

# Formação COBOL

## Alta Plataforma Mainframe IBM

Professor: Vagner Bellacosa

Disciplina: Cobol & Além

# COBOL

## Modulo 01.02 – Mainframe



- 1) Vagner Bellacosa seu facilitador
- 2) Supercomputador Versus Mainframe : Diferenças marcantes
- 3) Porque usar o Mainframe?
- 4) Host e a virtualização
- 5) Disponibilidade
- 6) Single System
- 7) IBM GDPS
- 8) Escalabilidade
- 9) Segurança
- 10) Compatibilidade Continua
- 11) Arquitetura sempre em evolução
- 12) Arquitetura von Neumann

# COBOL

## Modulo 01.02 – Mainframe



- 13) Storage
- 14) IPL
- 15) Address Space
- 16) Mainframe e seus componentes
- 17) Uso de Memoria
- 18) Armazenamento em Disco 3390
- 19) LPAR
- 20) Duvidas

# COBOL



## 1) Vagner Bellacosa seu facilitador

Analista programador desde 1989, onde comecei como auxiliar e tecnólogo em processamento de dados desbravando os primórdios da computação brasileira, um eterno aprendiz em processos Mainframe..

Desde então trabalhei em centenas de projetos, em 4 países e dezenas de instituições financeiras, ora como funcionário, consultor externo e freelancer.

A mais memorável foi o Banco REAL, uma verdadeira escola, onde aprendi muito e tive a oportunidade de participar no aliciante Projeto Y2K, o temível bug do milênio.

Particpei em projetos na CESP, Fundação CESP, Transbrasil, ABSA, Real Seguros, BPN, Skandia, DGITA, BES, CGD, BPI, Barclay, Skandia, IBM Italia, Sistemi Informativi, Unicredit, Zurich Assicurazione, Banco Safra e Banco Itaú.

Atualmente trabalho na Spread no Projeto BRB em Brasília, via remoto.

# COBOL

**Supercomputador  
Deucalion**



**IBM Mainframe Z16**



## 2) Diferenças marcantes.

Um SuperComputador é construído para uma necessidade específica que necessita de grande poder computacional (cpu + memória) e ao mesmo poder em cálculo matemático, por exemplo INPE Tupã ( cálculo meteorológico ) e o Deucalion estudos de engenharia.

O Mainframe é vocacionado ao mundo empresarial com grande capacidade para atender demandas governamentais e empresariais.

# COBOL



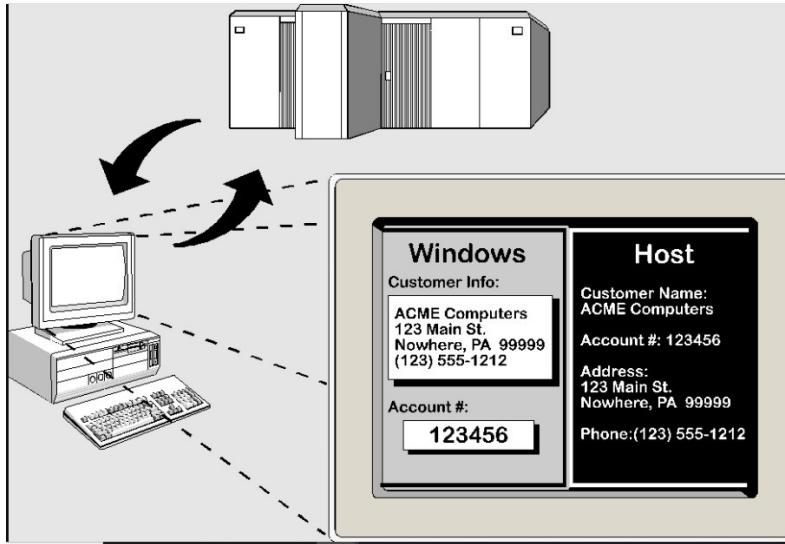
## 3) Porque usar o Mainframe?

Six Sigma em processos garante uma disponibilidade 99,9999%, menos de 5 minutos de parada por ano e redundância de equipamentos.

A IBM garante aos clientes

- 1) Disponibilidade
- 2) Escalabilidade
- 3) Segurança
- 4) Compatibilidade Continua
- 5) Arquitetura sempre em evolução

# COBOL



## 4) Host e a virtualização

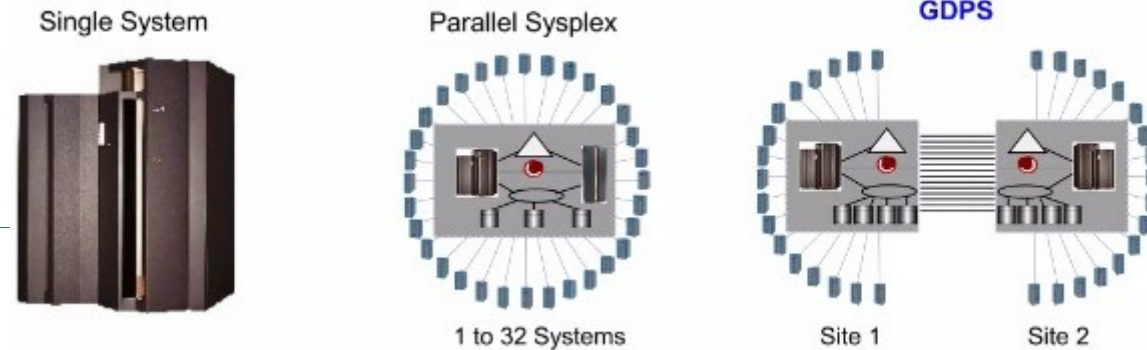
Um Sistema Remoto é acessado por uma conexão local com o Mainframe, disponibilizada via TCP-IP e software zOS é carregado no micro por via Emulador 3270, garantindo o acesso ao Sistema Central a partir de qualquer microcomputador seja seu S.O. Windows, Linux ou MacOS.

Com o avanço da miniaturização, os computadores avançaram das válvulas para os microchips e aumentaram o poder computacional com grande capacidade de cálculo, alocação de memória e espaço de armazenamento com menor custo.

Temos o melhor dos mundos: a facilidade do uso do Microcomputador com o grande poder de processamento do Mainframe.

O usuário pode solicitar serviços, extração de dados e trabalhar localmente em seu computador.

# COBOL



## 5) Disponibilidade

É uma medida de frequência com que o Mainframe esta disponível para uso com total capacidade.

O Mainframe tem inúmeras rotinas de validação que constantemente monitora a saúde de todos os componentes internos e externos, periféricos, rede e em caso de anomalia inicializa processos de redundância para não ocorrer paradas.

Garantindo a manutenção e troca de componentes com o mainframe em funcionamento, sem o usuário final se aperceber destas paradas.

A IBM garante o funcionamento 24 horas por dia, 7 dias por semana e 365 dias por ano e seu hardware tem garantia de 100 anos.



# COBOL



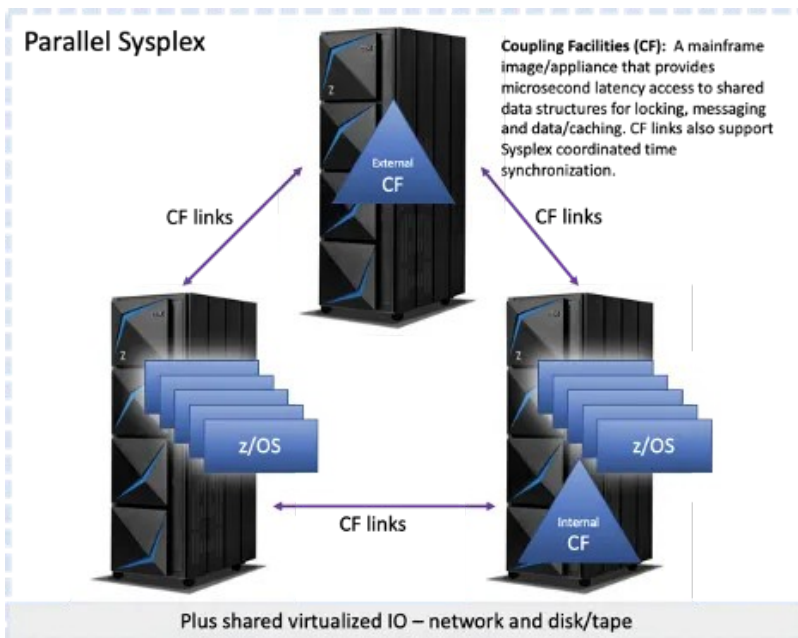
## 6) Single System

De acordo com a necessidade em MIPS ( milhões de instruções por segundos) é a medida de desempenho dos processadores e também uma das formas que a IBM cobra os serviços.

A arquitetura Single System é a utilização de um único mainframe com sua capacidade de processamento, memória e armazenamento em um único local para atender as necessidades do cliente.

Lembrando que o cliente ao adquirir um produto IBM adquire o hardware, software e consultoria técnica.

# COBOL



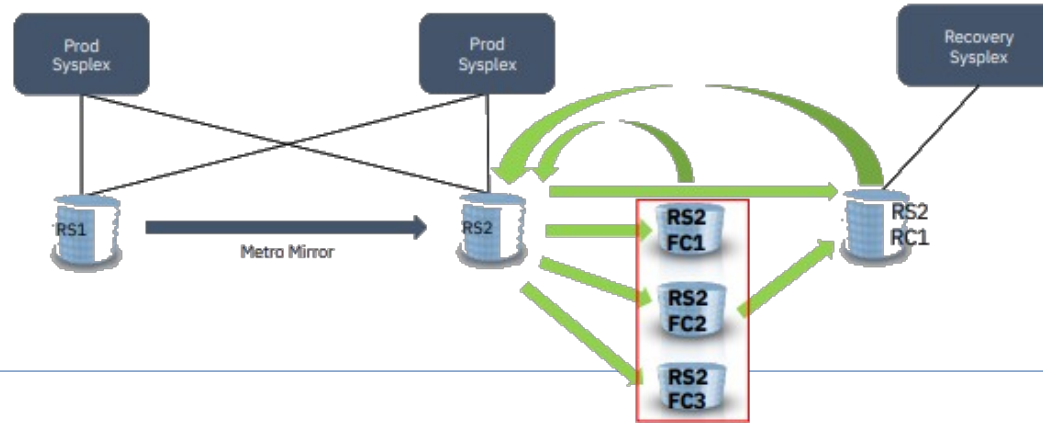
## 6) Parallel Sysplex

Este recurso permite unificar até 32 mainframes num único Mainframe virtual, conectado e funcionando com uma plataforma de computação lógica única.

São executadas diversas cópias do z/OS a partir de um único ponto de controle compartilhando todos os seus recursos (hardware e software).

Controlando todos os pontos de falha e caso um dos elementos fique offline, os restantes assumem sua funcionalidade.

# COBOL



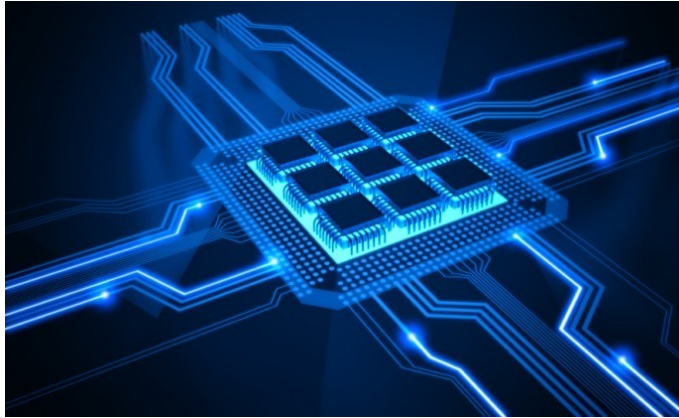
## 7) IBM GDPS

É uma família de software de recuperação de desastre e resiliência para IBM zOS, garantido em caso de desastre, imediatamente uma máquina alternativa sobe e assume o processamento, garantindo a continuidade dos serviços de um mainframe.

Por segurança o ideal é termos mainframes distantes até 50 quilômetros entre si, protegendo contra ataques nucleares, paneis elétricas, falha nas telecomunicações.

Este software garante a replicação dos dados e a garantia de backups e funcionamento sem paradas, causando menor impacto as atividades dos clientes.

# COBOL



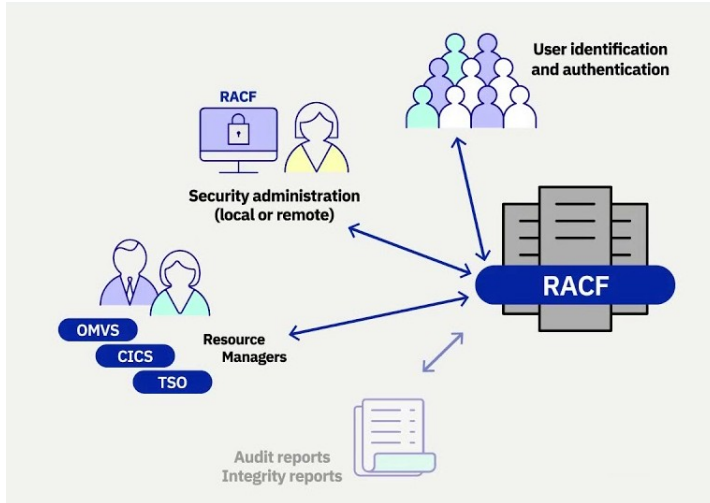
## 8) Escalabilidade

Permite adicionar recursos ( cpu, memoria e hds) para aumentar o desempenho de uma maquina.

A Escalabilidade Vertical adiciona recursos numa única maquina limitado a quantidade de slots (channels) disponíveis.

A Escalabilidade Horizontal adiciona mais maquina aumentando drasticamente o poder computacional.

# COBOL



## 9) Segurança

Um ambiente totalmente seguro, quase a prova de hackers, são diversas funcionalidades que garantem os dados e equipamentos seguros, seja ataque intencionais como acidentais.

Um grande destaque é o RACF, o xerife que garante o controle de acesso a todos os recursos.

Curiosamente os mainframe chama-se família Z, e este Z refere-se a Zero Downtime, ou seja, nunca estão offline.



# COBOL



## 10) Compatibilidade Continua

A IBM garante que qualquer código codificado para Mainframe independente do ano e equipamento, continua a funcionar sem alteração ou recodificação.

No absurdo programas da década de 50 do século passado, poderão funcionar no z16 de nossos dias.

Garantindo economia de recursos em recodificação e recompilações.

# COBOL



## 11) Arquitetura sempre em evolução

Desde o primeiro mainframe lançado em 1954 até os nossos dias, a IBM e seus laboratórios criam novos equipamentos, aumentando a capacidade de armazenamento, o poder de processamento de dados e o uso de memória.

Foram inúmeros lançamentos que com o passar dos anos acabaram se popularizando e migrando entre plataformas.

Lembrando que um dos primeiros computadores pessoais foi lançado pela IBM e em tempos existia o OS2, que era um sistema operacional da IBM para microcomputadores.

# COBOL

## 12) Arquitetura von Neumann





# COBOL



## 13) Storage

Em termos simples, computadores modernos, ou terminais, conectam-se aos dispositivos de armazenamento, diretamente ou por meio de uma rede.

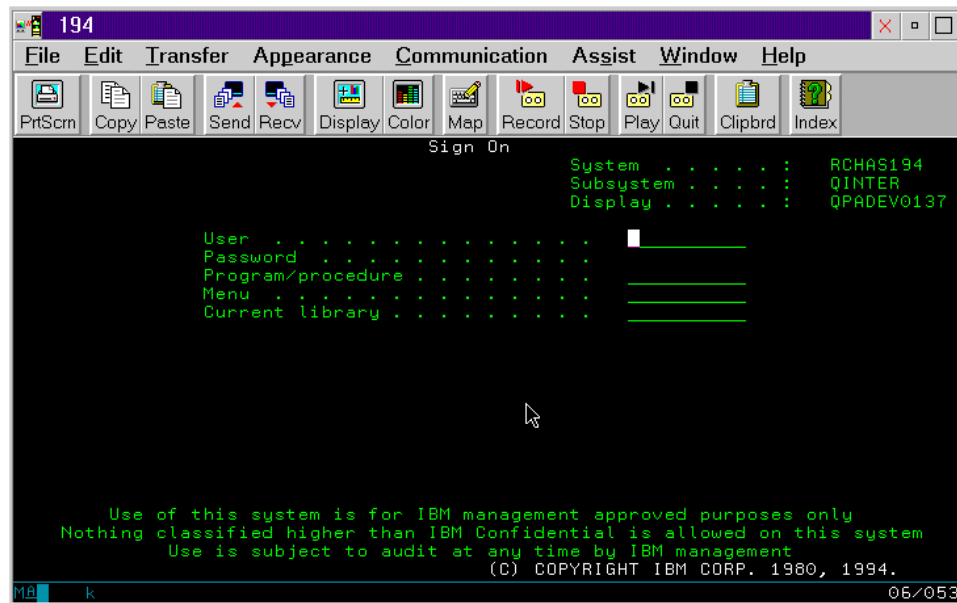
Os usuários instruem os computadores a acessarem e armazenarem os dados nesses dispositivos de armazenamento.



Entretanto, em um nível básico, existem dois fundamentos para o armazenamento de dados: a forma que os dados têm e os dispositivos em que eles são registrados e armazenados.

Atualmente existem 2 tipos de unidades externas para armazenamentos: cartridge e disco rígido, em algumas instalações ainda possuem leitores de fita.

# COBOL



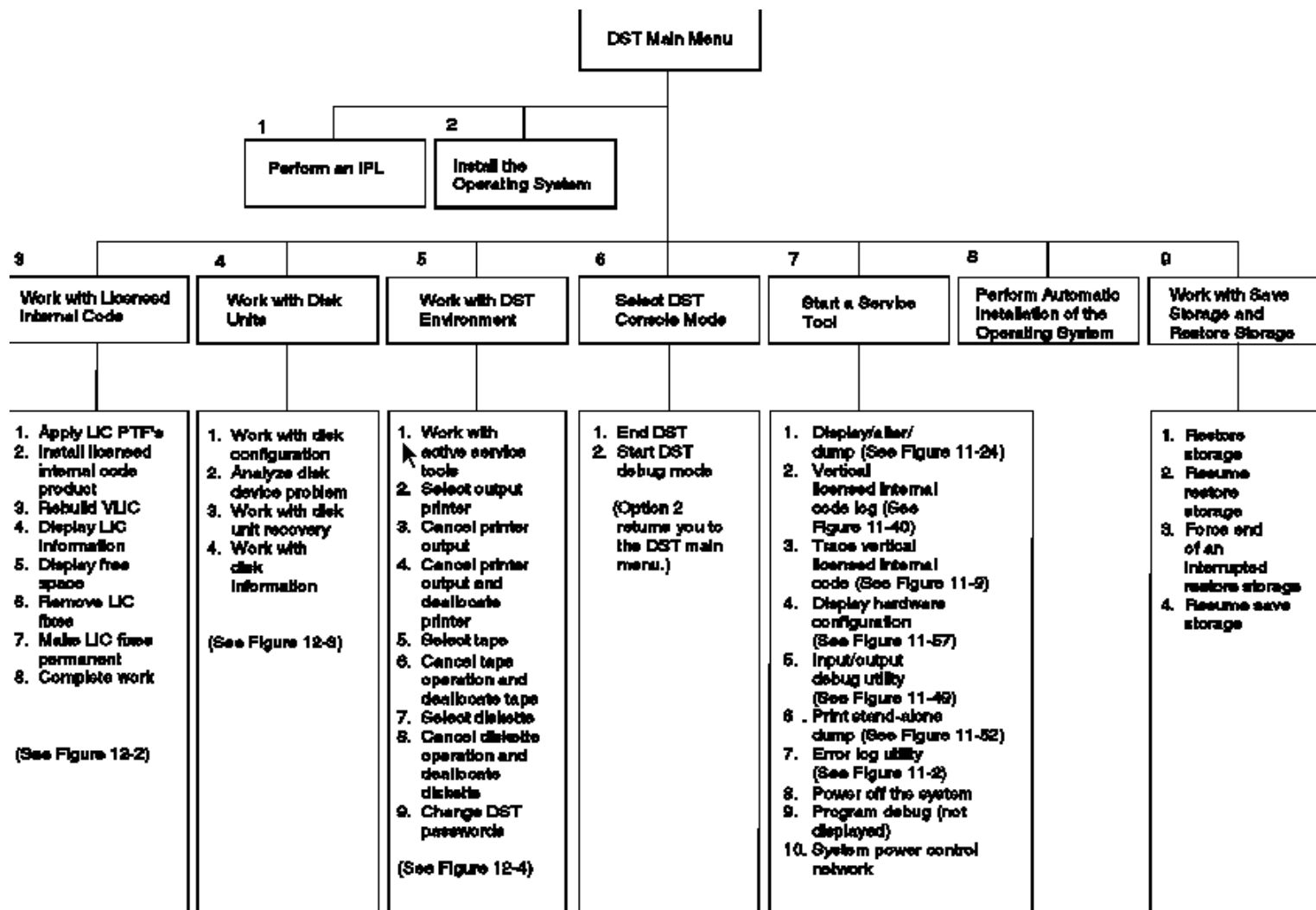
## 14) IPL

Initial Program Load é a operação de Start do Mainframe, seria o equivalente ao antigo Autoexec.bat e Config.sys dos microcomputadores pré-Windows.

Em algum momento, quando um mainframe é ligado, ocorre este processo de carregamento de softwares, teste de hardwares e início do funcionamento do computador.

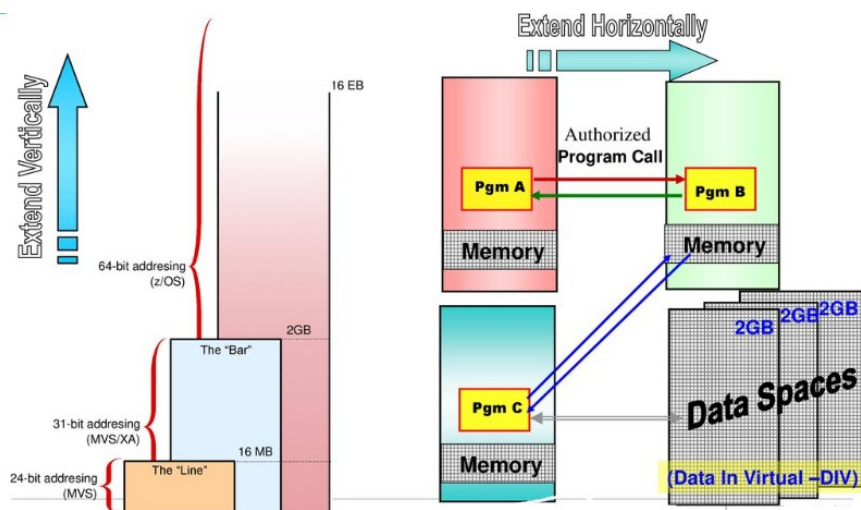
Esta atividade é executada pela análise de Produção sendo invisível para os desenvolvedores, pois o processo é muito, mas muito rápido.

# COBOL



# COBOL

## 15) Address Space



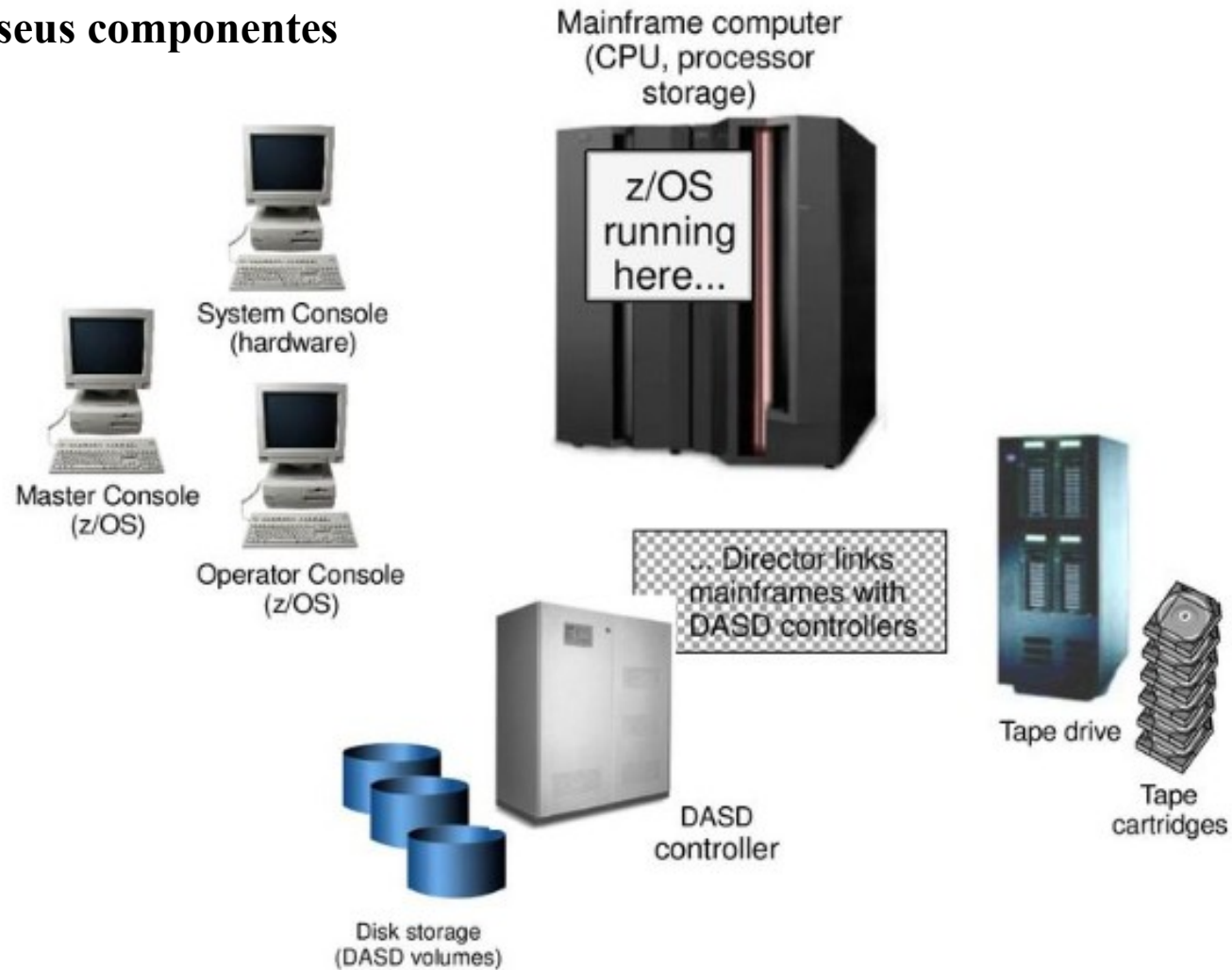
O intervalo de endereços virtuais que o sistema operacional atribui a um usuário ou programa executado separadamente é chamado de **espaço de endereço**. Esta é a área de endereços virtuais contíguos disponíveis para execução de instruções e armazenamento de dados.

O intervalo de endereços virtuais em um espaço de endereço começa em zero e pode se estender até o endereço mais alto permitido pela arquitetura do sistema operacional.

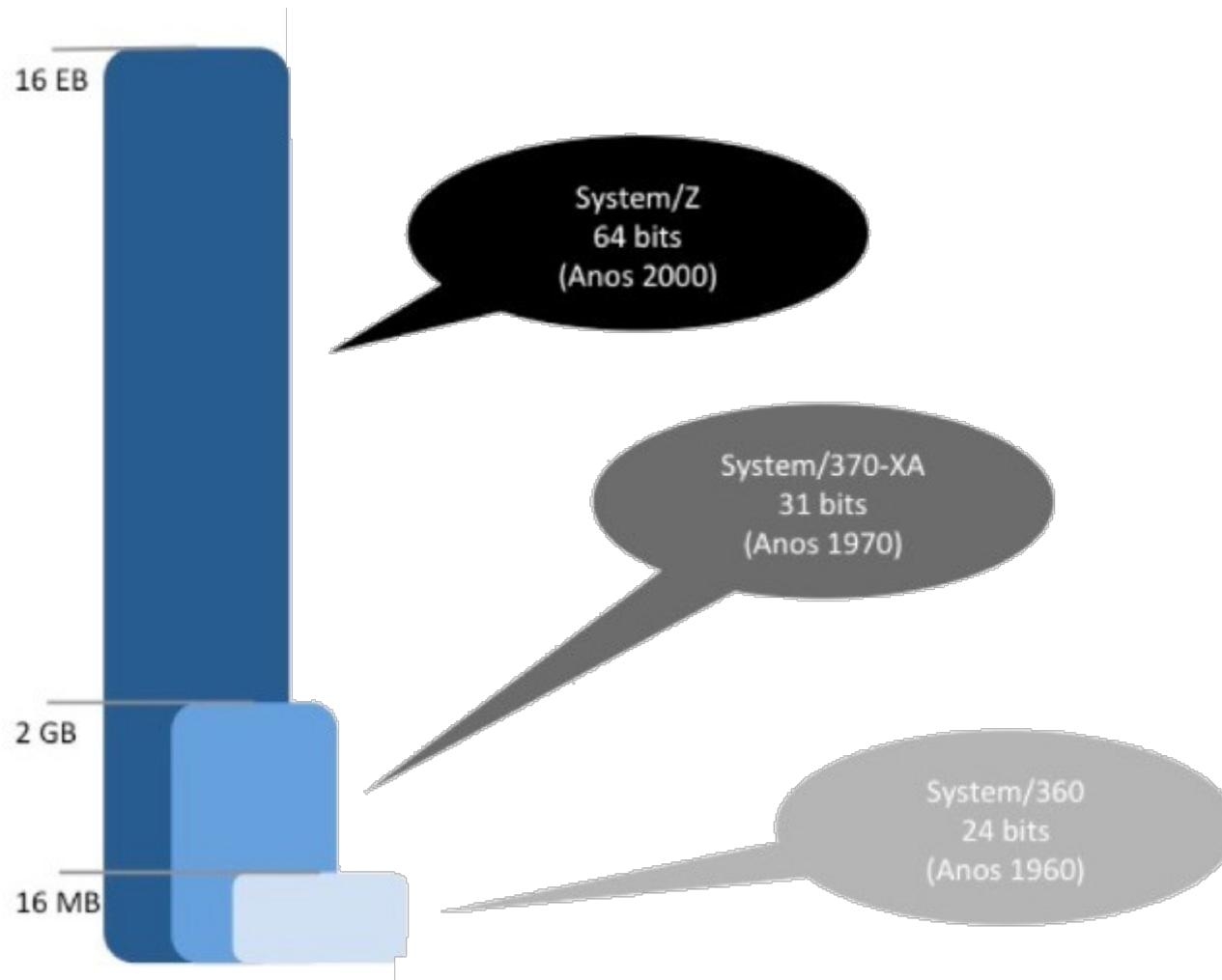
O z/OS fornece a cada usuário um espaço de endereço exclusivo e mantém a distinção entre os programas e os dados pertencentes a cada espaço de endereço. Dentro de cada espaço de endereço, o usuário pode iniciar múltiplas tarefas, utilizando blocos de controle de tarefas ou TCBs que permitem multiprogramação.

# COBOL

## 16) Mainframe e seus componentes



# COBOL



## 17) Uso de Memória

Lembra a história da compatibilidade, por isso vemos estes 3 tipos de endereços de memória.

Originalmente 24 bits, evoluindo para 31 bits e atualmente 64 bits.

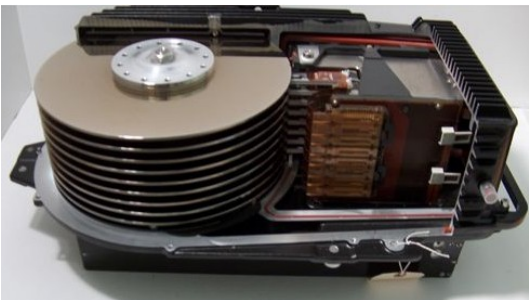
# COBOL



## 18) Armazenamento em Disco 3390

Os volumes DASD são usados para armazenar dados e programas executáveis (incluindo o próprio sistema operacional) e para armazenamento temporário de trabalho. Os rótulos DASD identificam os volumes DASD e os conjuntos de dados que eles contêm.

Um volume DASD pode ser usado para muitos conjuntos de dados diferentes e o espaço nele pode ser realocado e reutilizado. Em um volume, o nome de um conjunto de dados deve ser exclusivo. Um conjunto de dados pode ser localizado por tipo de dispositivo, número de série do volume e nome do conjunto de dados.

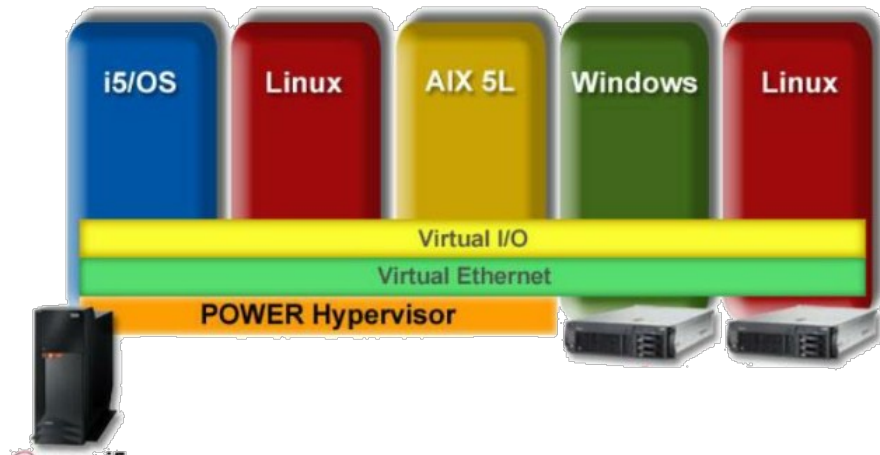


Embora os volumes DASD sejam diferentes em aparência física, capacidade e velocidade, eles são semelhantes em gravação de dados, verificação de dados, formato de dados e programação. A superfície de gravação de cada volume é dividida em muitas **trilhas** concêntricas .

O número de faixas e sua capacidade variam de acordo com o dispositivo. Cada dispositivo possui um mecanismo de acesso que contém cabeçotes de leitura/gravação para transferir dados à medida que a superfície de gravação passa por eles.



# COBOL



## 19) LPAR

Logical Partition é o mainframe virtual criado pelo PR/SM, criado pela IBM na década de 70, atualmente é possível criar até 60 LPARS num mainframe z/Series.

Cada LPAR funciona como um mainframe independente e isolado, possibilitando instalar outros z/OS inclusive Linux e receber um IPL sem afetar os outros LPAR.

PR/SM, ou prisma: Processor Resource System Manager é um recurso de hardware que surgiu na década de 70 que aumentaram o poder dos LPARS.



# COBOL

```

A - SHIELD6
File Edit View Communication Actions Window Help
Work with Communication Resources
System: S10E8AFT
Type options, press Enter.
5=work with configuration descriptions 7=Display resource detail

Opt Resource      Type Status      Text
-  CMB02           6B03 Operational Comm Processor
-  LIN01           6B03 Operational Comm Adapter
-  CMN02           6B03 Operational Comm Port
-  CMB03           6B03 Operational Comm Processor
-  LIN02           6B03 Operational Comm Adapter
-  CMN01           6B03 Operational Comm Port
-  CMB04           268C Operational Comm Processor
-  LIN03           268C Operational LAN Adapter
-  CMN03           268C Operational Ethernet Port
-  CMB05           268C Operational Combined function IOP
-  LIN04           6B26 Operational Comm Adapter
-  CMB06           290B Operational Comm Processor
-  CTL02           290B Operational Comm Adapter
-  CMB07           290B Operational Comm Processor
-  CTL01           290B Operational Comm Adapter

F3=Exit F5=Refresh F6=Print F12=Cancel
MA* A
07/003
  
```

Duvidas???