Introduction to z/OS

Introduction to z/OS - Course Objectives

On successful completion of this class, the student should be able to:

1. Describe the concepts underlying IBM zSeries computer systems in general (hardware: processors, memory, tape, disk, devices in general; software: operating systems, application environment, application programs)

2. Correctly use terms related to mainframe computer systems: such as data set / file, fields, records, data set organization

3. Understand terms specifically related to z/OS, such as DDname, data set name, PDS, PDSE, VSAM, label, VTOC, directory, catalog, TSO, ISPF, JCL

4. Describe the role SMS (System Managed Storage) plays in z/OS

5. Describe the roles played in application development of CLIST, REXX, JCL, and TSO/ISPF

6. Describe the role of Unicode in the mainframe world, and the support for Unicode provided in z/OS

7. Describe capabilities of the latest IBM compilers for COBOL, PL/I, and C as well as the Assembler, the binder, and Language Environment

8. Describe the capabilities of DB2, in broad, general terms, and understand the salient features of the latest version of DB2

9. Compare and contrast the two major transaction processing environments: CICS/TS and IMS, and the role of MQSeries

10. Describe the facilities available under z/OS for running UNIX applications, including a web server and email

11. Send text messages to a cell phone and / or emails to the Internet from a batch job, (providing their system is configured to do so).

Introduction to z/OS - Topical Outline

Introduction: What's Hot? What's Cold? What's Strange?
z/Architecture - A hardware overview
    zSeries
    CPC - Central Processor Complex
    I/O Channels, PR/SM, LPARs, and Sysplex, zBX, Tapes and Disk

z/OS - A software overview
z/OS Workloads
    Capacity utilization
    Workload manager
    z/OS Workloads
    Tuning

z/OS Fundamentals
    Data management terms
    Data organizations
    Sequential data set
    VTOC
    Partioned Data Set (PDS)
    Catalog
    PDSE
    The UNIX File model: the z/OS File System (zFS)
    Batch
    JCL
    TSO/ISPF
    CLIST and REXX
    Dialog manager
    SMS - System Managed Storage

Unicode
    z/OS support for Unicode

DB2 - IBM's Premier relational data base
    The Basics
    Indexes
    DB2 Architecture
    Embedded SQL
    Components
    DB2 LUW

Transaction monitors
    CICS/TS
    IMS
    The role of MQSeries

Languages
    Common threads
    Language Environment (LE)
    Assembler
    Enterprise COBOL
    Enterprise PL/I
    C/C++
    The program binder

z/OS and UNIX System Services
    TSO User ID
    Profiles
    UNIX User ID
    z/OS UNIX - The shell interface under OMVS
    Things you can do under z/OS UNIX
        Standard commands and utilities
        Compile / assemble / bind
        HTTP sever - host web site
        Use sed file to convert flat file to HTML
        Use sendmail and ftp
        Code / compile / run Java

Sending notes, e-mails, and text messages
    Communications possibilities
    Sending emails from a batch job
    Sending text messages from a batch job to a cell phone
    SMTP notes
    Communications possibilities conclusion
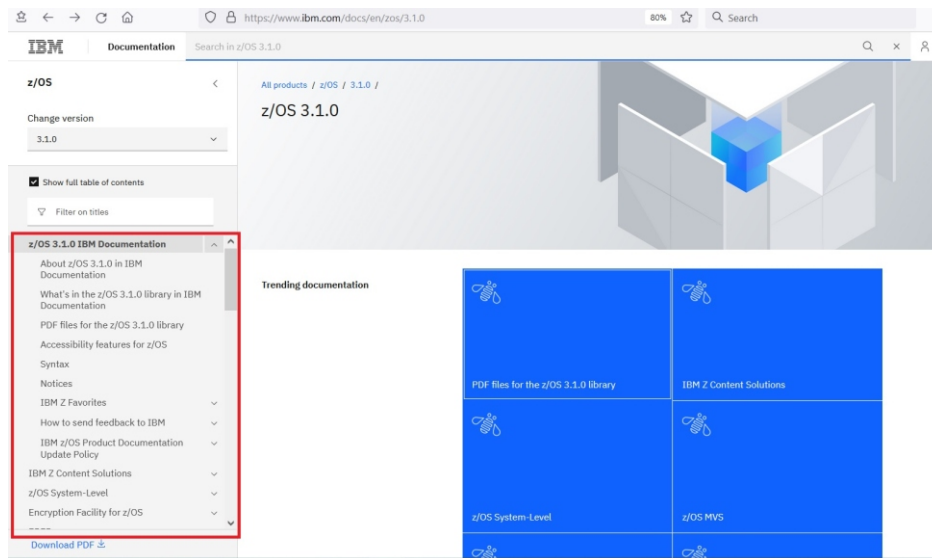

Conclusion - Final thoughts

Sources of information

To get to relevant IBM manuals on the web, start with:

> https://www.ibm.com/docs/en/zos/

You will see a page something like this:

Click on the box for the version of z/OS you are running and you'll see something like this:

On the left hand side is a scrollable box of topics, some even of interest (ISPF, MVS, TSO/E, and so on). If you click on one of these, you will see a list of related documents and you can download the PDF version of any of these, say, ISPF:

Sources of information, continued

Or, if instead you click on the blue box on the right labeled "PDF files for the z/OS v.r.m library", you get a different scrollable list, *e.g.*:

And both lists shown in boxes above are scrollable.

Notice these methods allow you access to a large number of documents, including Assembler (HLASM), REXX, DFSORT, C/C++. But, if you want information on COBOL or PL/I. use these links:

https://www.ibm.com/support/pages/enterprise-cobol-zos-documentation-library

https://www.ibm.com/docs/en/epfz

Sources of information, continued

For information about other components / products:

DB2

      https://www.ibm.com/support/pages/db2-12-zos-product-documentation
      https://www.ibm.com/docs/en/db2-for-zos/13

CICS

      https://www.ibm.com/docs/en/cics-ts/6.1?topic=available-documentation-in-pdf

IMS

      https://www.ibm.com/docs/en/ims

MQ (formerly MQSeries)

      https://www.ibm.com/docs/en/ibm-mq/9.3

What's The Hottest "Thing" in IT?

❒ **The Internet and the World Wide Web**

    Everyone has to have a "web presence"

    Everyone has to have their applications "web enabled"

    Everyone needs to support eCommerce

❒ **What technologies are driving the Internet and the Web?**

    UNIX - which says ASCII and Unicode

    C, C++, Java, perl, awk, php, ruby - all originally from the UNIX world

    HTML, XML - which require Unicode

    TCP/IP - and sockets

    Objects - CORBA, DOM, SOAP

    Open source - *e.g.***: Linux**

    Web Services - dynamic discovery and use of available resources

    Mobile apps - tablets and phones run web-style apps

What's the Coldest, Oldest "Thing" in IT?

☐ **The Mainframe - perceived as**

    1960's technology

    EBCDIC, closed, proprietary

    COBOL, PL/I, Assembler

    Batch

    Expensive - high total cost of ownership

☐ **Although some may concede a few benefits**

    Rock solid - seldom crashes

    High performance - especially I/O and transaction rates

    Secure - never hacked

    Evolutionary - old code still runs, even as new features have been added

    Returns real value - recent survey: 5-10% of corporate net profits from IT attributed to eCommerce, 90-95% to traditional processing

What's The Strangest "Thing" in IT?

❐ **The mainframe has been transformed over the last 15-20 years**

❐ **Today's mainframe has hardware that supports**

    64-bit addressing

    Null-terminated strings, ASCII data, Unicode data

    Support dozens of processors in a single system, with built in cryptograhy, encryption, error detection and correction

    On-chip Artificial Intelligence ("AI") acceleration

    Linux in native mode

❐ **Today's mainframe has software that supports**

    All the traditional environments (CICS, IMS, TSO, batch)

    Relational database that support triggers, stored procedures, remote access, BLOBs and CLOBs, the cloud

    UNIX System Services - which implies

        ✗ Internet / Web capabilities

        ✗ C, C++, Java, perl, awk, php, ruby

        ✗ HTML, XML

        ✗ TCP/IP, sockets, including IPv6 and SSL

Which Means...

❏ **Your mainframe can run all your mission critical, day to day, "bread-and-butter" applications**

   Securely, at higher and higher speeds

❏ **PLUS your UNIX / Intranet / Internet / Web apps**

   All on a single box (or complex of boxes)

❏ **The major pieces of this story are...**

   z/Architecture - a new class of machines; the latest mainframe hardware

   z/OS - the latest mainframe operating system; the 64-bit version of MVS and OS/390

   Language Environment - a software component of z/OS (and OS/390) that enables easier communication between programs written in different languages as well as providing a suite of callable services

   UNIX System Services - the ability to run UNIX on the mainframe

   New compilers for COBOL, PL/I, C, C++, and full Java support

Section Preview

❐ **z/Architecture - A Hardware Overview**

      zSeries

      CPC - Central Processor Complex

      I/O channels

      PR/SM, LPARs, and Sysplex

      Tapes and Disk

❐ **"Mainframe - an obsolete device still used by thousands of obsolete companies, serving billions of obsolete customers, and making huge obsolete profits, for their obsolete shareholders. And this year's run twice as fast as last year's."**
      - Phil Payne, Isham Research

zSeries

❐ **The most recent family of mainframes from IBM is called the "zSeries"**

There are these product lines currently:

- ✗ z800 announced February, 2002; enhanced October, 2002; enhanced December, 2002
- ✗ z890 announced April, 2004
- ✗ z900 announced October, 2000
- ✗ z990 announced May, 2003; enhanced October, 2003
- ✗ z9-109 announced July, 2005; renamed the z9-EC (Enterprise Class) April 2006
- ✗ z9-BC (Business Class) announced April, 2006
- ✗ z10-EC announced February, 2008
- ✗ z10-BC announced October, 2008
- ✗ zEnterprise 196 (a.k.a. z196) announced July, 2010
- ✗ zEnterprise z114 announced July, 2011
- ✗ zEnterprise EC12 (zEC12 for short) announced August, 2012
- ✗ zEnterprise BC12 (zBC12) announced July, 2013
- ✗ IBM Z z13 announced January, 2015
- ✗ IBM Z z14 announced July, 2017
- ✗ IBM Z z15 announced September, 2019
- ✗ IBM Z z16 announced April 2022 (Models A01, A02, AGZ)

All the machines implement z/Architecture, with differing amounts of capacity and speed and features

For our purposes, we use the top-of-the-line z16 model A01 which has more options and features than the other models

- ✗ But the concepts and general architecture applies across the zSeries family of machines

CPC - Central Processor Complex

❑ **All zSeries machines come with multiple processors**

❑ **A single z16 model A01 machine consists of a CPC containing up to 200 characterizable Processor Units (PUs)**

Packaging is in "frames" and "drawers" within frames

Here's an cutting from https://developer.ibm.com/blogs/a-tour-inside-the-ibm-z16 :

But I said I'd get *inside* the IBM z16! At a high level, a fully loaded 4-frame system you can have up to 12 I/O drawers in a Power Distribution Unit (PDU, for air-cooling) system and 4 Central Processor Complex (CPC) drawers. Inside that quartet of drawers, you can have up to 200 processors and 40TB of memory.
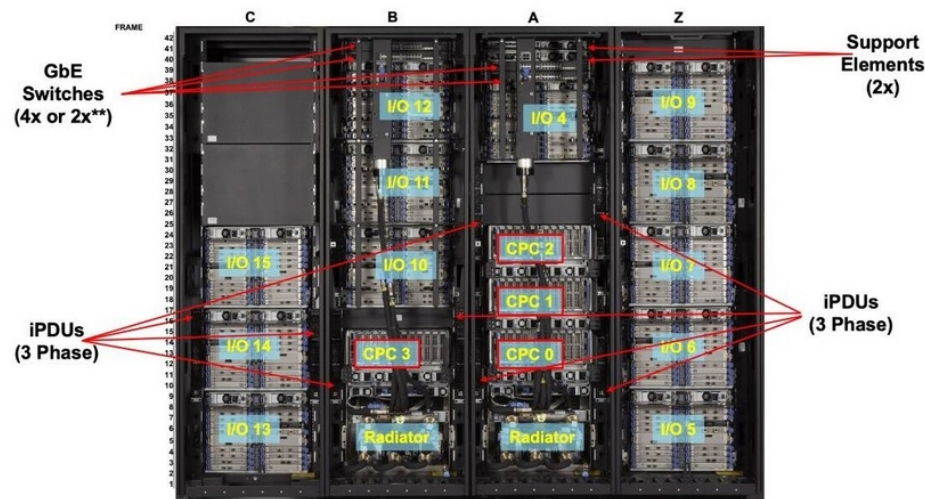
❑ **A maximum configuration weighs 2,667 kilograms (5,880 pounds)**

CPC, 2

❒ **PUs must be <u>characterized</u> - assigned a role which will be fulfilled at Power On Reset (POR) time by the loading of role-specific microcode**

PUs are very flexible and may be assigned any of these roles:

✗ CP - Central Processor (operating system use; can run z/OS, Linux, z/VM, z/TPF, z/VSE, or Coupling Facility Control Code (CFCC))

✗ SAP - System Assist Processor (I/O work support)

✗ ICF - Internal Coupling Facility (can run CFCC, Linux, or z/VM)

✗ IFL - Integrated Facility for Linux (can run Linux, z/VM, or CFCC)

✗ IFP - Integrated Firmware Processor (manages isolation and recovery)

✗ zIIP - z Integrated Information Processor (for 'new' workloads)

❒ **There are some rules / restrictions / limitations**

Which really don't concern us here

CPC, 3

☐ **Miscellaneous notes / points**

CPC has a synonym of CEC (Central Electronic Complex) for historical reasons; CPC is now the preferred term

Each PU includes a Recovery Unit (RU) that keeps a copy of the complete state of the PU (registers, condition code, and so on)

  ✗ If an error is detected while executing an instruction, the instruction is retried once; if the instruction fails again, the information in the RU is transferred to a spare CP and execution continues on that CP; the failing CP is reported

Each CP has a set of cryptographic support instructions included, and two optional crypto assist features may be ordered

A CPC can theoretically connect to up to 63 other CPCs, so 32 CPCs in a parallel sysplex and thus 32*64 = 2048 CPs

zIIPs were introduced to lower the cost of computing: work on these processors does not incur software capacity / usage charges

You can have multiple z16's in a configuration, or mix with earlier models of system z

CPC, 4

❏ **Miscellaneous notes / points, continued**

Each CP is capable of running in 24-bit addressing mode, 31-bit addressing mode, or 64-bit addressing mode and includes

✗ 16 64-bit general purpose registers (GPRs)

✗ 16 64-bit control registers (CRs)

✗ 16 32-bit access registers (ARs)

✗ 16 64-bit floating point registers, supporting:

➢ short, long, extended Hexadecimal Floating Point

➢ short, long, extended Binary Floating Point

➢ short, long, extended Decimal Floating Point

✗ 128-bit program status word (PSW)

Hardware instructions support these data types

✗ Character string - EBCDIC, ASCII, Unicode (UTF-8, UTF-16, and UTF-32); fixed length; null-terminated

✗ Bit string

✗ Packed decimal

✗ Binary integer - halfword, fullword, doubleword

✗ Floating point (see above)

✗ Conversions between these, as appropriate

✗ Addresses - 24-bit, 31-bit, 64-bit

I/O Channels

❑ **All zSeries machines have channel subsystems, composed of channels of various kinds, and connectors / adapters as configured by the customer**

A channel itself is a special processor with buffers and an adapter in a zSeries I/O cage

For I/O to external devices, channels connect to control units which, in turn, connect to devices such as printers, tape drives, and disk drives

Channels can also connect to

✗ LAN / WAN networks

✗ Outboard cryptography devices

✗ Other zSeries systems - internal (PU-PU communications) and external (CPC-CPC communications)

✗ Sysplex timer to coordinate PU clocks

A wide variety of channel types are supported, but not discussed here

External connections are generally fiber optic

External control units may be up to 100 km from the CPC (using repeaters and extenders)

PR/SM, LPARs, and Sysplex

❒ **zSeries machines come with a component called PR/SM (for Processor Resource / System Manager, pronounced "prism") which has a feature called Logical Partitioning (LPAR)**

> **The LPAR feature allows you to configure a single CPC into multiple LPARs (Logical Partitions)**

> **Each LPAR (common usage is LPAR, but LP is more accurate) consists of one or more CPs, I/O channels, and memory (main storage)**

> > ✗ Running a copy of z/OS, Linux, z/VM, z/TPF, CFCC, or z/VSE

> **The physical CPs, I/O channels, and memory can be dedicated to an LPAR, or can be shared among multiple LPARs**
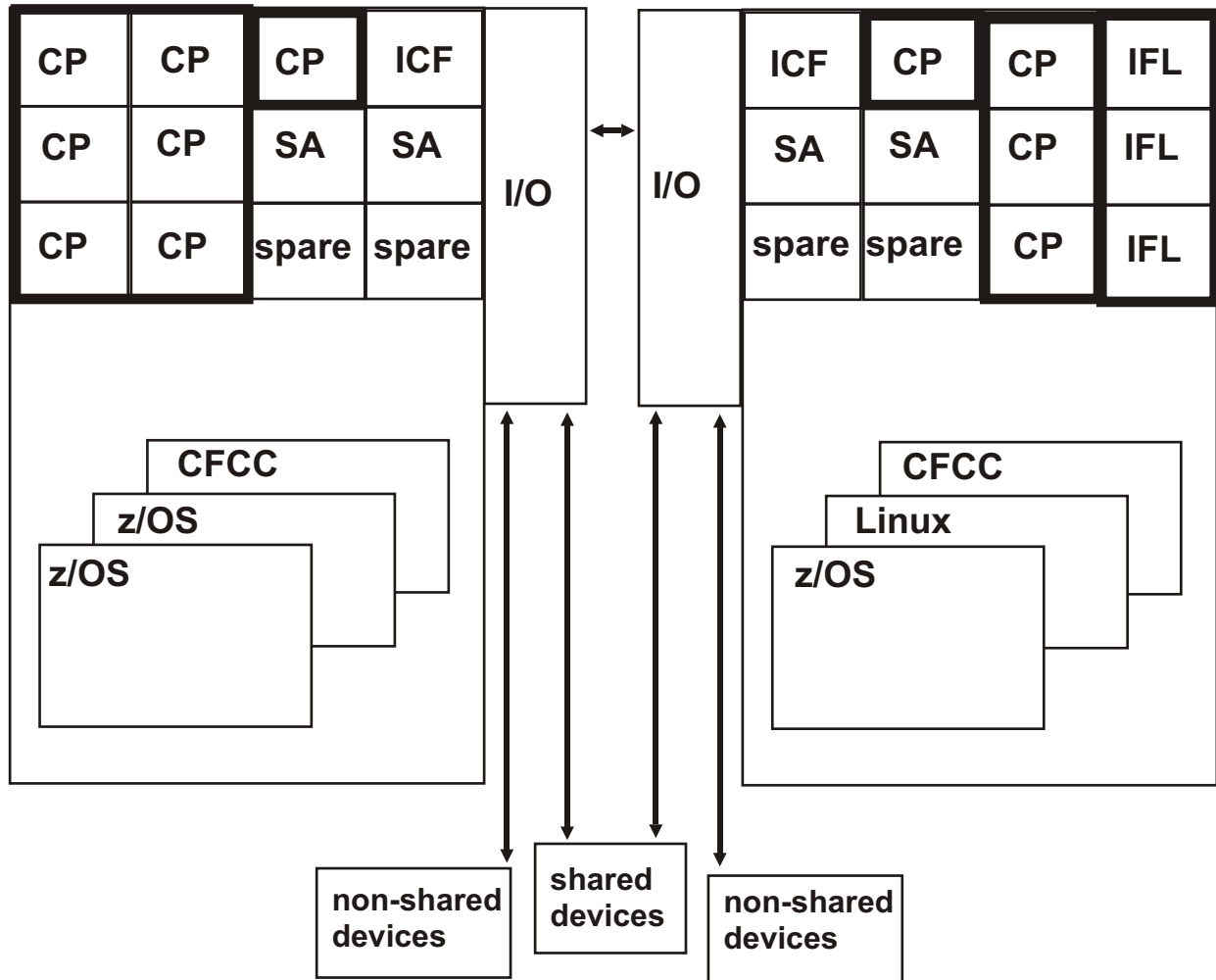
> **The collection of LPARs running in a CPC, communicating with each other using z/OS software, is called a <u>Sysplex</u>**

> **A collection of CPCs, each with one or more LPARs, communicating with each other using CFCC is called a <u>Parallel Sysplex</u>**

> > ✗ The CPCs can be up to 100km apart from each other, forming a <u>**Geographically Dispersed Parallel Sysplex**</u>

More LPARs and Sysplex

☐ **As a conceptual diagram, we show two CPCs each containing 12 PUs divided into various LPARs running various OSs**

CP	CP	CP	ICF
CP	CP	SA	SA
CP	CP	spare	spare

I/O ↔ **I/O**

ICF	CP	CP	IFL
SA	SA	CP	IFL
spare	spare	CP	IFL

CFCC
z/OS
z/OS

CFCC
Linux
z/OS

non-shared devices   **shared devices**   **non-shared devices**

☐ **Not shown: zIIPs, lots of details and options**

Tapes and Disk

❏ **The peripheral attached to a mainframe system today fall into these categories:**
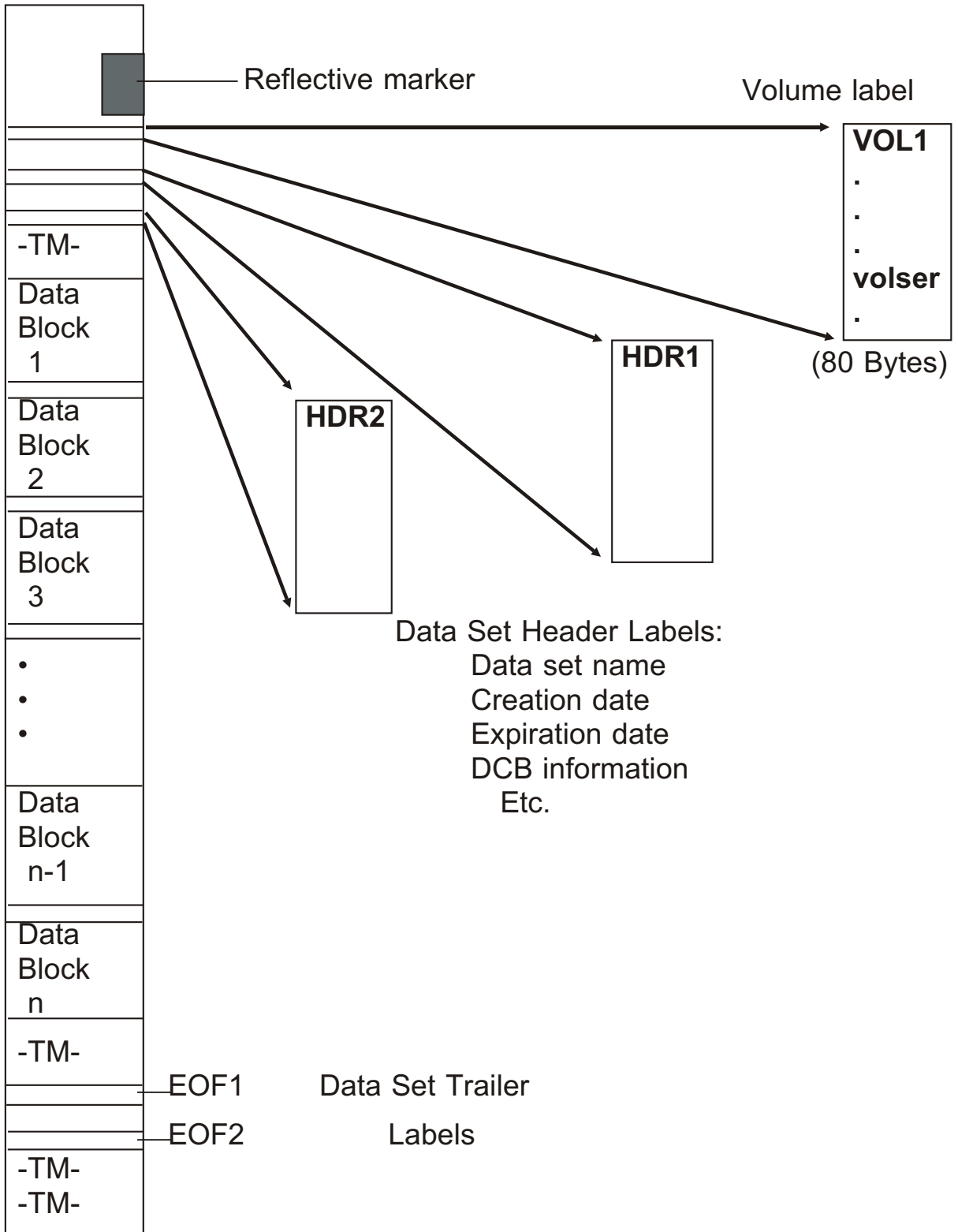
       Printers - all kinds

       Terminals / consoles

       Networks - LANs / WANs / SNA / TCP/IP

       Tape - discussion next

       Disk - what IBM calls DASD; discussed after tape

Tape Layout

Reflective marker

Volume label

VOL1
.
.
.
volser
.

(80 Bytes)

HDR1

HDR2

-TM-

Data
Block
1

Data
Block
2

Data
Block
3

•
•
•

Data
Block
n-1

Data
Block
n

-TM-

EOF1   Data Set Trailer

EOF2   Labels

-TM-
-TM-

Data Set Header Labels:
    Data set name
    Creation date
    Expiration date
    DCB information
        Etc.

Tape Media

☐ **The previous page describes the layout of records on an IBM standard labelled tape**

    Note that for tape the larger the block size the better, in terms of using the media and usually in terms of performance

    Maximum block size for tape is 256KB, with design allowing blocksize up to 2GB in the future

    Theoretical capacity about 180MB for a standard reel of tape

☐ **The newest technologies do not physically record data on tape this way, but they logically simulate this arrangement, so your program (and your JCL) cannot tell the difference**

Tape Media, 2

❏ **The latest technologies involve**

    Cartridges ("square tape") replacing traditional reels ("round tape")

        ✗ No one manufactures reel tape anymore

    New generations of cartridges - up to 800MiB capacity
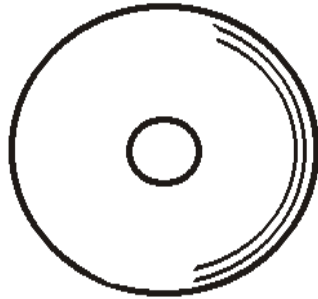
    Hardware data compaction - effectively giving as much as 900GiB per cartridge

        ✗ Selectable through JCL (**TRTCH=COMP** or **TRTCH=NOCOMP** on the DD statement at create time)

    Automated (robotic) cartridge mounting and demounting
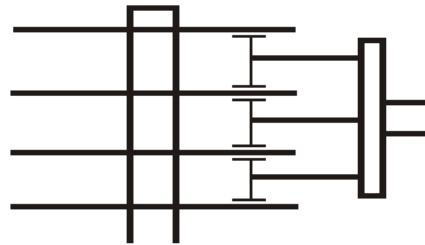
    WORM (Write Once Read Many) technology - for archiving

DASD Concepts

Concentric tracks

Platter

Cylinder concept

Read / Write heads

☐ **D**irect **A**ccess **S**torage **D**evice

DASD

DASD - Reality

❏ **As with tape, DASD is not physically manufactured like this anymore**

    Mostly use mini-disc type technology, RAID, and simulate classic 3390 DASD architecture in terms of bytes per track, tracks per cylinder, and so on

        ✗ RAID - Redundant Array of Independent Disks (in some literature, the "I" stands for "inexpensive")

            ➢ Supports multiple copies and extra error checking

❏ **So although tape and disk do not look like they used to, physically, they look like they used to logically**

    A big help in maintaining consistency and continuity of programs, JCL, commands, and so on

    Currently, the largest logical volume can store about 55GiB of user data

        ✗ With the latest version of z/OS and physical drives, a new option, the Extended Address Volume (EAV) offers capacity of up to 64 PB of user data per EAV (a petabyte [PB] is 1,000 terrabytes [TB] or 1,000,000 gigabytes [GB]

❏ **Also, many DASD storage units can be attached to heterogenous systems, for sharing capacity**

    z/OS, Windows, and UNIX systems can share different logical volumes within the same physical box, for example (non-trivial)