem (JES):

- JES is classified into JES2 and JES3:
  - **JES2** is primarily designed for single processor systems / uni-processor environment.
  - **JES3** is designed for multiple processor systems / multiprocessing environment (decided at the time of system initialization)
- Each MVS system uses either JES2 or JES3.
- JES3 has additional scheduler functions than JES2 (for example: schedule job at specific time of the day, interdependent job scheduling).

Job Management Functions

- Job Management Functions are listed below:
  - Receive the job into operating system
  - Schedule the job for processing by O/S
  - Execute the Job
  - Process the output



- Job Preparation:
  - User keys-in commands using Editor.
  - User saves it as a member in PDS.

- Job Scheduling:
  - It is initiated using TSO SUBMIT command.
  - It is not necessarily done on FIFO basis.
  - Prioritization is implemented using class and priority code.
- Job Execution:
  - It involves the actual execution of the job by a processor.

- End of execution (normal, erroneous):
  - Intimate the user
  - Job log management
  - Job output management
  - Printer output
  - Data set output

Erroneous Termination of a Job

- Type of execution errors include the following:
  - Incorrect commands (command syntax errors)
  - Required resources (Data Sets, Program Library, Program Load Module) not available
  - Violation of access permissions for data sets, program load module, and so on.
  - Mismatch in data set status, as required by job and as it actually exists, for example: a create is issued for a data set which already exists.
  - Program errors
  - Mismatch for Dataset - between program definition and actual characteristics
  - Infinite loop
  - Data Type mismatch - numeric variable contains non-numeric data
- Any abnormal termination of program is called as “Abend”.

How does a Job enter into the System?

- To enter, or submit, the job into the system, the terminal user issues a SUBMIT command.
- When you submit the job, JES reads the job stream (sequence of JCL commands) from a DASD file and copies it to a job queue, which is a part of a special DASD file called JES SPOOL.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Note: The JES component that processes the input job stream is called an **internal reader**.

How is a Job Scheduled for Execution

- MVS does not necessarily process jobs in the order in which they are submitted.
  - Instead, JES examines the jobs in the job queue and selects the most important jobs for execution. That way JES can prioritize its work, giving preference to more important jobs.
- JES uses two characteristics to classify a job's importance, both of which can be specified in the job's JCL:
  - Job Class
  - Job Priority

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 9

How is a Job Scheduled for Execution?

- If two or more jobs are waiting to be executed, then the JES scheduler selects the one with higher priority.
- Each job class is represented by a single character, either a letter (A-Z) or a digit (0-9).
  - Job classes are assigned based on the processing characteristics of the job.

How is a Job Scheduled for Execution?...contd

- INITIATOR: An initiator is a program that runs in the system region of an address space.
  - Each initiator can handle one job at a time.
    - It examines the JES spool, selects an appropriate job for execution, executes the job in its address space, and returns to the JES spool for another job.
  - The number of active initiators on a system and as a result the number of address spaces eligible for batch job processing determine the number of batch jobs that can be multiprogrammed at once.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

How is a Job Scheduled for Execution?...contd

- Each initiator has one or more job classes associated with it. It executes jobs only from those classes.

Initiator	Eligible Job Classes
1	A
2	B,C,D,H,L,T
3	B,C,D,H,L,T
4	B,C
5	B,C
6	C

How is a Job Scheduled for Execution?...contd

- Within a job class, initiator selects jobs for execution based on their priorities, which can range from 0 to 15.
- If two or more jobs have same class and priority, then they are executed in the order in which they are submitted.

How is a Job Scheduled for Execution?...contd

- Once an initiator has selected a job for execution, it invokes a program called the interpreter.
- The interpreter's job is to examine the job information passed to it by JES and create a series of control blocks in the SWA, a part of the address space's private area.
- Control blocks describe all the datasets the job needs.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

How is a Job Executed?

- Initiator goes through three phases for each step of job:
  1. Allocation (required resources are allocated)
  2. Processing (region is created and program is loaded and executed)
  3. De-allocation (resources are released)
- This continues until there are no more job steps to process.

 15

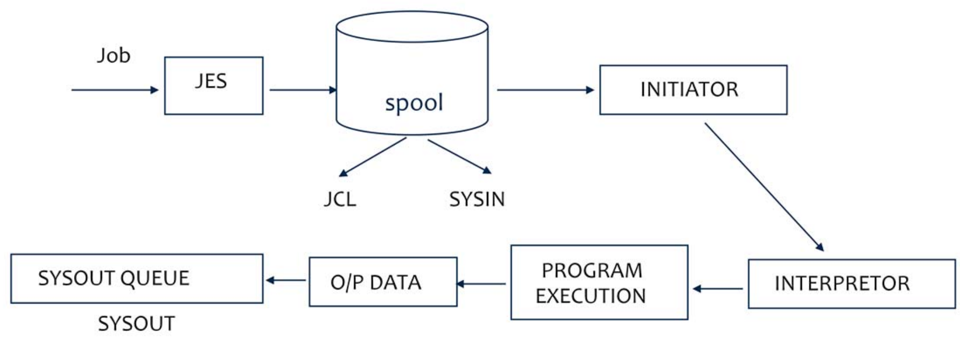
How is a Job Executed?

- Then, the initiator releases the job and searches the spool again for another job from the proper class to execute.
- As a user's program to execute, it can retrieve data that was included as part of job stream and stores in the JES spool.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 16

How is a Job Executed?...contd



<u>Job Execution</u>:

How Job Enters the System?

- Job can enter the system:
  1. From local or remote card readers (now obsolete).
  2. By starting a cataloged JCL procedure (for example: when user logs in, a predefined set of commands are executed as a batch job. These commands are stored as cataloged JCL procedure)
  3. By interactive users ‘thru’ SUBMIT command. Users can create a PDS member in which commands are specified. On issuing SUBMIT command, these are executed as a job.
- We will focus on the third approach.

<u>Job Execution (contd.)</u>:
Input:

- On **SUBMIT**, the **internal reader** reads the JCL and creates an **input stream**.
- **JES2** reads the **input stream**, assigns a **Job Number** and places input stream in SPOOL data set (a message is sent to TSO user about the job number).
- Job is put in the conversion queue.

Conversion:

- Converter program analyzes JCL statements.
- It then converts into converter / interpreter text.
- It then checks for Syntax errors.
    - If any error, Job is queued for output processing.
    - If no error, Job is queued for processing.

Processing:

- Selection is based on job class and priority.
- Selected job is passed to **Initiator**.
- Initiator invokes **Interpreter**.
- Interpreter builds **control blocks** from converter / interpreter text in a **Scheduler Work Area (SWA)**. SWA is part of address space's private area.
- Control blocks describe the data sets required by the job.
    - Initiator allocates resources required by the Job.
    - Initiator starts the program to be executed.
    - It builds the user region.
    - It loads the program in the user region.
    - It transfers control to the program.
    - On completion of the program execution, initiator de-allocates the resources.
- The process of allocation / execution and de-allocation is repeated for each job step.

Initiator Characteristics:

- Each initiator can handle one job at a time.
- There can be multiple initiators.
- Each initiator has a job class associated with it.
- System Operators can control the number of initiators and the class/es associated with each initiator.

Job Execution (contd.):

Input Data:

- Input data to the user's program can be specified in the job.
- It is called **in-stream data** or **SYSIN data**.
- SYSIN data is read and stored in JES spool.
- SYSIN data is treated like a data coming from card reader.

Output:

- It is concerned with management of System Messages, User Data Sets that need to be Printed / Punched.
- It is organized by output class and device set-up requirements.
- User's program can produce output data that is stored in a JES spool. It is called SYSOUT data.
- Each SYSOUT data is assigned an output class.
- Output class indicates the printer selection.
- "Held" output: Special class (usually Z) is assigned to "hold" the output.
- "Held" output remains in the SYSOUT indefinitely. It is usually used to verify before printing. User can change the class and thus release the "held" output.

Hard-Copy:

- It involves local or remote processing.
- It involves device selection.
- You can queue the output for print /punch.

Purge:

- It involves releasing SPOOL and Job Queue space.
- It further involves intimating TSO user about job completion.

Job Output:

- Output is produced at each stage of job processing.
- It includes output produced by:
  - ➢ JES
  - ➢ MVS
  - ➢ User's program where SYSOUT is allocated as output device.
- Job output is available to user (you can see it dynamically).
- It can be viewed using ISPF.

<u>Job Execution (contd.)</u>:

Components of Job Output:

Component 1- Separator Page:

- It is the first and last page of job output.
- It is Inserted by JES.
- It helps operator to the segregate the job outputs when directed to printer.

Component 2 – part I Job Log:

- It contains messages produced by JES.
- It is also displayed on operator’s console.
- If the job Abends, then error messages are logged in Job Log.

Component 2 – part II Job Statistics:

- It contains the summary information of system resources used by the job for example:
    - Number of JCL cards
    - Spool usage
    - Execution time

Component 3 - JCL Listing:

- It contains the list of JCL that was processed for the job.
- It should be same as what user has created.

Component 4 - Message Log:

- It includes messages regarding job execution.
- It includes messages produced by MVS.
- It includes details of:
    - Resource Allocation
    - Program Execution
    - Resource De-allocation
- It consists of Message label and message text. Message label starting with IEF indicates a MVS message.
- It contains installation specific messages.

Component 5 – SYSOUT:

- It contains separate sub-component for each SYSOUT allocation.
- Each SYSOUT can have different characteristics, for example: class, record length, and so on.

How is the Job's Output Processed?

- Like Jobs, SYSOUT data is assigned an output class that determines how the output will be handled.
- Common O/P classes are as follows:
  - A : Printer
  - B : Card Punch O/P
  - X : Held O/P

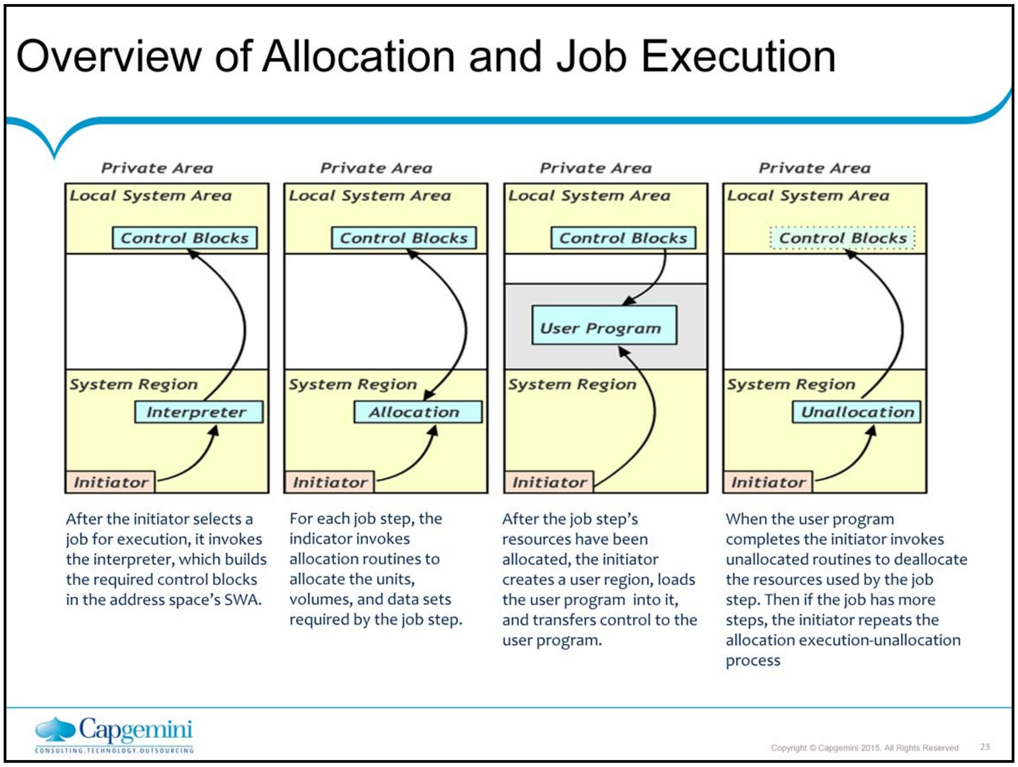
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Note: Held O/P stays on the sysout queue indefinitely. Usually, O/P is held so that it can be examined from a TSO terminal.

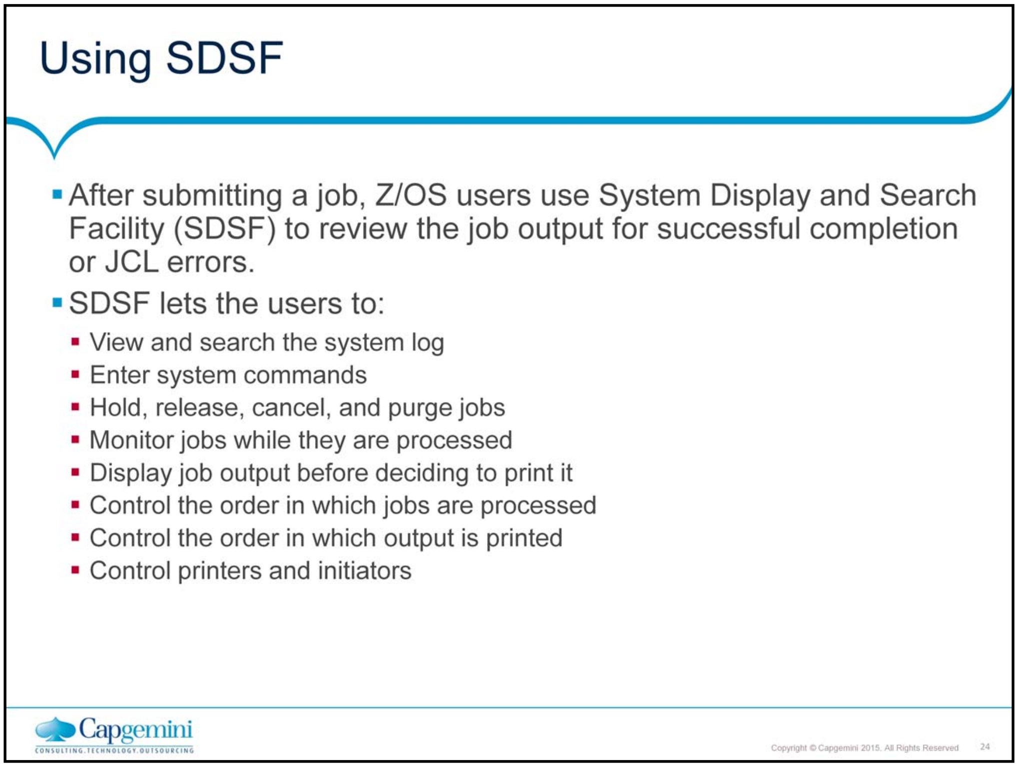
How is a Job Purged?

- After the job's output has been processed, the job is purged from the system.
  - That is to say, JES spool space, the job used, is freed so it can be used by other jobs, and any JES control blocks associated with the job are deleted.
- Once a job has been purged, JES no longer knows of its existence.

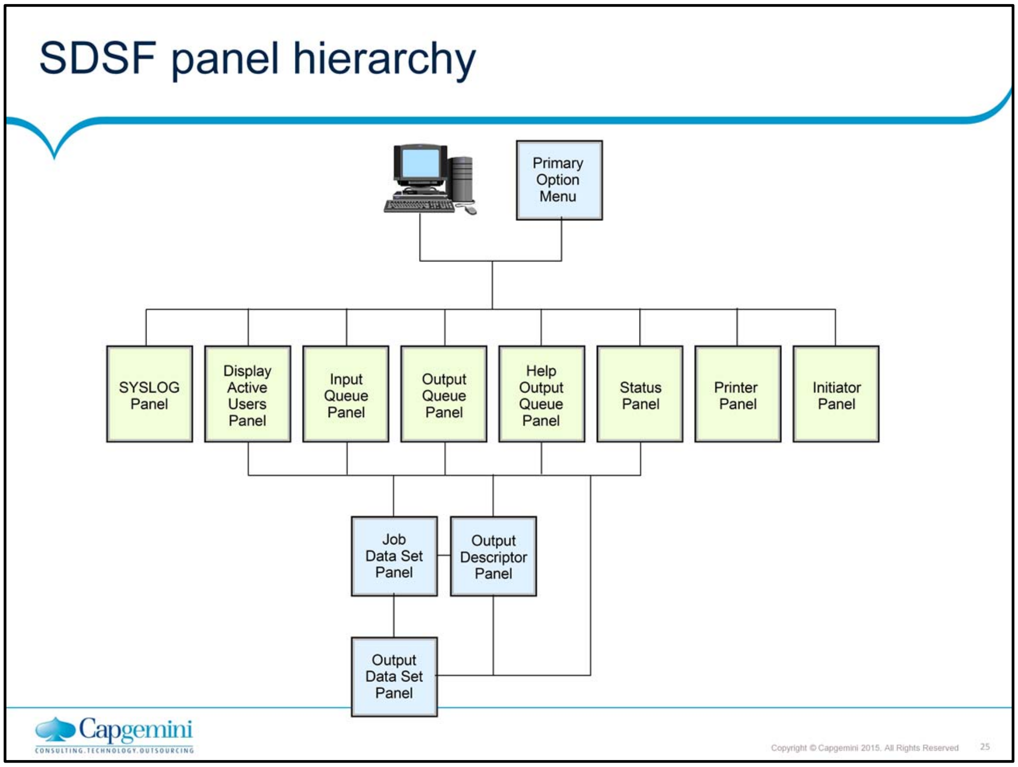
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

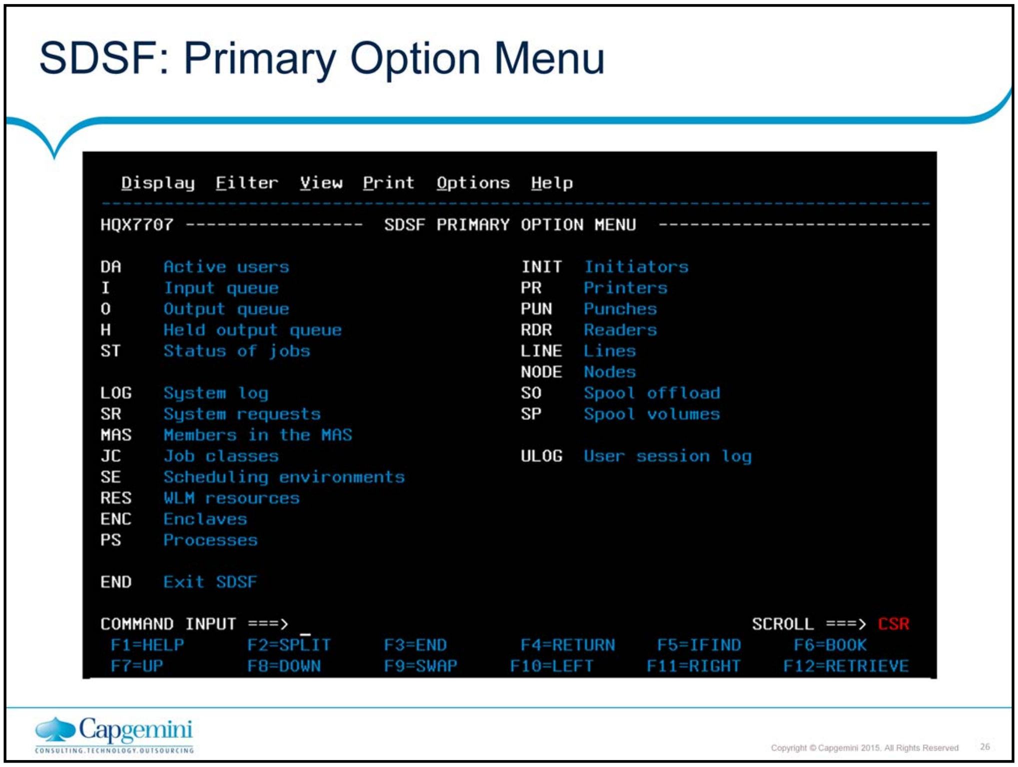


Note: When the user program completes, the initiator invokes **Unallocation** routine to deallocate the resources used by the job step.



SDSF is a software product whose primary purpose is to display printed output held in the JES spool area. Much of the printed output sent to JES by batch jobs (and other jobs) is never actually printed. Instead it is inspected using SDSF and deleted or used as needed.


SDSF panel hierarchy
Primary Option Menu
SYSLOG Panel
Display Active Users Panel
Input Queue Panel
Output Queue Panel
Help Output Queue Panel
Status Panel
Printer Panel
Initiator Panel
Job Data Set Panel
Output Descriptor Panel
Output Data Set Panel
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved
25



This is the SDSF primary options menu. Some of the options shown are:

DA The Display Active panel shows information about MVS address spaces (jobs, started tasks, and TSO users) that are running.

I The Input Queue panel displays information about jobs, started tasks, and TSO users on the JES2 input queue or executing.

O The Output Queue panel displays information about SYSOUT data sets for jobs, started tasks, and TSO users on any *nonheld* JES2 output queue.

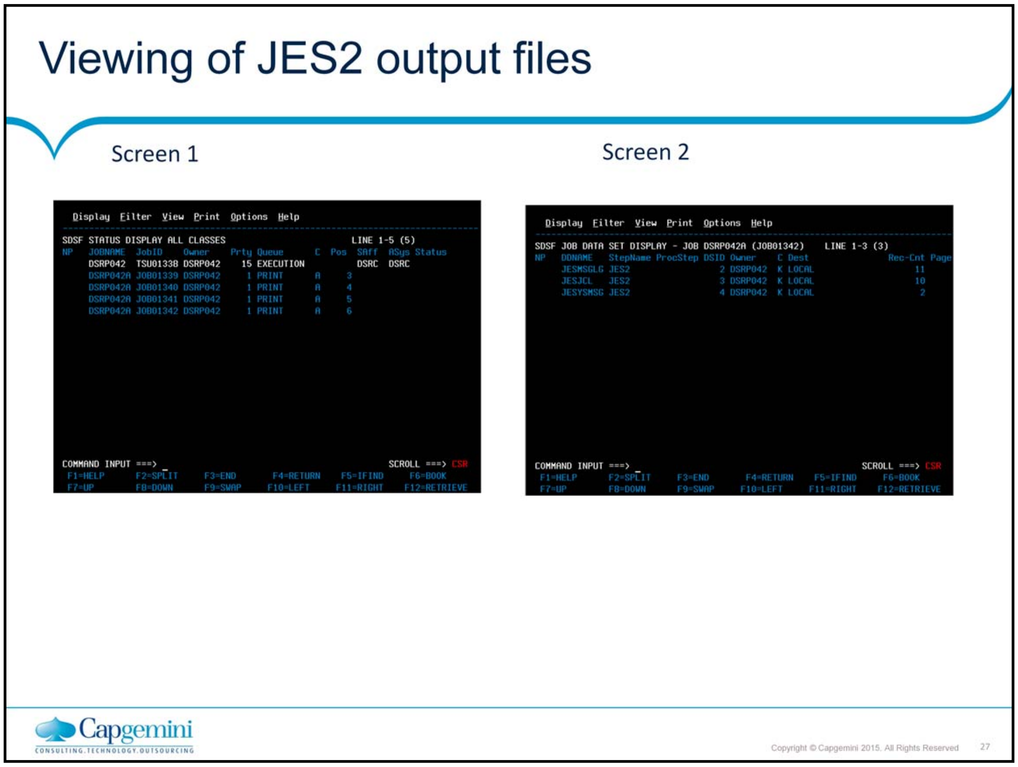
H The Held Output panel shows information about SYSOUT data sets for jobs, started tasks, and TSO users on any *held* JES2 output queue.

ST The Status panel displays information about jobs, started tasks, and TSO users on the JES2 queues.

LOG The system Log panel displays the log and lets you search it.

PS The Processes panel displays information about z/OS UNIX System Services processes.

PR The Printers panel displays information about JES2 printers printing jobs, started task, and TSO user output.



The first screen shown on the slide displays a list of the jobs we submitted and whose status is directed to the spool region. In this case there are four job has been submitted and executed.

Issuing a ? command in the NP column displays the output files generated by job 1342.

The second screen displays three ddnames: the JES2 messages log file, the JES2 JCL file, and the JES2 system messages file.

This option is useful when you are seeing jobs with many files directed to SYSOUT and you want to display one associated with a specific step. You issue an S in the NP column to select a file you want.

To see all files, instead of a ?, type S in the NP column;

SDSF: Display active users (DA)

```
SDSF DA DSRC   DSRC     PAG    0 SIO     4 CPU    4/  4  LINE 1-17 (71)
NP   JOBNAME  StepName ProcStep JobID    Owner    C Pos DP Real Paging    SIO
     *MASTER*                   STC08264 +MASTER+   NS  FF  11T   0.00   0.00
     ALLOCAS  ALLOCAS                               NS  FF 1343   0.00   0.00
     ANTAS000 ANTAS000 IEFPROC                      NS  FB 3168   0.00   0.00
     ANTMAIN  ANTMAIN  IEFPROC                      NS  FF 8696   0.00   0.00
     BPXOINIT BPXOINIT BPXOINIT                     IN  FF  166   0.00   0.00
     CATALOG  CATALOG  IEFPROC                      NS  FF 1917   0.00   0.00
     CICS1    CICS1    CICS     STC05508 CIC1USR    NS  FB 9210   0.00   0.00
     CICS2    CICS2    CICS     STC05511 CIC2USR    NS  FB 9888   0.00   0.00
     CICS3    CICS3    CICS     STC05514 CIC3USR    NS  FB 9156   0.00   0.00
     CICS4    CICS4    CICS     STC00768 CIC4USR    NS  FB  10T   0.00   0.00
     CICS5    CICS5    CICS     STC05518 CIC5USR    NS  FB 9760   0.00   0.00
     CICS6    CICS6    CICS     STC01013 CIC6USR    NS  FB 9625   0.00   0.00
     CICS7    CICS7    CICS     STC05524 CIC7USR    NS  FB 8972   0.00   0.00
     CNMPROC  CNMPROC  NETVIEW                      NS  FB  12T   0.00   0.00
     CNMPSSI  CNMPSSI  NETVIEW                      NS  FB  190   0.00   0.00
     CONSOLE  CONSOLE                               NS  FF 1003   0.00   0.00
     CSQ1CHIN CSQ1CHIN PROCSTEP STC05497 RACFSTC    IN  FB 3333   0.00   0.00
COMMAND INPUT ===> owner *                                 SCROLL ===> CSR
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Demo

- Demo on:
  - COBOL program execution and JCL program
  - Output of COBOL program

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 29

Summary

- In this lesson, you have learnt:
  - Job is an execution of one or more related programs in sequence.
  - JCL is a file containing control statements that provide the specifications necessary to process a job.
  - Job goes through various stages, such as:
    - Preparation
    - Scheduling
    - Execution
    - Termination

Review Question

- Question 1: ___ selects job for execution.
  - Option 1: Initiator
  - Option 2: JES
  - Option 3: Scheduler
- Question 2: Job Output are always held.
  - True/ False
- Question 3: Job Priority ranges from ___ to ___.

Multiple Virtual Storage (MVS)

Lab Book

Document Revision History

Date	Revision No.	Author	Summary of Changes
22-Jun-05	Version: 1.01		Added screen shots for options 3.12, 3.13, 3.14 and 3.15
26-Oct-09	Version 2.0D	Padmaja Purandare	Added extra Lab assignment
03-Dec-09	Version 2.0	CLS Team	Review
18-Oct-10	Version 3.0	Vaishali Kasture	Review and Extra Assignments in Appendix
30-June-11	Version 4.0	Rajita Dhumal	Added some lab exercises after Integration
8th-Aug-12	Version 4.1	Rajita Dhumal	Revamped after Assignment Review

Table of Contents

Getting Started

Overview

This lab book is a guided tour for learning Mainframe Environment and setup. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'To Do' assignments given.

Setup Checklist for MVS

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)
- Mainframe Connectivity using the Passport Client (Pc-to-Host software)

Please ensure that the following is done:

- CA-Relia software is already installed
- Passport PC-to-Host Terminal Software is installed to connect to MF Server

Instructions

- Note the Mainframe userid (for example: DSRP002 or DSRB002) and password for connecting to mainframe environment which would be given by the faculty. Remember the Mainframe ID to be used from now onward through all the MF courses henceforth.

- Create any PDS or PS with the following naming convention
  <<Userid>>.<<yourname>>.<<pdsname>>
- For Creating PS
  <<Userid>>.<<yourname>>.<<psname>>

- All lab exercise will be kept in a proper PDS.

- Debug all the COBOL programs offline without using MF connectivity (Dry run using Notepad and/or word documents for Pseudo Code and/or Code generation) prior to uploading in mainframe environment and check all programs are error and warning free.

Learning More (Bibliography if applicable)

- MVS JCL by Doug Lowe

- OS/VS2 MVS OVERVIEW by IBM
- THE MVS PRIMER by David Shelby Kirk
- EXPERT MVS/XA JCL by Caranthsis Mani
- MVS JCL, 2/ED. By Doug Lowe

Lab 1. Mainframe Environment

Goals	• Logging in the mainframe environment. • Understand the TSO and the different ISPF menus
Time	15 Minutes

1.1: Logging in the mainframe environment.

Login to Mainframe Environment using Passport Client Software.

Solution:

Step 1: From the Start menu navigate to Programs → Passport Client → Passport PC-to-Host.

Step 2: Key in the **IP host name** (Get it from the faculty). Once IP is entered, the IBM mainframe will be connected.

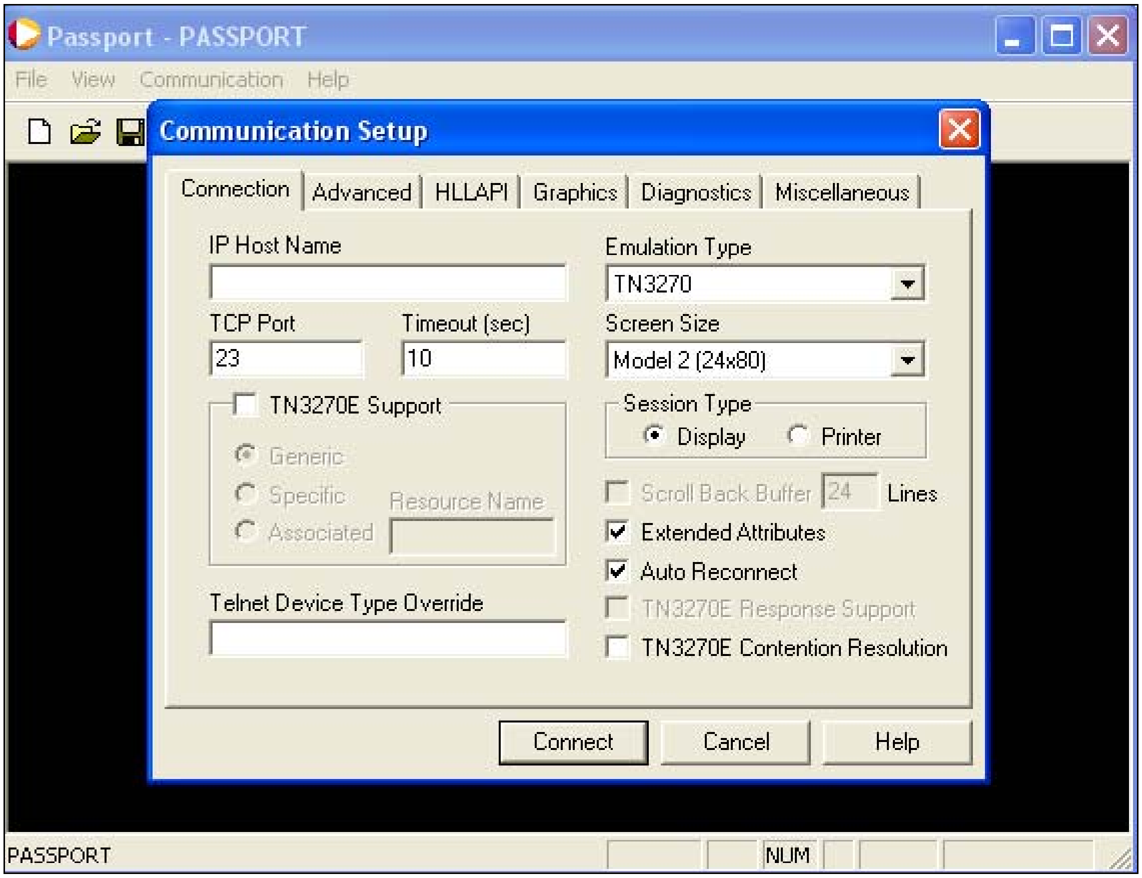


Figure 1: PC-to Host software

Step 3: Key in the TSO option. Key in the "**TSO"** command**.**

Step 4: Key in the Mainframe user id and password to logon to mainframe environment.

Note: Change the password at the very first logon and inform the faculty about the newly changed password.

Step 5: Key in the command "**P**" to enter the **ISPF menu**, which allows you to key in various commands.

1.2 Understand the TSO and the different ISPF menus

To Do: Go through the Primary Options Menu under ISPF

To Do: Use option 1 for Display source data or output listings (Refer to Appendix for various options under BROWSE)

To Do: Use Option 3 for understanding various utility functions

To Do: Exiting ISPF

Step: To terminate ISPF you can type =x at the command line or use the PF3 key to exit

Lab 2. Creation of PDS (Partitioned Dataset)

Goals	• PDS Creation
Time	15 Minutes

2.1: Creation of COBOL PDS

Problem: Need to create PDS to save all COBOL programs.

Solution:

Step 1: Key in the command **3.2 (ISPF/PDF PRIMARY OPTION MENU Screen)**, which will take you to utilities for allocation (**UTILITY SELECTION MENU Screen)**

Step 2: Key in the option as **A (allocate) on DATA SET UTILITY Screen**, type your name as “group”, and give an appropriate name for the PDS.

Example for the Name of PDS: DSRP001.SCOTT.COBOL

Step 3: Key in the option for dataset allocations on **ALLOCATE NEW DATA SET Screen**. The important ones are space units, primary quantity of units, secondary quantity of units, directory blocks (for storing member information in PDS), record format, record length, block size, and mainly the dataset name type (PDS).

Step 4: Press **ENTER** key **on DATA SET UTILITY Screen** to view the details of the dataset that is newly allocated.

2.2: Creation of JCL PDS

TO DO

Problem: Need to create PDS to save all JCL programs.

Hint: PDS is created is same as COBOL PDS

2.3: Creation of LOADLIB PDS

TO DO

Problem: Create a PDS for storing Load Module of the COBOL programs.

Hint: PDS is created is same as COBOL PDS. Change the parameter Record Format to ‘U’ as RECFM = U while creating LOADLIB PDS

To Do

Problem: Create a VSAM PDS same as JCL PDS.

Problem: Create a CICS PDS and DB2 PDS

Problem: Delete PDS
Step: The **D (delete)** option can be used to delete the PDS.

Lab 3. Creation of PSDS (Physical Sequential Data Set)

Goals	Create a flat file EMPLOYEE with the following structure
Time	15 minutes

3.1: Creation of PSDS

Problem: Create one flat file called EMPLOYEE with the following structure:

EMPNO	**C**	**3**
EMPNAME	**C**	**10**
EMPDEPT	**C**	**3**
EMPDESIG	**C**	**4**
EMPSAL	**N**	**5**

Solution:

Step 1: Key in the command **3.2** which will take you to utilities for allocation (**Primary Option Menu screen)**

Step 2: In the **Data Set Utility screen**, key in the option as **A (allocate)**, type your file in single quotes at **Data Set Name ….** (Here the name of the Data Set is **employee** under your own COBOL PDS)

Step 3: Key in the appropriate parameters **on Allocate New Data Set screen.** The difference in PDS and PSDS parameters is that for PSDS, **Directory Blocks** is Zero and **Data Set name type** should be blank for a PSDS.

Step 4: To see information about your Data Set allocation, the option should be kept blank. Press **ENTER** key **on DATA SET UTILITY Screen** to view the details of the dataset that is newly allocated

Step 5: Once the dataset is allocated, the records can be entered. Select option **2** from **ISPF (Edit)**. Type the dataset name in the **Edit Entry Panel**.

Step 6: Enter 10 records using the editor. Leave appropriate spaces for fields with less data.

Step 7: Type **Save** for saving the file or **Cancel** to discard changes. Press **F3** (function key) to save and exit.

3.2: Creation of Employee PSDS

Create a PSDS for storing Employee record using some other alternative method.

Hint: Check Option 2 in ISPF menu

Lab 4. View the Newly created PDS/PSDS, Browse, Edit

Goals	To view your PDS and file in the dataset list.
Time	10 minutes

Solution:

Step 1: Type option **3** in ISPF.

Step 2: Select **4** option for data set listing.

Step 3: Type the **login id** followed by ***** to see all the datasets or login id along with specific dataset.

Step 4: For any operation to be done on the Dataset, type **/ (slash)**.

Step 5: The following screen appears after the **/** option is selected. You can operate further on the file with the various options available. Select option **1** for editing the file. This will take you to the **Edit Entry Panel**.

Lab 5. Move, Copy, Rename members of PDS

Goals	Move / Copy, rename the members from one PDS to another.
Time	15 minutes

5.1: Move / Copy, rename the members from one PDS to another

TO DO

Problem: Move newly created member EMPLOYEE from an existing PDS to some other PDS

Step 1: Key in option **M** for moving on MOVE/COPY Utility Screen. Give the source data set name with details (Group, Type, and Member).

TO DO

Problem: Copy newly created member EMPLOYEE from an existing PDS to some other PDS

Step 1: We can copy by selecting the option **C (Copy)**.

Lab 6. Using Data Shift Commands

Goals	Upload the COBOL program using the utility available in ISPF in mainframe environment and indent as required.
Time	15 minutes

6.1: Using Shift Commands, indenting the programs.

Solution:

Step 1: Upload the COBOL Program using the **ISPF menu 6**.

Step 2: Store the Cobol Program in COBOL PDS by specifying appropriate path.

Step 3: Use **Edit** option to change the COBOL program.

Step 4: Use data shift commands to indent the COBOL program. (As a part of lab practices, do code alignment offline itself in notepad/text editor)

Hint: Refer **Appendix** to use the Data Shift commands

Lab 7. Editor Commands

Goals	Study the various editor commands.
Time	20 minutes

7.1: Studying the various Editor commands

Solution:

Step 1: Select option **2** from **ISPF** to go to the **Edit Entry Panel**.

Step 2: Select the file name to be edited, and give the right project, group.

Step 3: Type **COLS** in the command area to see the column numbers. This will be helpful while typing COBOL program.

Step 4: Use **d** to delete a line.

Step 5: To delete a set of lines, mark a block of lines with **DD** at the start and end.

Step 6: Select a line for copying by giving a **C** command at the source. At the destination, the command **B** or **A** can be used to copy before / after the current line.

Step 7: Similarly blocks for **copy** and **move** can be marked with **CC** and **MM**, respectively, and for copying before and after the command **B** or **A** can be used.

Step 8: You can use the **I10** command to insert 10 lines, **d10** to delete 10 lines. (Refer Appendix B).

Step 9: Finally, after all the changes are done, you can give the command to save.

Step 10: For changing a string, use the **C** Source-String Destination-String **ALL** command.

Step 11: Use the command **copy <filename>** to copy contents of a file to the current file.

Lab 8. Split Session

Goals	Split up the sessions and work in second session.
Time	10 minutes

8.1: Splitting a Session and working in the second session

Solution:

Step 1: Use **F2** key to split the session. Use **F9** key to switch between the sessions.

The following figure shows the look and feel of a split screen.

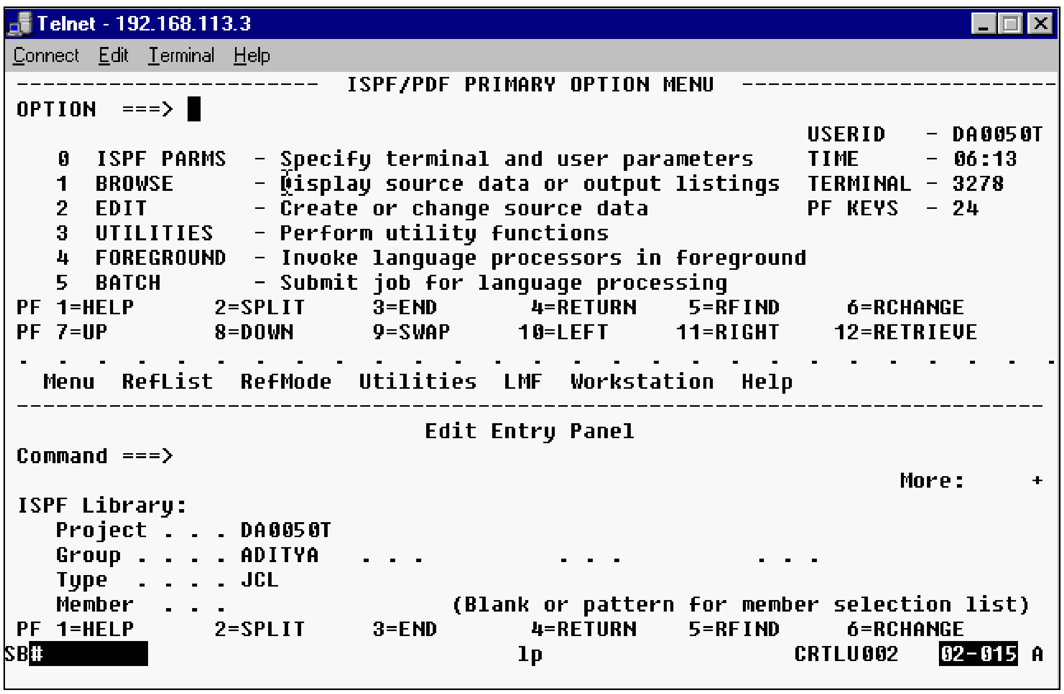


Figure 1: A split session

Lab 9. XMIT Command

Goals	Transfer (XMIT) the given member JCL to another user.
Time	15 minutes

9.1: Using the XMIT Command

Solution:

Step 1: Select Option **6** (command) in ISPF.

Step 2: Type the command as shown in the following figure. The **da0053t** is the destination user who will be transmitted the **aditya.jcl** PDS.

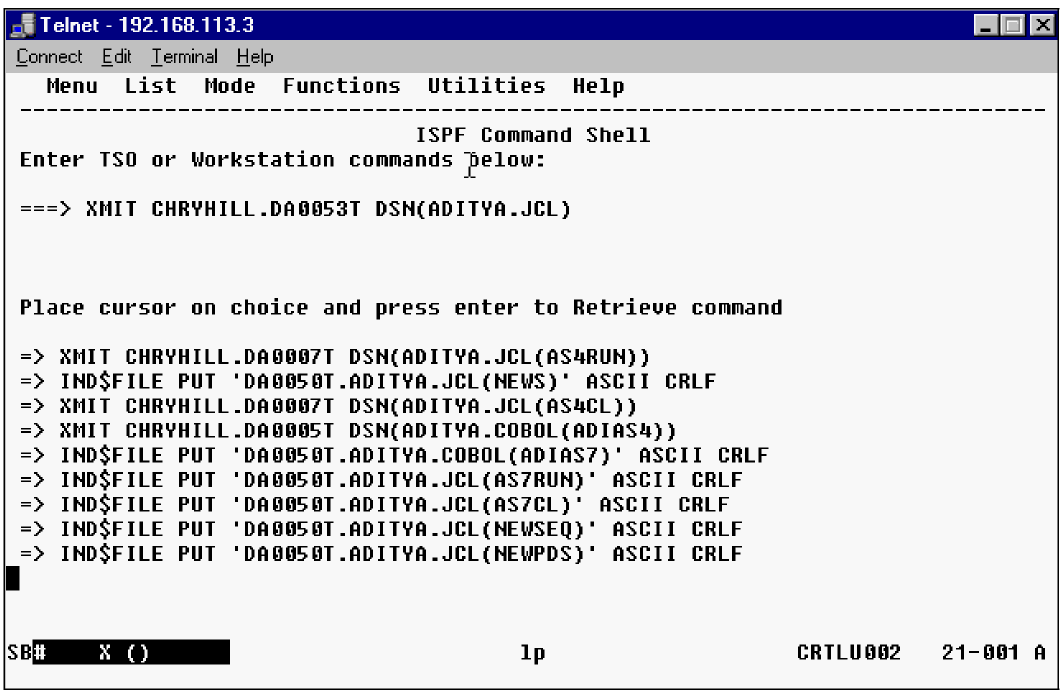


Figure 2: Transmitting aditya.jcl PDS

Step 3: The **da0053t** will have to receive the PDS with the **RECEIVE** command.

Lab 10. Comparison of two datasets (Standard)

Goals	To compare two datasets
Time	15 minutes

10.1: Comparing two standard datasets

Solution:

Step 1: From the **ISPF/PDF Primary Option Menu**, select option **3** to go to the **Utility selection Panel**.
Step 2: In the **Utility Selection Panel**, select option **12** to invoke the **SuperC** compare utility.
Step 3: Specify the name of the 'new' dataset that you want to compare. An '*****' mark against a **Member** option indicates that you want to compare all the datasets. Press **Enter** key to continue.

Step 4: Specify the name of the 'old' dataset, which you want to compare with previously selected 'new' datasets. Press **Enter** key to continue.

Step 5: As the comparison process begins, a message " ***** SuperC LINE Compare invoked ***** " is displayed on the screen.

Step 6: The matches and the differences are displayed under the **SUPERC.LIST** view. Here, in the statistics, the letter '**I**' indicates **Insert**, that is added to the new data set, and does not appear in old data set. The letter '**D**' indicates **Delete**, that is it is visible in the old data set but is absent (deleted) from the new data set.

Step 7: Press **F11** key to see the matches / differences details. These details list the total number of matches and also suggest the deletions and insertions to be made to make the datasets identical.

Step 8: Press **F8** key to go forward to see the statistics.

Step 9: Press **F9** key to move down the screen to see the non-paired files, which cannot be compared since there are no matching files in the new data set.

Step 10: This screen provides the options that are used by this utility for listing type, columns to be compared, the longest line, and processing.

10.2: Comparison of two datasets (Extended)

Step 1: In the **Utility Selection Panel**, select option **13** to invoke the **SuperCE Utility** for dataset comparison utility.

Step 2: Key in the old and new dataset names that have to be compared, and mention the types of comparison to be made. There are four ways to compare the datasets, namely file, line, word, and byte.

Step 3: Press **Enter** key to continue. A message saying " ***** SuperC FILE compare invoked. ***** " will be flashed on the screen.

Step 4: And then you get the statistics.

Step 5: Press **F9** key to continue.

Step 6: For line-by-line comparisons, select the compare type as **2** while setting the parameters for the **SuperCE utility**.

Step 7: Invoking the **3.13** utility with compare option **2** is similar to using the **3.12** utility. **3.12** option.

Step 8: Get the statistics, and press **F9** key to continue.

Step 9: The following screen gives the overall statistics for Line based comparison of data sets.

Step 10: Press **F11** key to see the hidden statistics on the right.

Step 11: To perform the word based comparison, select option **3** as the compare type.

Step 12: Press **F9** key to get the word-wise details.

Step 13: Select option **4** as the **Compare Type** to perform byte–wise comparison of datasets.

Step 14: The byte-wise statistics will be displayed.

Lab 11. Search-For Utility (Standard)

Goals	To understand the use of Search-For Utility
Time	20 minutes

11.1: Using the Search-For Utility

Solution:

Step 1: Select option **14** to invoke the **Search-For utility**.

Step 2: Key in the Search String, for example: EMPFILE, and the names of the datasets in which the string needs to be searched.

Step 3: Press **Enter** key to search. A message saying " ***** Search in progress ***** " is flashed on the screen.

Step 4: Press **F8** key to continue and to see the statistics.

11.2: Search-for utility (Extended)

To Do

Step 1: In the **Utility Selection Panel**, select option **15** (**Search-ForE**) to invoke the **Extended Search-For** utility.

Step 2: Key in the strings to be searched. Here, you can search for multiple strings.

Lab 12. Working with File Manager

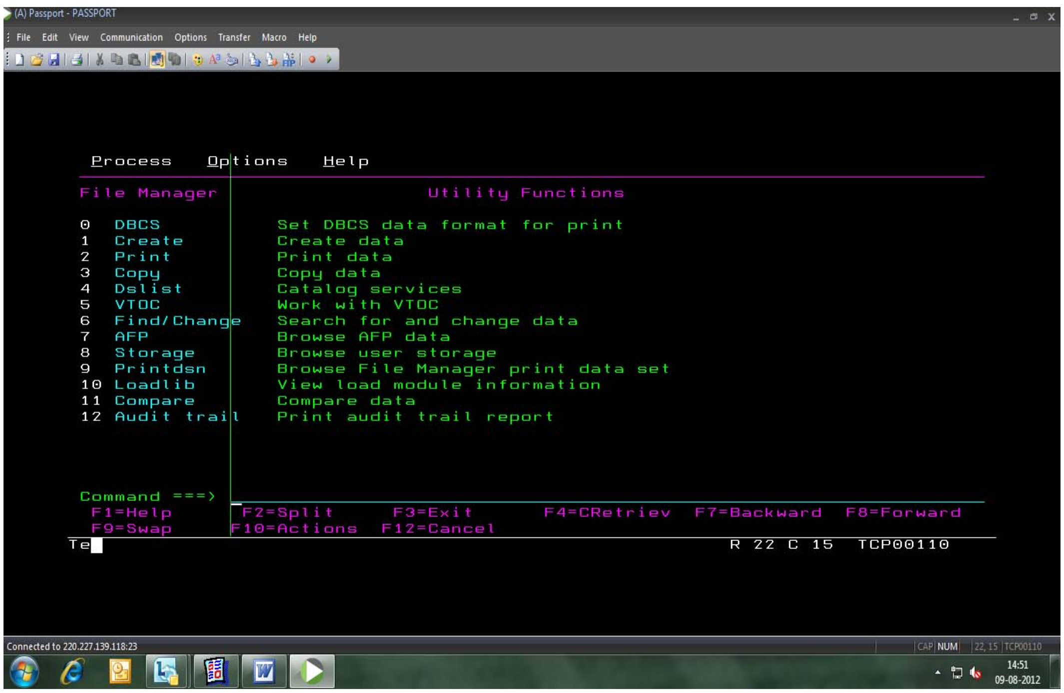
Goals	To understand the use of File Manager Utility
Time	60 minutes

11.1: Using the File Manager Utility

To Do:

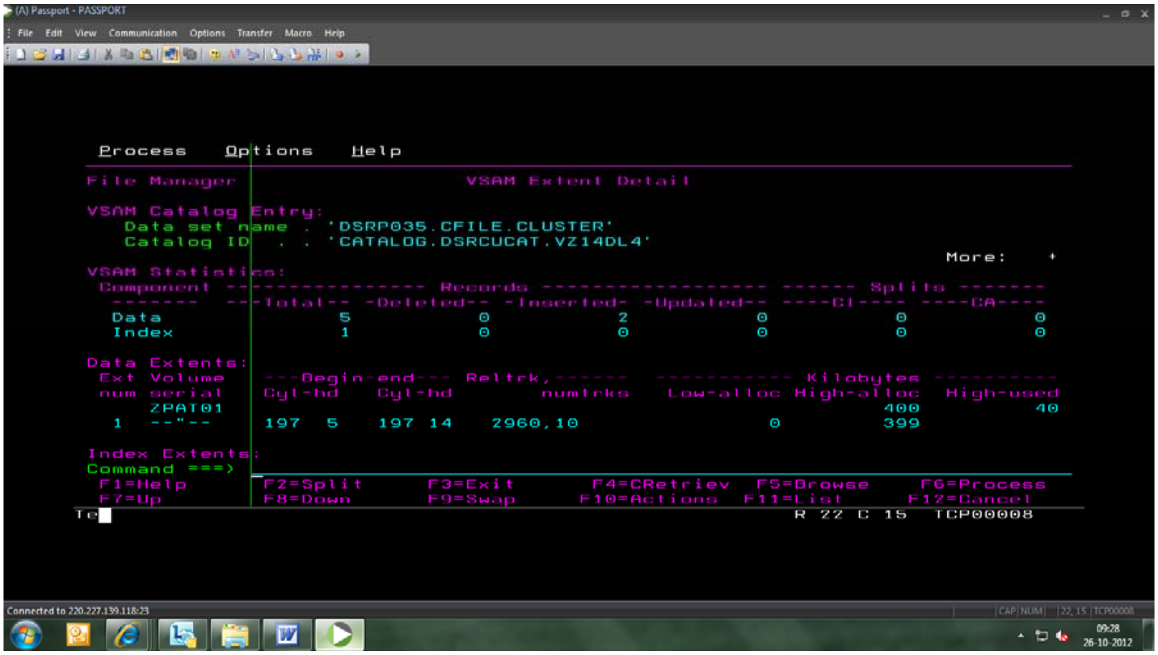
Do perform various editing operations across existing datasets such as find, copy, edit, print, compare and display data using File Manager utility.

(HINT: Use option 16 in ISPF Primary Options)



To Do:

Get the record count in VSAM file using file manager as shown in below screenshot.



(Hint: From the "UTILITY FUNCTIONS" screen choose option 4 (Dslist). On the "DATA SET LIST" screen put an "I" against the vsam file and enter. On the "VSAM ENTRY DETAIL" screen, press F11)

Appendix

Appendix – A: Compiling, linking, and executing a COBOL program

Solution:

Step 1: Key in the COBOL program using the **Edit Entry Panel**.

Step 2: We need to compile, link, and execute the COBOL program. Use the **Edit Entry Panel** to edit the **complink** JCL.

Step 3: Give the **source file name** and the **loadlib name** along with details.

Step 4: If there are no errors, then we can submit the JCL. Use the **SUBMIT** command.

Step 5: After submitting the JCL, a unique job id will be returned to us. This can be further used for identifying this job.

Step 6: If compilation is successful, then the return code will be Zero.

Step 7: Open the **Run JCL**, provide the **loadlib name** with details of the PDS, and submit the job.

Step 8: After submitting the JCL job, a unique job id will be given for the Job. The return code should be zero for successful execution of the program. View the spooler to view the output.

Step 9: Use the command Start ST;SD **on the command prompt to view the details.**

Step 10: Select the right job in the spooler area.

Step 11: Further details will be displayed. Select the SYSOUT RUN to view output.

Appendix - B: Compilation and Run JCL

```
EDIT       DA0001T.PRASANNA.JCL(COMPLINK) - 01.99        Columns 00001 00072
 Command ===>                                                    Scroll ===> CSR
 ****** ***************************** Top of Data ******************************
 000100 //DA0001TC  JOB  LA2719,'PRASANNA',NOTIFY=DA0001T,
 000110 //                MSGCLASS=X,TIME=(0,1)
 000112 //***********************************************************
 000120 //* STEP TO COMPILE A PROGRAM
 000130 //* COMPILER PROGRAM NAME - IKFCBL00
 000140 //* LIBRARY NAME - SYS1.COBCOMP
 000150 //* SYSLIN - OUTPUT FILE NAME
```



```
//* SYSIN  - INPUT FILE NAME (I.E. COBOL PROGRAM NAME)
//* SYSUT1,2,3, -  TEMPORARY FILES REQUIRED BY COBOL COMPILER
//************************************************************
//COB          EXEC PGM=IKFCBL00,REGION=1024K,
//
PARM='NOTRUNC,NODYNAM,LIB,SIZE=4096K,BUF=116K,APOST,NORES'
//SYSLIB      DD  DSN=SYS1.COBCOMP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSLIN      DD DSN=&&TEMP,DISP=(NEW,PASS),
//                     UNIT=SYSALLDA,SPACE=(TRK,(40,40))
//SYSUT1     DD  UNIT=SYSALLDA,SPACE=(TRK,(6,1))
//SYSUT2     DD  UNIT=SYSALLDA,SPACE=(CYL,(6,1))
//SYSUT3     DD  UNIT=SYSALLDA,SPACE=(CYL,(6,1))
//SYSUT4     DD  UNIT=SYSALLDA,SPACE=(CYL,(6,1))
//SYSIN       DD  DSN=DA0001T.PRASANNA.COBOL(PRG1),DISP=SHR
//*****************************************************
//* STEP TO LINK THE COBOL PROGRAM
//* LINKER PROGRAM NAME - IEWL
//* LIBRARY NAME - SYS1.COBLIB
//* SYSLMOD - OUTPUT DATASET NAME
//* SYSLIN - INPUT DATASET NAME
//*****************************************************
//LKED          EXEC  PGM=IEWL,PARM='LIST,XREF,LET,MAP',
//                      REGION=4096K,COND=(0,LT,COB)
//SYSLIN        DD  DSN=&&TEMP,DISP=(OLD,DELETE)
//SYSLIB        DD  DSN=SYS1.COBLIB,DISP=SHR
//SYSLMOD    DD  DSN=DA0001T.PRASANNA.LOADLIB(PRG1),
//                       DISP=SHR,UNIT=SYSALLDA
//SYSUT1       DD  UNIT=SYSALLDA,SPACE=(1024,(50,20))
//SYSPRINT   DD SYSOUT=*
//
****** **************************** Bottom of Data **************************
```



```
EDIT       DA0001T.PRASANNA.JCL(RUN) - 01.23            Columns 00001 00072
 Command ===>                                                    Scroll ===>
CSR
 ****** ****************************** Top of Data ******************************
 000100 //DA0001TR JOB LA2719,'PRASANNA',NOTIFY=DA0001T,
 000200 //        MSGCLASS=X,TIME=(0,1)
 000300 //*************************************
 000400 //*STEP TO RUN COMPILED COBOL PROGRAM
 000500 //COBRUN      EXEC PGM=PRG1
 000600 //STEPLIB      DD DSN=DA0001T.PRASANNA.LOADLIB,DISP=SHR
 000700 //SYSPRINT     DD SYSOUT=*
 000810 //INF1         DD DSN=DA0001T.EMPDATA,DISP=SHR
 000900 //OTF1         DD DSN=DA0001T.L3,DISP=(NEW,CATLG,DELETE),
 001000 //                UNIT=SYSDA,SPACE=(TRK,(1,1)),
 001100 //                DCB=(LRECL=80,RECFM=FB,BLKSIZE=800,DSORG=PS)
 001200 //SYSOUT       DD SYSOUT=*
 ****** ***************************** Bottom of Data *************************
```


Appendix - C: ISPF (Interactive System Productivity Facility)

Primary Options Menu:

Access to ISPF is gained by keying in ISPF at the READY prompt. This is done as default in the auto executed CLIST at startup. When this command is keyed in, you get the Primary Options Menu.

```
----------------------- ISPF/PDF PRIMARY OPTION MENU ------------------------
 OPTION  ===> pfshow                                         USERID   - DA0034T

   0  ISPF PARMS      - Specify terminal and user parameters         TIME     - 06:58
   1  BROWSE          - Display source data or output listings       TERMINAL -
3278
   2  EDIT            - Create or change source data                 PF KEYS  - 12
   3  UTILITIES       - Perform utility functions
   4  FOREGROUND      - Invoke language processors in foreground
   5  BATCH                 - Submit job for language processing
   6  COMMAND         - Enter TSO Command, CLIST, or REXX exec
   7  DIALOG TEST     - Perform dialog testing
   8  LM UTILITIES    - Perform library administrator utility functions
   9  IBM PRODUCTS    - Additional IBM program development products
  10  SCLM                  - Software Configuration and Library Manager
   C  CHANGES         - Display summary of changes for this release
   T  TUTORIAL        - Display information about ISPF/PDF
   X  EXIT            - Terminate ISPF using log and list defaults
   D  DATACENTER      - Perform Datacenter Defined Functions
   S  SDSF                  - Spool Display and Search Facility
   U  USER                  - Perform User Defined Functions

 F1=HELP    F2=SPLIT    F3=END    F4=RETURN   F5=RFIND   F6=RCHANGE
  •   F7=UP       F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT
     F12=RETRIEVE
```


Browsing Datasets (Option 1)

```
------------------------- BROWSE - ENTRY PANEL  ------------------------------
 COMMAND ===>

 ISPF LIBRARY:
   PROJECT ===> DA0034T
   GROUP    ===> TRG      ===>         ===>         ===>
   TYPE       ===> JCL
   MEMBER ===>                (Blank or pattern for member selection list)

 OTHER PARTITIONED OR SEQUENTIAL DATA SET:
   DATA SET NAME  ===>
   VOLUME SERIAL  ===>        (If not cataloged)

 DATA SET PASSWORD ===>        (If password protected)
 MIXED MODE       ===> NO      (Specify YES or NO)
 FORMAT NAME      ===>
```

Browse Commands

- Type “COLS” command for displaying columns.
- Scroll up, down, left right with PF7, PF8, PF10, and PF11 respectively.
- Set Scroll amount to CRSR, HALF, PAGE, n lines, max, DATA
- Scroll by n lines, to top, or bottom
- Define / LOCATE {line number}/label.
- FIND string {NEXT/PREV/FIRST/LAST/ALL}.
- PF5 for repeat find and use of “&”.
- Use of PF12 to recall last command.
- Terminate Browse with PF3 Key.
- FIND string {NEXT/PREV/FIRST/LAST/ALL} {CHAR/PREFIX/SUFFIX/WORD} col-1 col-2
- Column limitation search
- T ‘text’ - for case insensitive search
- X ‘hex-string’ for a hex search

Editing Datasets (Option 2)

- The Primary Editor entry is similar to that for Browse as regards concatenating datasets and dataset selection.
- Labels can be defined as in browse but may be entered as line commands.
- Error messages may be removed by typing RESET on the command line.

Standard editing commands

I/In	Insert 1 or n lines.
D(n)	Delete line or n lines.
DD	Delete the block marked by the 2 DD line commands.
R(n)	Repeat 1 or n lines.
RR	Repeat the block marked by the 2 RR line commands.
C(n)	Copy 1 or n lines.
CC	Copy the block marked between the 2 CC line commands.
M(n)	Move 1 or n lines.
MM	Move the block marked between the 2 CC line commands.
A(n)	Copy or Move lines 1 or n times after this line.
B(n)	Copy or Move lines 1 or n times before this line.

Creating datasets and exiting editor

- To create a new member, specify non-existent member name in the current PDS.
- You can quit the editor without saving changes by the CANCEL command.
- You can update the dataset with the SAVE command.
- You can exit with implicit save using the END command or PF3 key.

Edit Profiles

- Edit profiles control editing options.
- Normal editing of a new dataset uses the default profile - the dataset type.
- To display the edit profile, type PROFILE on the command line in the editor.
- To remove it from the screen, type RESET.
- This gives you a display as shown below:

```
EDIT ---- DA0034T.TRG.JCL(JCL1) - 01.27 --------------------- COLUMNS 001 072
 COMMAND ===>                                            SCROLL ===> CSR
 ****** ***************************** TOP OF DATA ******************************
 =PROF> ....STD (FIXED - 150)....RECOVERY OFF....NUMBER ON STD.................
 =PROF> ....CAPS ON....HEX OFF....NULLS ON STD....TABS ON STD....SETUNDO
OFF....
 =PROF> ....AUTOSAVE ON....AUTONUM OFF....AUTOLIST OFF....STATS ON.............
 =PROF> ....PROFILE UNLOCK....IMACRO NONE....PACK OFF....NOTE ON................
 =BNDS> <
 =TABS>
 =COLS> ----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
 000100 //DA0034TA JOB LA2719,'PARAG',NOTIFY=DA0034T,
 000200 //        CLASS=A,MSGCLASS=X
 000300 //*
 000400 //COBRUN EXEC PGM=PROG11
 000500 //STEPLIB DD DSN=DA0034T.TRG.LNK,DISP=SHR
 000510 //*STEPLIB DD DSN=DA0034T.TRG.COBOL2,DISP=SHR
 000600 //INVMAS DD DSN=DA0034T.TRG.INVMAS,DISP=SHR
 000700 //OP1 DD SYSOUT=*
 000710 //*OP1 DD DSN=DA0034T.TRG.EXE7,DISP=(NEW,CATLG,CATLG),
```

Profile Settings

- To switch to a different profile, key in “profile <“profile-name”>
- To lock a profile, at the command line key in “PROFILE LOCK”
- Any changes made to the locked profile are not saved permanently.
- Caps, number Pack, and STATS modes are set each time you begin an edit session.
- To define tab stops, key in TABS on the command line and place ‘@’ on the tabs line one character before where you would like a tab stop. On the command line, key in TABS ON/OFF <tab-character>.
- If you omit the tabbing character, hardware tabbing is assumed.
- Line control Commands:
  - Nonumber/NUM OFF turns off line numbering
  - NUM ON turns on line numbering
  - AUTONUM resequence line numbers on save

- RENUM resequence line numbers
- NUM ON COBOL checks for valid COBOL numbering
- NUM ON STD checks for standard line numbering
- UNNUM removes line numbering.

Edit Modes

- STATS ON/OFF : Controls dataset statistics
- AUTOLIST ON/OFF : Controls Automatic listing
- NULLS ON/OFF : Controls if nulls or spaces are padded
- RECOVERY ON/OFF : Recovers a dataset being edited in case of a system crash.
- UNDO command : This works up to the last save only
- HEX ON/OFF : Displays data in HEX/ASCII mode
- CAPS ON/OFF : Converts Lower case letters to uppercase if set to on Line. Commands for this function are LC or UC. LCLC and UCUC are blocked line commands.
- PACK ON/OFF : Specifies that the data is stored in compressed mode.
- AUTOSAVE ON/OFF PROMPT/NOPROMPT : Auto save data when PF3 key is pressed
- IMACRO : Specify initial macro to be run at startup.

Advanced Edit Options

To locate a String within another use the following command:

FIND string range NEXT/PREV/FIRST/LAST/ALL CHARS/PREFIX/SUFFIX/WORD X/NX col-1 col-2

where:

Range	: is denoted by 2 labels
String	: is the string to be found
NEXT	: indicates start search at current line and locate the next occurrence of the string (default).
PREV	: indicates start search at current line and locate the previous occurrence of the string.
FIRST	: indicates locate the first occurrence of the string
LAST	: indicates locate the last occurrence of the string
ALL	: indicates same as first but count the occurrences in the file.
CHARS	: indicates any occurrence of the string
PREFIX	: indicates string must be at the beginning of the word
SUFFIX	: indicates string must be at the end of a word
X/NX	: indicates search only excluded/Non excluded lines
col-1 col-2	: indicates starting and ending column numbers defining the search boundaries.

To Modify/Change a string with another String;

CHANGE string1 string2 range NEXT/PREV/FIRST/LAST/ALL
CHARS/PREFIX/SUFFIX/WORD X/NX col-1 col-2

String2 replaces string1

Shifting text source

<--------------Data shift---------->   <---------- Column shift ----------->

<	**<n**	**<<**	**(**	**(n**	**((n**	**left shifts**
>	**>n**	**>>**	**)**	**)n**	**))n**	**right shifts**

Data shifts

- Does not drop blank characters
- Does not combine words by drooping spaces
- Does not delete spaces within apostrophes
- COPY [member] [AFTER/BEFORE label]
- MOVE [member] [AFTER/BEFORE label]
- CREATE [member] [range]
- REPLACE [member] [range]
- Edit member-name to edit recursively

Library Utility

Option 3.1

```
--------------------------- LIBRARY UTILITY  --------------------------------
 OPTION  ===>

   blank - Display member list   B - Browse member
   C - Compress data set           P - Print member
   X - Print index listing         R - Rename member
   L - Print entire data set        D - Delete member
   I - Data set information         E - Edit member
                                    S - Data set information (short)
 ISPF LIBRARY:
   PROJECT ===> DA0034T
   GROUP   ===> TRG      ===>         ===>         ===>
   TYPE    ===> JCL
   MEMBER  ===>               (If   "P", "R", "D", "B", "E" or blank selected)
   NEWNAME ===>               (If  "R" selected)

 OTHER PARTITIONED OR SEQUENTIAL DATA SET:
   DATA SET NAME  ===>
   VOLUME SERIAL  ===>        (If not cataloged)

 DATA SET PASSWORD ===>        (If password protected)
```

DSLIST Commands

M	-	Member list
C	-	Catalog a dataset
D	-	Delete a dataset
E	-	Edit a dataset
F	-	Free unused dataspace in a dataset
I	-	Display information for a dataset
M	-	Display a memberlist
P	-	Print a dataset
R	-	Rename a dataset
S	-	Display a shortened version of dataset information
U	-	Uncatalog a dataset
X	-	Print a dataset indexed listing
Z	-	*Compress a dataset*
=	-	Repeat the last command

Primary Commands

LOCATE - To locate a dataset
TSO SUBMIT - To execute Clists from the command line
SHOWCMD ON/OFF - To show the expanded form of the command
CONFIRM ON/OFF - Same as Confirm delete request Yes/NO on the delete panel
SORT - Sorts the dataset list based on the fields shown on the next transparency
FIND - Finds occurrence of a string with the list of datasets
SAVE dataset-name - Saves the current dataset list into the dataset name specified
SELECT pattern [line command] - To make a selection of datasets to be acted upon determined by the line command

Reset

Option 3.5

```
-------------------------- RESET ISPF STATISTICS ----------------------------
 OPTION  ===>

   R - Reset (create/update) ISPF statistics
   D - Delete ISPF statistics

 NEW USERID                 ===>        (If userid is to be changed)
 NEW VERSION NUMBER         ===>        (If version number is to be changed)
   RESET MOD LEVEL                  ===> YES     (YES or NO)
   RESET SEQ NUMBERS        ===> YES     (YES or NO)

 ISPF LIBRARY:
   PROJECT ===> DA0034T
   GROUP   ===> TRG
   TYPE    ===> JCL
   MEMBER  ===>                  (Blank or pattern for member selection
                          list, '*' for all members)

 OTHER PARTITIONED DATA SET:
   DATA SET NAME  ===>
   VOLUME SERIAL  ===>           (If not cataloged)
```


Appendix - D: File Aid

File-AID is a data manipulation program developed by COMPUWARE that consolidates the functions of most standard IBM Utilities.

File-AID Utility has two possible modes of Operation:

- Online Mode (Using =F option within ISPF Menu)
- Batch Mode (JCL)

(**Note:** This presentation aims at unfolding the frequently used online capabilities of File-AID utility)

F.1 Browse

File-AID enables you to browse a file created through any standard MVS access method (including IAM files). You can display the entire dataset or a selected subset of records.

You can supply record layouts and view your data in three display modes:

i) Character

ii) Vertical formatted

iii) Formatted

Character Mode:

Step 1: Key in dataset name and record layout of the dataset that you want to browse.

```
File-AID --------------  Browse - Dataset Specification  -------------------------
COMMAND ===>

 Browse Mode                ===> C         (F=Formatted; C=Char; V=Vertical)

 Specify Browse Information:
   Browse dataset name           ===> “USERID9.FASAMP.EMPMAST”
   Member name                   ===>           (Blank or pattern for member list)
   Volume serial          ===>          (If dataset is not cataloged)


 Specify Record Layout and XREF Information:
   Record layout usage           ===> S         (S = Single; X = XREF; N = None)
   Record layout dataset         ===> FASAMP.LAYOUTS
   Member name                   ===> EMPLOYEE   (Blank or pattern for member list)
   XREF dataset name             ===>
   Member name                   ===>           (Blank or pattern for member list)

 Specify Selection Criteria Information:     (E = Existing; T = Temporary;
   Selection criteria usage      ===> N         M = Modify; Q = Quick; N = None)
   Selection dataset name        ===>
   Member name                   ===>           (Blank or pattern for member list)
```


F.1 Browse screen

```
File-AID - Browse - USERID9.FASAMP.EMPMAST -------------- LINE 0000 COL 1  8
 COMMAND ===>                                                    SCROLL ===> PAGE|
********************************** TOP OF DATA **********************CAPS OFF-*|
00090MARTIN       EDWARD    M  AIRPLANE MANUFACTURER    427890125 101954
- - - - - - - - - - - - - - - - - - - - - - - - - -  1 RECORD(S) NOT SELECTED|
00200JACKSON      JOSEPH     C   ORATOR                        275587177 020462
10000ANDREWS      GEORGE        ACTOR                          576312032 042248
15000MURPHY       RONALD    L   PAINTER                        987654321 120255
18034SCHNEIDER    ELLEN      C   NURSE                         341559549 032960
21035JONES        GEORGE   B   COUNTRY SINGER                 463813456 090944
25100ROBERTS      WILLIAM  R   POLITICIAN                     879563325 050865
27007ALLEN        JOYCE       M  AUTHOR                        783458334 012132
30001RICHARDS     REX         W  RODEO CLOWN                  632764534 040140
31000SAVAGE       JONATHON  C  ELECTRICIAN                    348567992 062250
34010SMITH        JANET            AIRLINE ATTENDANT          557782984 112359
34011JACOBS       DIANA            DOCTOR                      225368395 021757
******************************** BOTTOM OF DATA *******************-CAPS OFF-*
                                                  |
 Enter FMT for formatted mode, VFMT for vertical format, HEX ON  for He
```


Vertical Formatted Mode (VFMT) :

Step 2: Type **VFMT** at command prompt from Character Mode or Select Browse Mode as “V” at F.1 screen.

```
File-AID - Browse - USERID9.FASAMP.EMPMAST  -----------
 COMMAND ===> VFMT
----+----1----+----2----+----3----+----4----+----5----+-
********************************* TOP OF DATA **********
00090MARTIN        EDWARD    M  AIRPLANE MANUFACTURER
```

Step 3: This expands record as per the file layout columns

```
File-AID - Browse - USERID9.FASAMP.EMPMAST -------------- LINE 0000 COL 1  4|
 COMMAND ===>                                        SCROLL ===> PAGE|
EMP-NUMBER EMP-LAST-NAME   EMP-FIRST-NAME EMP-MID-INIT FILLER   EMP-
TITLE      |
 5/AN       15/AN          10/AN         1/AN        2/AN     30/AN
 (1-5)      (6-20)         (21-30)       (31-31)     (32-33)  (34-49)
 1--------- 2-------------- 3------------- 4----------- 5------- 6--------------
 ********************************* TOP OF DATA *********************-CAPS OFF-*
 00090      MARTIN         EDWARD        M                AIRPLANE MANUFA
 - - - - - - - - - - - - - - - - - - - - - - - - - - -  1 RECORD(S) NOT SELECTED
 00200      JACKSON        JOSEPH        C                ORATOR
 10000      ANDREWS        GEORGE                         ACTOR
 15000      MURPHY         RONALD        L                PAINTER
 18034      SCHNEIDER      ELLEN         C                NURSE
 21035      JONES          GEORGE        B                COUNTRY SINGER
```


Format Mode (FMT):

Step 4: Type **FMT** at command prompt from Character Mode or select Browse Mode as "F" at F.1 Screen.

```
File-AID - Browse - USERID9.FASAMP.EMPMAST  ------------------------ COL 1  92
 COMMAND ===>                                   SCROLL ===> PAGE
 RECORD:    1          EMPLOYEE-MASTER-FILE           LENGTH:   198  ---- FIELD LEVEL/NAME --
----- COLUMNS- ----+----1----+----2----+----3----+----4|
 5 EMP-NUMBER                    1    00090
 5 EMP-LAST-NAME                 6    MARTIN
 5 EMP-FIRST-NAME               21    EDWARD
 5 EMP-MID-INIT               31    M
 5 FILLER                   32
 5 EMP-TITLE                  34    AIRPLANE MANUFACTURER
 5 EMP-PERSONAL-INFO SYNC        64
  10 EMP-NATL-ID-NUMBER        64    427890125
  10 FILLER                  73
  10 EMP-DATE-OF-BIRTH         74    101954
  10 EMP-DOB-REDEF REDEFINES EMP-DATE-OF-BIRTH
  10 EMP-DOB-REDEF SYNC         74
   15 EMP-DOB-MM             74    10
   15 EMP-DOB-DD             76    19
   15 EMP-DOB-YY             78    54
  10 EMP-HIRE-DATE            80    920101
  10 EMP-MARITAL-STATUS        86    M
 5 EMP-WITHOLD-INFO SYNC         87
  10 EMP-LIFE-INS-WITHOLD-AMT    87    30000}
 Enter CHAR for character mode, VFMT for vertical format mode
```

Navigating To Browse Your Formatted Records

Each of the navigation commands has a corresponding **PF** key set as the default in your user profile.
The default settings are:

PF7	UP
PF8	DOWN
PF10	LEFT (BACK)
PF11	RIGHT (FORWARD)

You can specify a number of records to scroll forward. For example, if record number 10 is the currently displayed record.

F.2 Edit

Step 1: Select F.2 and key in the name of the dataset you want to edit. Press **Enter** key to go to next screen. Edit the data and type **SAVE** at command prompt to save the data.

```
File-AID --------------  Edit - Dataset Specification  ------------------------
COMMAND ===>

 Edit Mode                ===> C        (F=Formatted; C=Char; V=Vertical)

 Specify Edit Information:
   Edit dataset name      ===> “USERID9.FASAMP.EMPMAST”
   Member name            ===>          (Blank or pattern for member list)
   Volume serial          ===>          (If dataset is not cataloged)


 Specify Record Layout and XREF Information:
   Record layout usage      ===> N        (S = Single; X = XREF; N = None)
   Record layout dataset    ===>
  Member name             ===>          (Blank or pattern for member list)
   XREF dataset name        ===>
   Member name            ===>          (Blank or pattern for member list)

 Specify Selection Criteria Information:    (E = Existing; T = Temporary;
   Selection criteria usage ===> N          M = Modify; Q = Quick; N = None)
   Selection dataset name   ===>
   Member name            ===>          (Blank or pattern for member list)
```


F.3 Utilities

File Copy Utility:

Step 1: Key in **3** at the Option Prompt.

```
 File-AID 8.8.2 -------------  Primary Option Menu  ----------------------------
OPTION ===> 3

 0  PARAMETERS  - Specify ISPF and File-AID parameters      USERID   - USERID9
 1  BROWSE      - Display file contents                     PF KEYS  - 24
 2  EDIT        - Create or change file contents            TERMINAL - 3278
 3  UTILITIES   - File-AID/SPF extended utilities           TIME     - 01:10
 5  PRINT       - Print file contents                       JULIAN   - 05.194
 6  SELECTION   - Create or change selection criteria       DATE     - 05/07/13
 7  XREF        - Create or change record layout cross reference
 8  VIEW        - View interpreted record layout
 9  REFORMAT    - Convert file from one format to another
10  COMPARE     - Compare file contents
 C  CHANGES     - Display summary of File-AID changes
 T  TUTORIAL    - Display information about File-AID
 X  EXIT        - Terminate File-AID and return to ISPF

   Use END to terminate File-AID
```


Step 2: We will reach the below mentioned screen. Enter **3** for Copy option.

```
OPTION ===>  3

1  LIBRARY        - Display and modify directory entries; display load
                    module CSECT maps; browse, delete, rename PDS members
2  DATASET        - Display dataset information; allocate non-VSAM datasets
                    and GDGs; catalog, uncatalog, delete, or rename datasets
3  COPY           - Copy entire datasets; copy selected records; copy PDS
                    members based on name, statistics and/or content
4  CATALOG        - Display generic catalog entries or VSAM datasets on a
                    volume in list form and do dataset list processing
5  VSAM           - Allocate, display, delete, modify, or rename VSAM clusters,
                    alternate indexes, or paths; manage IAM files
6  SEARCH/UPDATE - FIND and CHANGE across PDS members.  Search for and/or
                    update data globally in any type of dataset.
7  VTOC           - Display and process datasets on a volume(s)

8  INTERACTIVE   - Execute File-AID/Batch

9  BATCH SUBMIT  - Build batch jobstreams

G  XMLGEN         - Generate an XML tagged document from data file
```


Step 3: Key in the dataset name to be copied, and the new dataset name. In case you wish to copy into an existing dataset, give the DISP as old.

```
 File-AID --------------------  Copy Utility  --------------------------------
COMMAND ===>

Specify "FROM" Dataset or HFS Path Information:
 Dataset or path ===> 'USERID9.FASAMP.EMPMAST'
 Volume serial   ===>                    (If not cataloged)

Specify "TO" Dataset or HFS Path Information:
 Dataset or path ===> 'USERID9.FASAMP.EMPMAST.NEW'
 Volume serial   ===>                    (If not cataloged)
 Disposition     ===> NEW                 (OLD, MOD, NEW)


Specify Execution Information:
 Process online or batch   ===> O       (O = Online; B = Batch)

Specify Selection Criteria Information:   (E = Existing; T = Temporary;
 Selection criteria usage  ===> T        M = Modify; Q = Quick; N = None)
 Selection dataset name    ===>
 Member name               ===>         (Blank or pattern for member list)
```


Step 4: The new dataset will have the same parameters as the original one. We can change the same over here. Press **Enter** key to move to next screen.

```
 File-AID ---------------  Allocate New SMS Dataset  ---------------------------
COMMAND ===>


Dataset name: USERID9.FASAMP.EMPMAST.NEW

 Management Class    ===> CS843I     (Blank for default)
 Storage Class       ===> CSNORM     (Blank for default)
  Volume serial      ===> CST006     (Blank for authorized default volume)
 Data Class          ===>            (Blank for default)
  Space units        ===> BLKS       (BLKS; TRKS; CYLS; KB; or MB)
  Primary quantity   ===> 120        (In above units)
  Secondary quantity ===> 24         (In above units)
  Directory quantity ===> 0          (Partitioned only)
  Record format      ===> VB
  Record length      ===> 255
  Block size         ===> 6233
  Expiration date    ===>            (YYYY/MM/DD or blank)
  Dataset Name Type  ===>            (Library (PDS/E); PDS; or blank)
  Number of Volumes  ===>            (No. of VOLS or blank for SMS default)
```


Step 5: Key in **1** at the **Option** prompt for selective Copying of records.

```
 File-AID - Selection Criteria Menu - TEMPORARY --------------------------------
OPTION ===> 1
                                             - Status -
     1  OPTIONS      - Enter selection criteria options        default
     2  FORMATTED    - Edit formatted selection criteria        0   sets
     3  UNFORMATTED  - Edit unformatted selection criteria      0   sets


Member list description ===> ________________________________

   Long    ===>
____________________________________________________________________
Description ===>
____________________________________________________________________

Use VIEW command to display selection criteria summary
Use SAVE command to write selection criteria request
Use END to continue processing
Use CANCEL to return to main panel
```


Step 6: We can provide the number of records to skip, select, and start record as options while copying as shown below:

```
File-AID --------------  Selection Criteria Options  --------------------------
COMMAND ===>

Specify Selection Criteria Options:
                                    Start at the following record key
                                    (both blank for start of dataset)
  Starting record key        ===>
    - OR -                              OR at the following RBA or RRN
  Starting RBA or RRN         ===>

  Initial records to skip     ===> 50        then skip this many records

  Subsequent Selection Interval:            then repeat the following
    Records to select         ===> 10       - select this many records
    Records to skip           ===> 5       - then skip this many records
                                    until
  Number of records to search ===> ALL      you have read this many records
  Number of records to select ===> ALL     or selected this many records

SEQ/VSAM processing direction ===> F       (F = Forward; B = Backward)


  Use ENTER to return to selection criteria menu
```


Step 7: Once we key in the required criteria and press **Enter** key, we get to the screen shown below. Here we can press **PF3** key or type 'enter' at command prompt to complete the copying process.

```
File-AID - Selection Criteria Menu - TEMPORARY --------------------------------
 OPTION ===>
                                                        - Status -
      1  OPTIONS      - Enter selection criteria options        default
      2  FORMATTED    - Edit formatted selection criteria       0   sets
      3  UNFORMATTED  - Edit unformatted selection criteria     0   sets


 Member list description ===> ________________________________

    Long    ===>
________________________________________________________________
 Description ===>
________________________________________________________________




 Use VIEW command to display selection criteria summary
 Use SAVE command to write selection criteria request
 Use END to continue processing
 Use CANCEL to return to main panel
```


Step 8: The screen given below shows that 71 records have been copied from the original file.

```
File-AID ---------------------  Copy Utility  -------------- 71 RECORDS COPIED
COMMAND ===>

Specify "FROM" Dataset or HFS Path Information:
  Dataset or path ===> 'USERID9.FASAMP.EMPMAST'
  Volume serial   ===>                    (If not cataloged)

Specify "TO" Dataset or HFS Path Information:
  Dataset or path ===> 'USERID9.FASAMP.EMPMAST.NEW'
  Volume serial   ===>                    (If not cataloged)
  Disposition     ===> OLD                  (OLD, MOD, NEW)


Specify Execution Information:
  Process online or batch   ===> O       (O = Online; B = Batch)

Specify Selection Criteria Information:   (E = Existing; T = Temporary;
  Selection criteria usage  ===> T        M = Modify; Q = Quick; N = None)
  Selection dataset name   ===>
  Member name              ===>        (Blank or pattern for member list)
```


Step 9: The below screen is for direct copying without any selection criteria. Please note that we need to set the selection criteria as **N**.

```
File-AID --------------------- Copy Utility --------------
COMMAND ===>

Specify "FROM" Dataset or HFS Path Information:
  Dataset or path ===> 'USERID9.FASAMP.EMPMAST'
  Volume serial   ===>                  (If not cataloged)

Specify "TO" Dataset or HFS Path Information:
  Dataset or path ===> 'USERID9.FASAMP.EMPMAST.NEW1'
  Volume serial   ===>                  (If not cataloged)
  Disposition     ===> NEW                (OLD, MOD, NEW)


Specify Execution Information:
  Process online or batch   ===> O       (O = Online; B = Batch)

Specify Selection Criteria Information:   (E = Existing; T = Temporary;
  Selection criteria usage  ===> N        M = Modify; Q = Quick; N = None)
  Selection dataset name    ===>
  Member name               ===>         (Blank or pattern for member list)
```

