Intro to z/OS: Part 1

Stephen Warren
Senior Technical Staff Member
Client Architect, IBM Worldwide Client Experience Center

November 2019
Session AF




Introduction

- This **class will cover** (at the 50,000 foot level) many **different areas** (but not all) of **z/OS**

Introduction

- This **class will cover** (at the 50,000 foot level) many **different areas** (but not all) of **z/OS**

- Don’t be overwhelmed. There is a ton of material here. The purpose of this session is to acquaint you with some of the major concepts of z/OS

Agenda

Part 1:

Why Z Matters

Hardware/LPAR

z/OS Components

Software Stack

App Dev, App Exec and Mgmt Envs

DASD

Data Sets / Allocation

TSO/E

ISPF

z/OS UNIX/ISHELL/OMVS/Remote

Address Spaces & Modes

Storage & DAT

Part 2 – Next

Batch Processing/JES/JCL

SDSF

Job Flow

System Log

VTOC & Catalogs

PDS & PDSE

SMS

IPL

Sysplex/GDPS

Serialization

Managing Workloads

Why Z Matters?

Why Z Matters

Utilities 

 **Retail**

 **Government**

Financial 

 **Healthcare**

Travel 

Telecoms 

 **Automotive**

Weather 

Why Z Matters

92 of the top 100 worldwide banks run on z

Why Z Matters

10 out of 10 of the world's largest **insurers**

Why Z Matters

>90% of the US's largest **retailers**

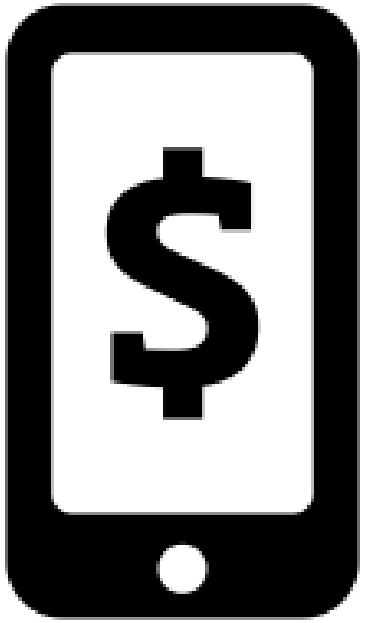
Why Z Matters

>90% of the US's largest **airlines**

Why Z Matters

1.3 million

CICS transactions every second of every day

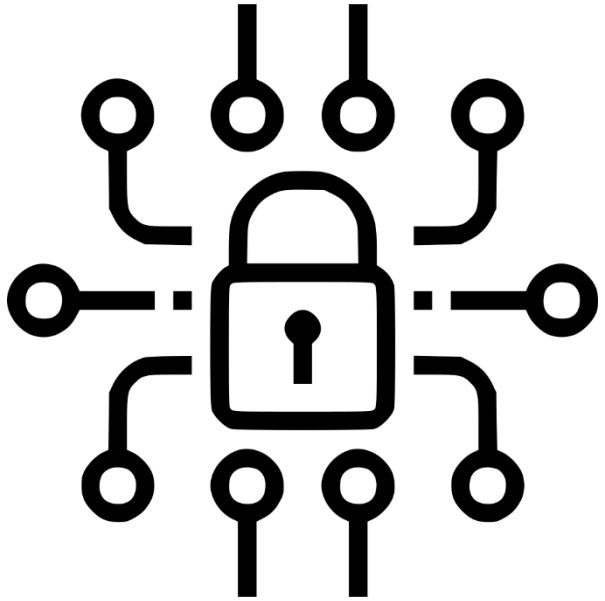


In comparison, there are **68,542** Google searches **every second** globally

Why Z Matters

The average cost of a security breach in the US is estimated at **$11 million**

IBM Z has the **highest** server security rating in the industry



Security is built into every level of the mainframe's structure, including the processor, operating system, communications, storage and applications

Why Z Matters

Mainframes process **30 billion**
business transactions per day

Mainframes process **1.3 million**
CICS transactions. Every second. Every day.

Mainframes enable **$6 trillion**
in card payments annually

80 percent of the world's corporate data originates on mainframes

91 percent of CIOs said new customer-facing apps are accessing the mainframe

Naming of OS

z/OS Lineage

- At the beginning there was:
  - OS/360 (1966)
    - Configuration options PCP, MFT and MVT
  - OS/VS2 R1 (~1972)
    - Retroactively renamed to SVS
  - OS/VS2 R2 (~1974)
    - Also called **MVS**
  - MVS/370 (~1977)
    - MVS/SP V1
    - Optionally DFDS and DFEF
  - MVS/XA (1983)
    - MVS/SP V2 or V3 and MVS/DFP
  - MVS/ESA (~1990)
    - MVS/SP V4 or V5
    - Either MVS/DFP or DFSMS/MVS
  - OS/390 (1996)
  - **z/OS** (2001)

Since **2001** the **name** has **remained z/OS**

This is the **longest** that it has **been called one name!**

Check Your knowledge

Check your Knowledge

- What do you think the “Z” stand for in z/OS ?

Check your Knowledge

- What do you think the “Z” stand for in z/OS ?
  - **Zero downtime**

Hardware Context of z/OS

IBM Z Server – the IBM z15

IBM Z Server – Support Element

Bottom layer is the *Support Element* (SE)

Provides the **human interface** (technicians and system programmers) to the CPC

Support Element (SE)

IBM Z Server – H/W, Microcode

Next layer is the **hardware** and **microcode** (a.k.a. **Firmware, BIOS**)

CP, Memory, I/O

Support Element (SE)

IBM Z Server – H/W, Microcode

Next layer is the **hardware** and **microcode** (a.k.a. **Firmware**)

Software that is **built into the hardware** and is **not accessible** by the user

CP, Memory, I/O

Support Element (SE)

IBM Z Server – H/W, Microcode

Next layer is the **hardware** and **microcode** (a.k.a. **Firmware**)

Up to 190 processors (CPs a.k.a. CPUs) are supported

64GB to 40TB of memory is supported

CP, Memory, I/O

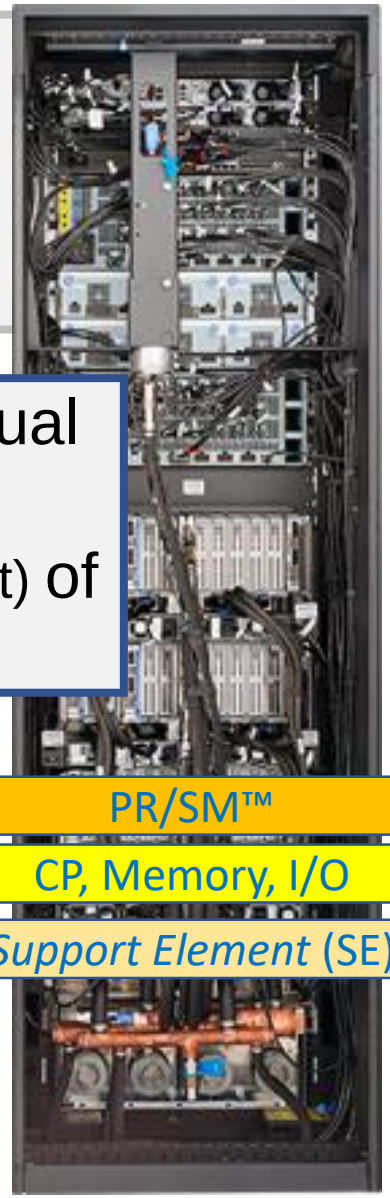
Support Element (SE)

IBM Z Server – PR/SM™

Next layer is *Processor Resource/Systems Manager* (PR/SM)

A **hypervisor** that managers virtual machines (enables the logical partitioning function (coming up next) of the CPC)

Exists in the **firmware** (software embedded in hardware and can be **updated**) of the CPC



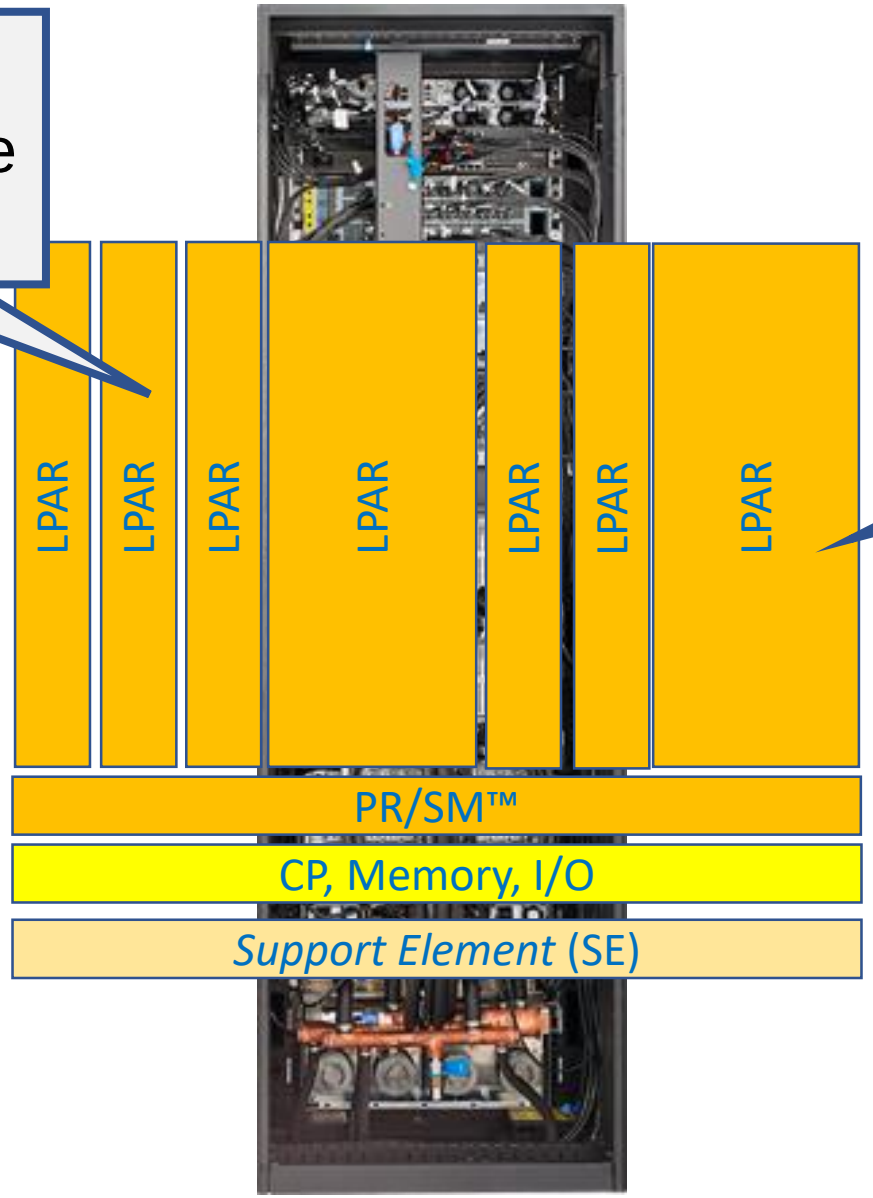
IBM Z Server – Single LPAR

Customer specifies the number of ***Logical Partitions*** (LPARs)

Think of an LPAR as **equivalent to a separate** CPC (i.e., a **virtual machine**)

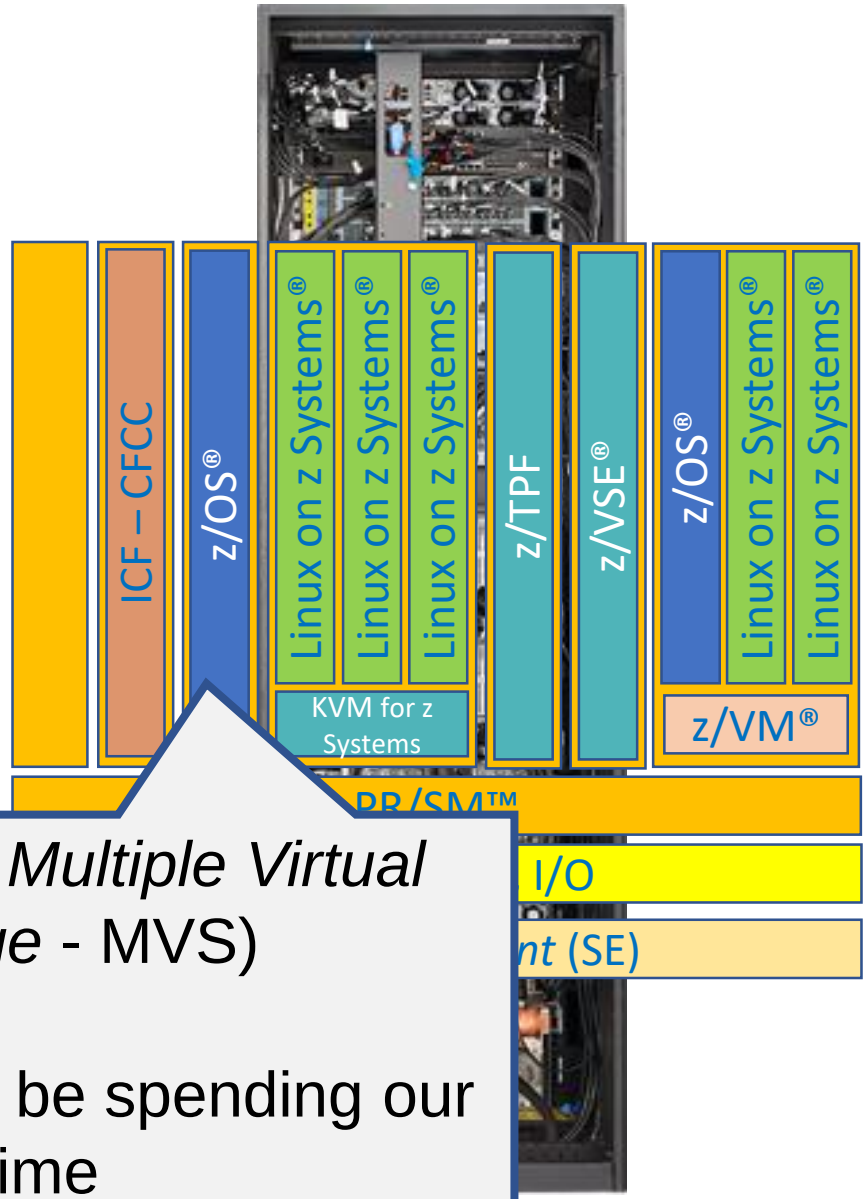
IBM Z Server – Multiple LPARs

Up to 85 LPARs can be defined



Each LPAR can be **allocated different amounts** of resources (up to 16TB memory)

IBM Z Hardware and z/OS



z/OS (a.k.a. *Multiple Virtual Storage* - MVS)

Where we will be spending our time

z/OS Elements, Features, and Components

z/OS Concepts – Elements, Features, Components

- z/OS is a **collection** of **elements**
  - Each element consists of a **collection of modules** (called **components**)

Examples:
IOS, RSM, GRS, Contents, Allocation, Consoles, NIP, Scheduler, SMF, BCPii, Logger

z/OS Concepts – Elements, Features, Components

- z/OS is a **collection** of **elements**
  - Each element consists of a **collection of modules** (called **components**)
  - **Base elements** are **always included** in z/OS
    - **BCP** *(Base Control Program)*
    - **DFSMSdfp** *(Data Facility Storage Management Subsystem)*
    - Deliver **essential** operating **system functions**

Other Base Elements:
Communication Server, z/OSMF, HLASM, TSO, ISPF, JES2, Metal C Runtime Library

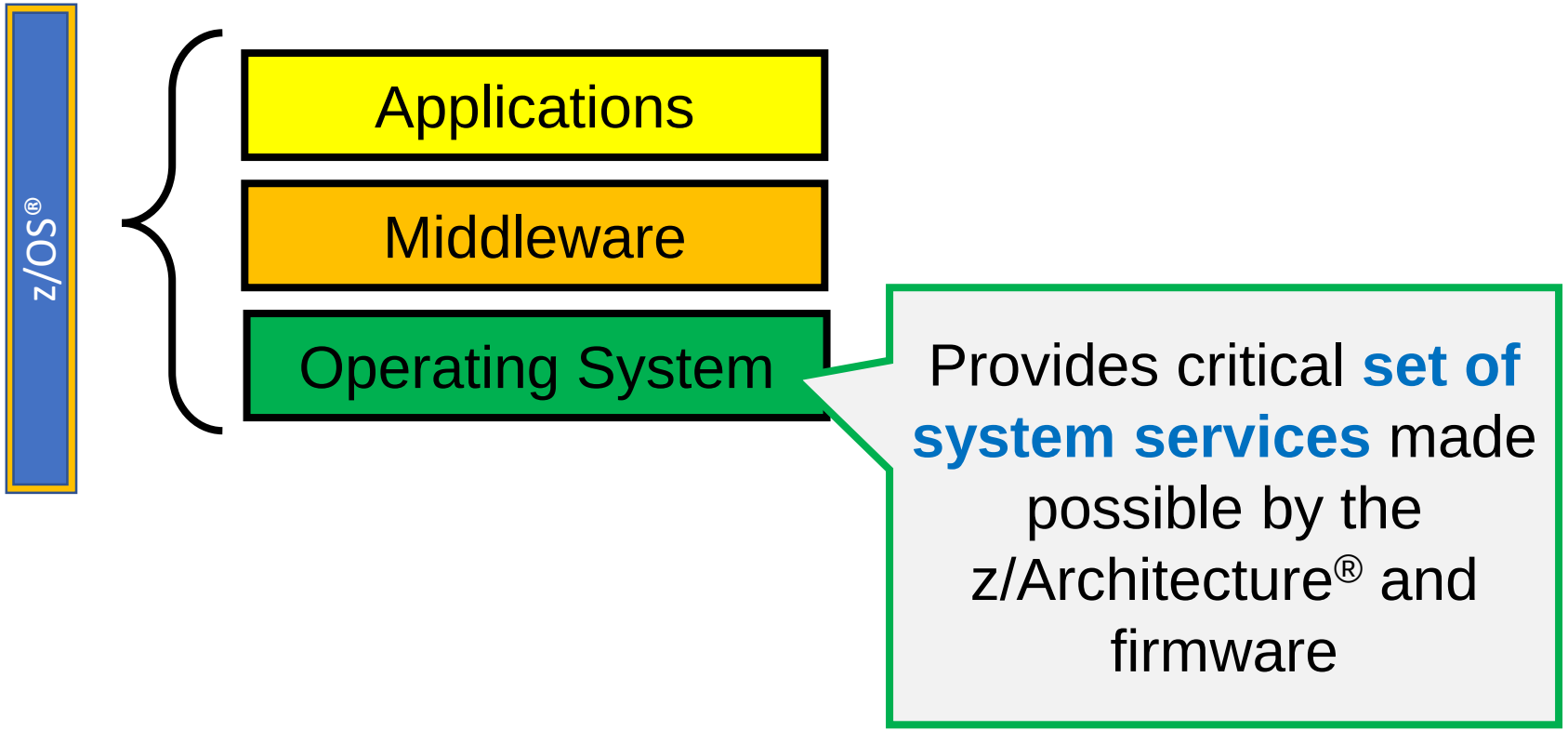
z/OS Concepts – Elements, Features, Components

- z/OS is a **collection** of **elements**
  - Each element consists of a **collection of m**
  - **Base elements** are **always included** in z/O
    - **BCP** *(Base Control Program)*
    - **DFSMSdfp** *(Data Facility Storage Managemen*
    - Deliver **essential** operating **system funct**
  - **Optional features** installed **in addition** to base elements
    - **Requested separately** from base elements
    - Can be **priced** or **free**

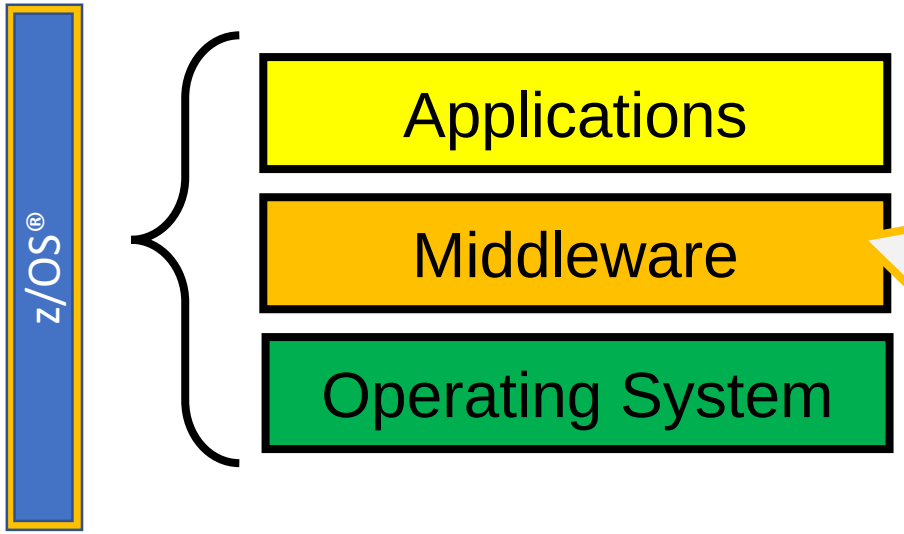
Optional Features Examples:
Security Server, SDSF, XL C/C++, RMF, JES3, DFSMShsm, DFSMSdss

Software Stack

IBM z/OS Software Stack

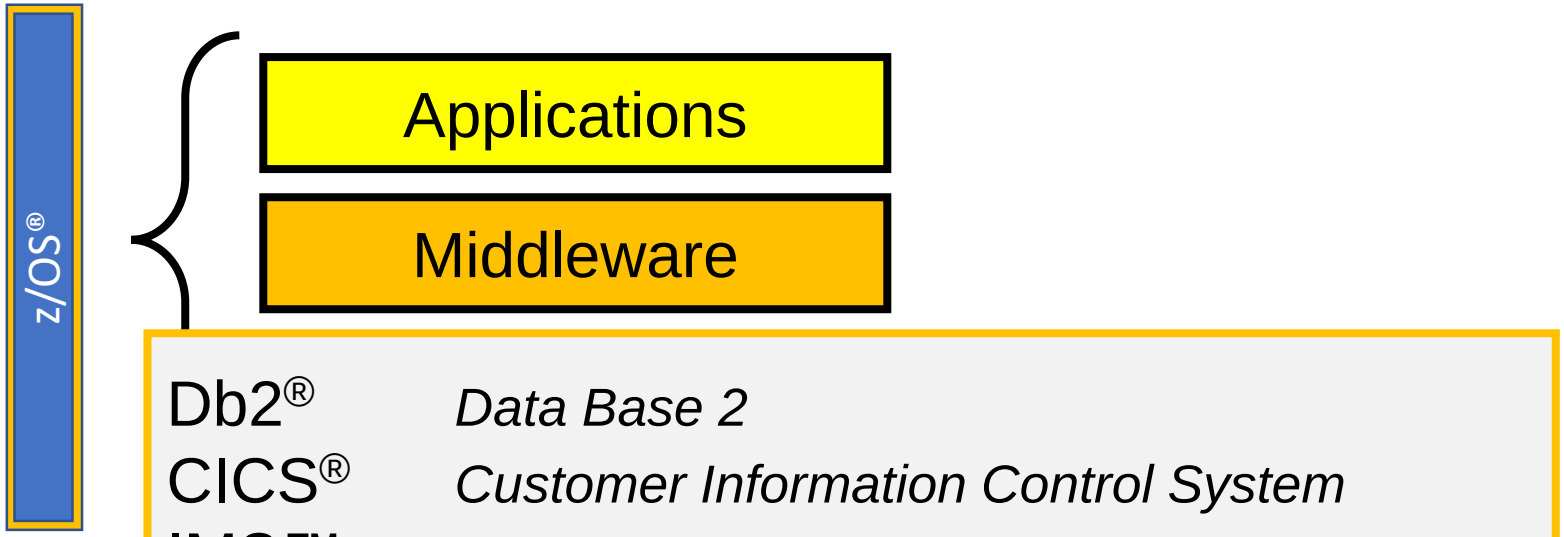


IBM z/OS Software Stack



Software that is **neither operating system code** nor an **end-user** application

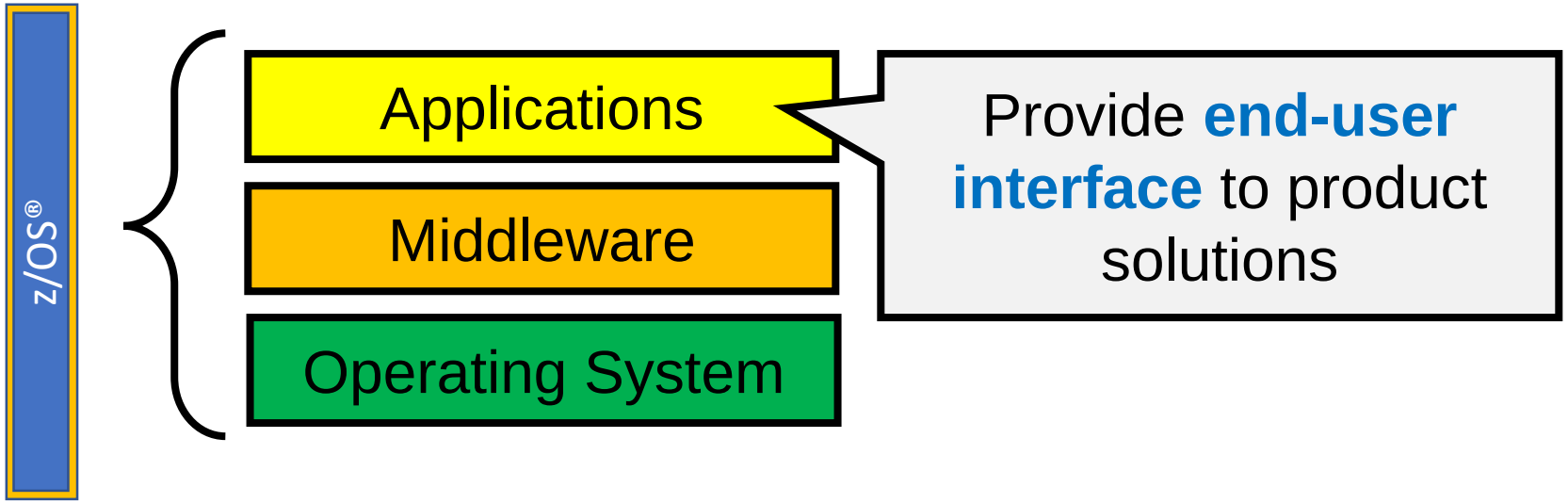
IBM z/OS Software Stack



Db2®	*Data Base 2*
CICS®	*Customer Information Control System*
IMS™	*Information Management System*
WAS	*WebSphere® Application Server*
IBM MQ	*IBM Message Queuing*
SAP®	*Systems, Applications and Products in Data Processing*
TSO/E	*Time Sharing Option/Extensions*

SAP is the registered trademark of SAP SE in Germany and in several other countries.

IBM z/OS Software Stack



ISPF	*Interactive System Productivity Facility*
SA z/OS	*IBM Tivoli® System Automation for z/OS*

Application Development Environments, Application Execution Environments and z/OS Management Environments

Application Development Env

z/OS supports many **programming languages** as well as modern **IDEs** and **environments**

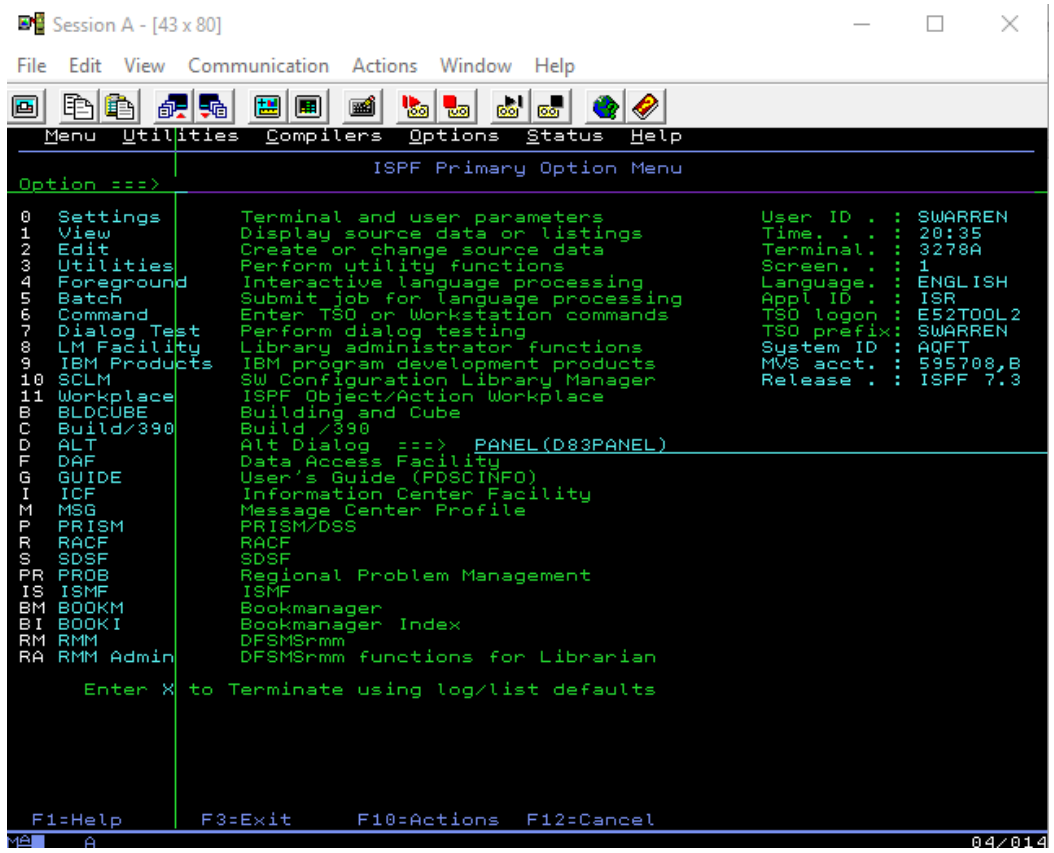
Application Development Env

z/OS supports many **programming languages** as well as modern **IDEs** and **environments**

- **Assembler**
- **COBOL**
- **C/C++**
- **PL/I**
- **Fortran**
- **REXX**
- **JAVA™**
- **Python**
- **Node.js**
- **Scala**

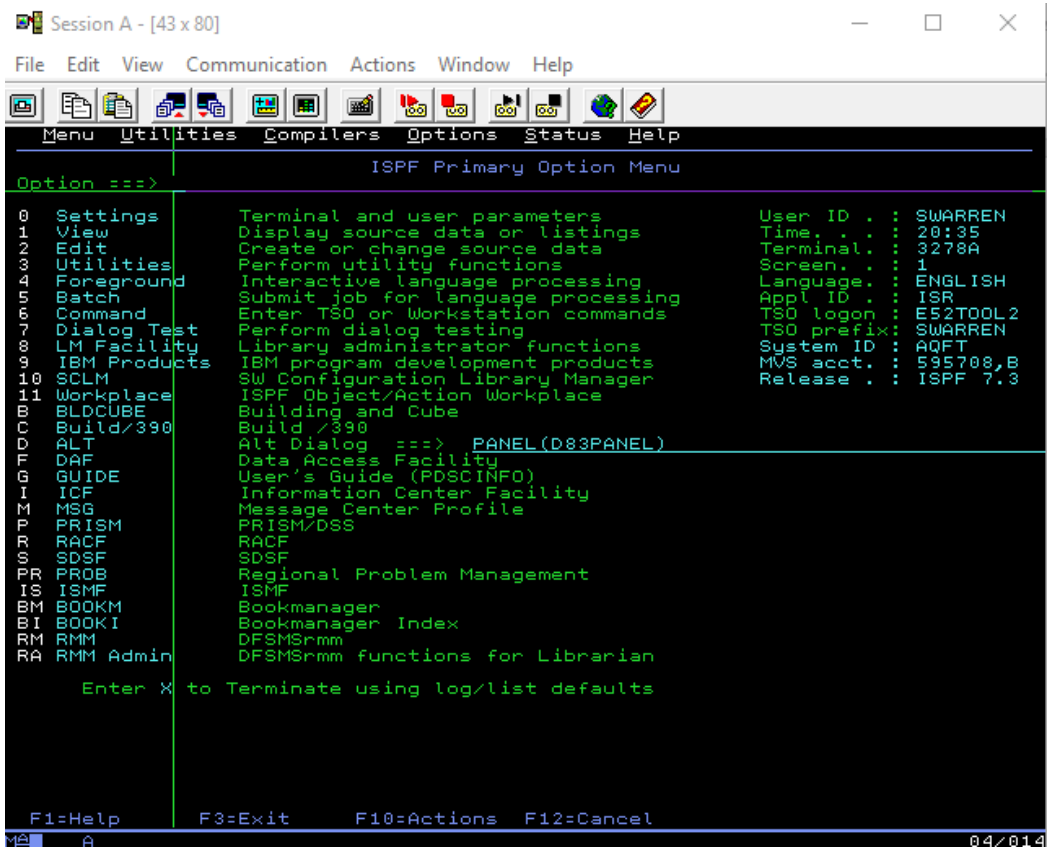
Application Development Env

Legacy interactive interface

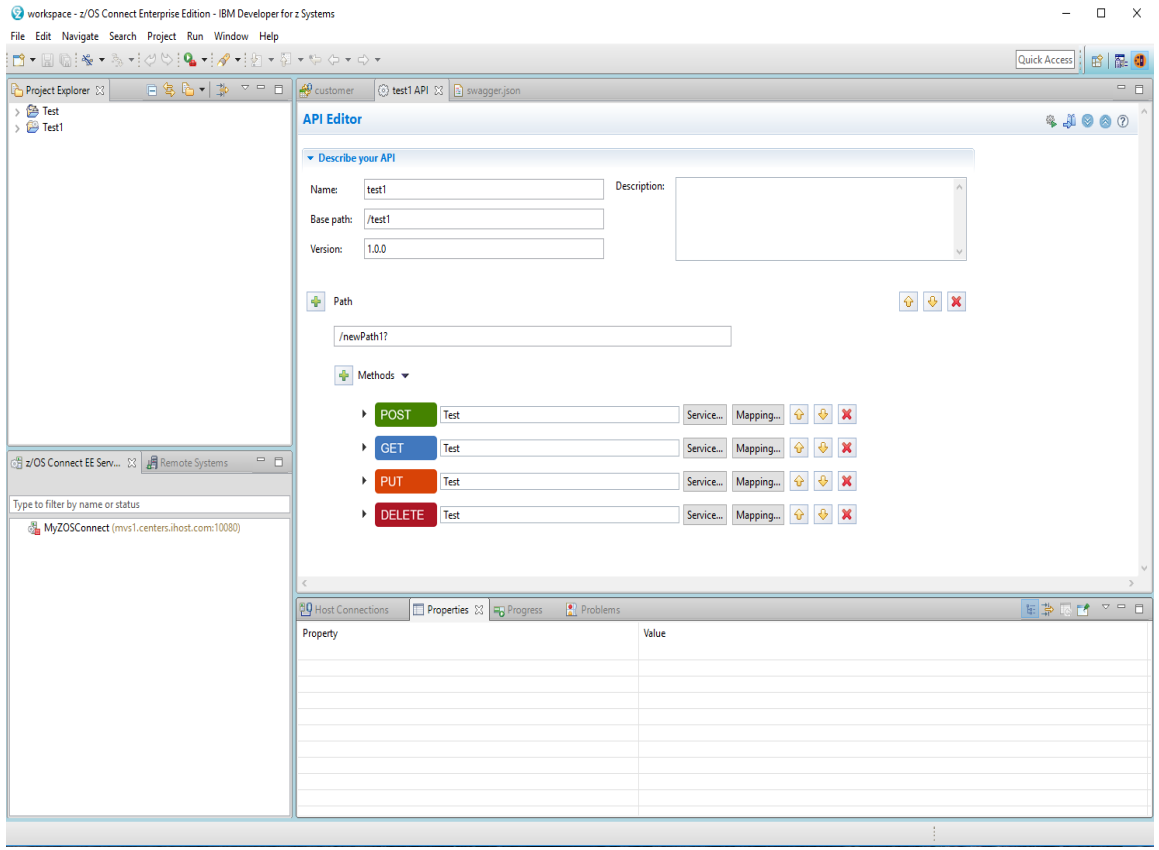


Application Development Env

Legacy interactive interface



Integrated Development Environment (IDE)

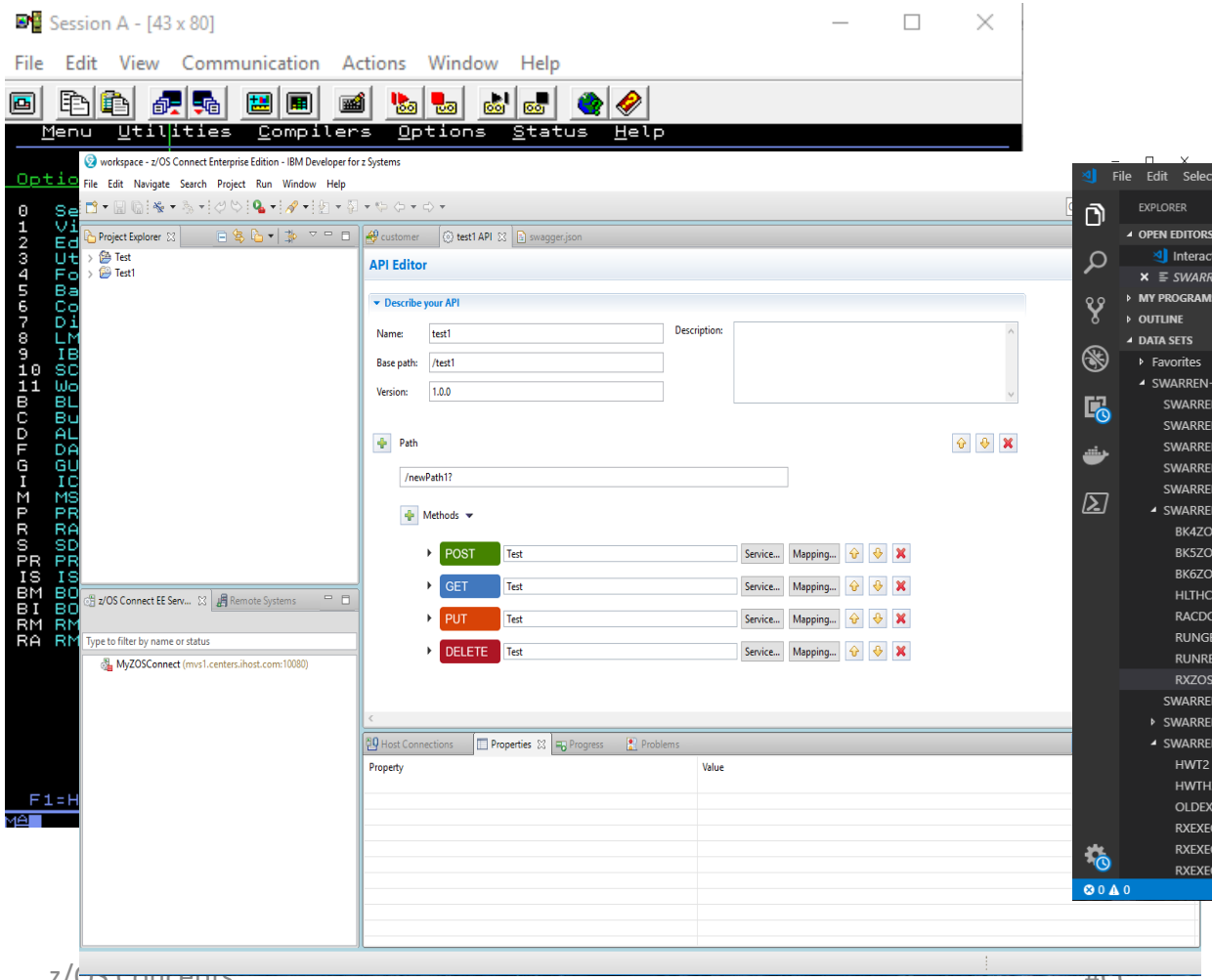


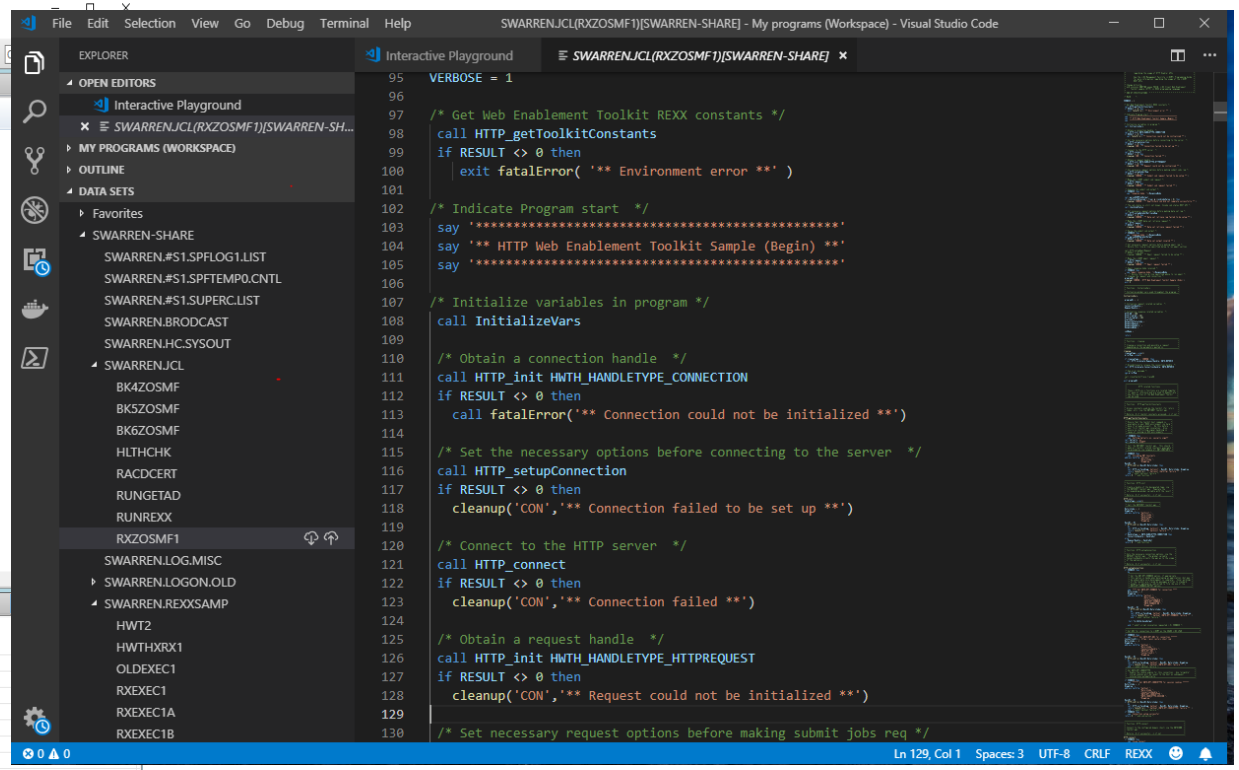
Application Development Env

Legacy interactive interface

Integrated Development Environment (IDE)

Other Modern Application Development Tools

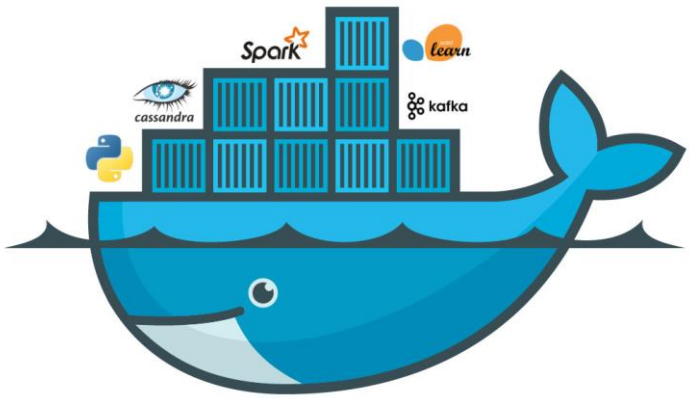




Application Development Env

Jenkins

Visual Studio Code



Travis CI

Application Execution Env

Traditional z/OS application environments, including middleware

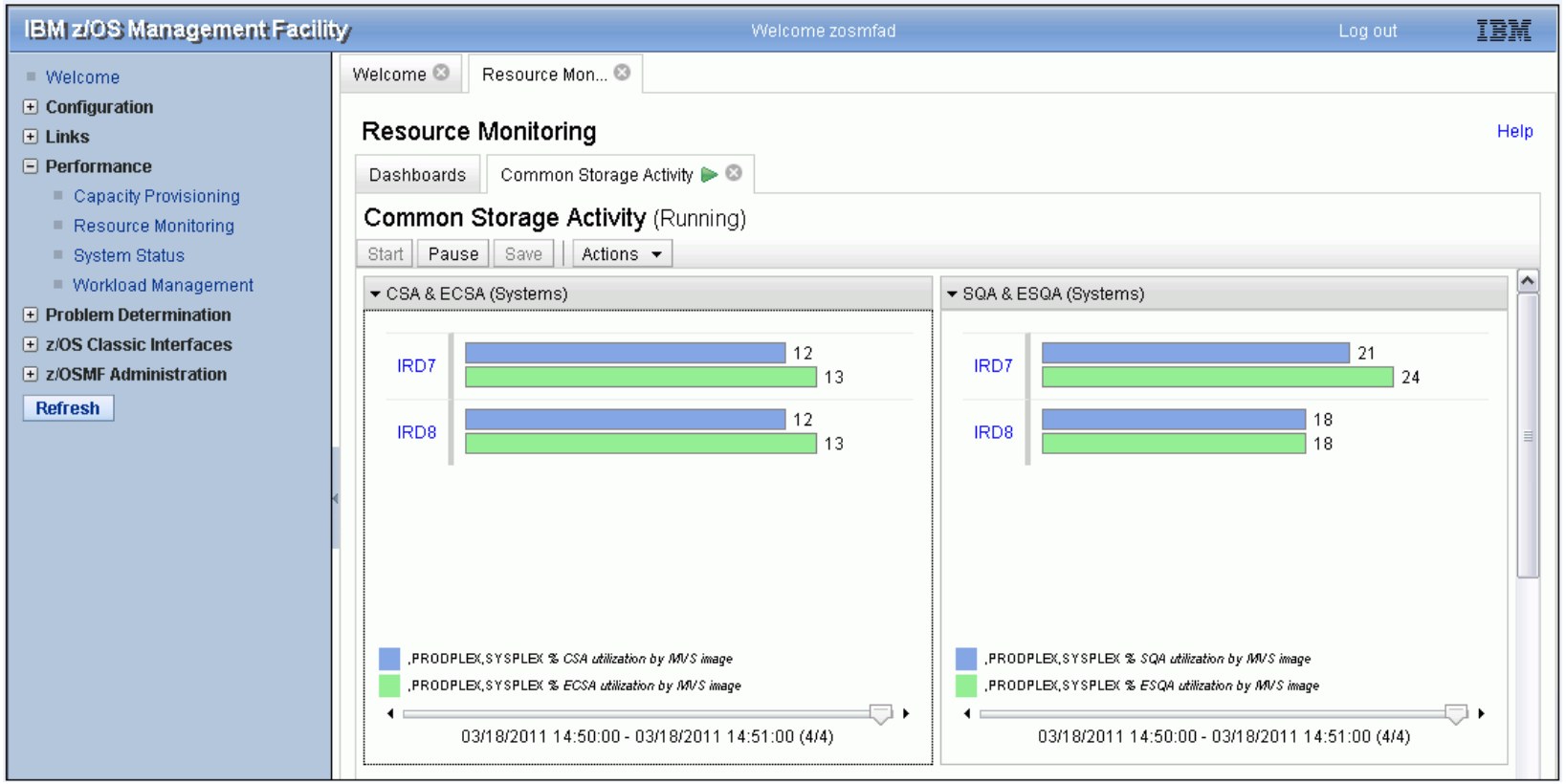
z/OS UNIX application environment

Java Virtual Machine application environment

z/OS Container Extensions application environment (new in z/OS V2R4)

Management Environments

z/OS Management Facility (**z/OSMF**) provides modern interface to help manage your z/OS systems



DASD

IBM Speak – DASD

- **DASD** – *Direct Access Storage Device*
  - Think "**Hard Drive**"

IBM Speak – DASD

- **DASD** – *Direct Access Storage Device*
  - Think "**Hard Drive**"
  - a.k.a. DASD **Volume**
  - Volume needs **a label**
    - *Volume Serial Number* (**VOLSER**)
    - 1 to 6 characters

Data Sets

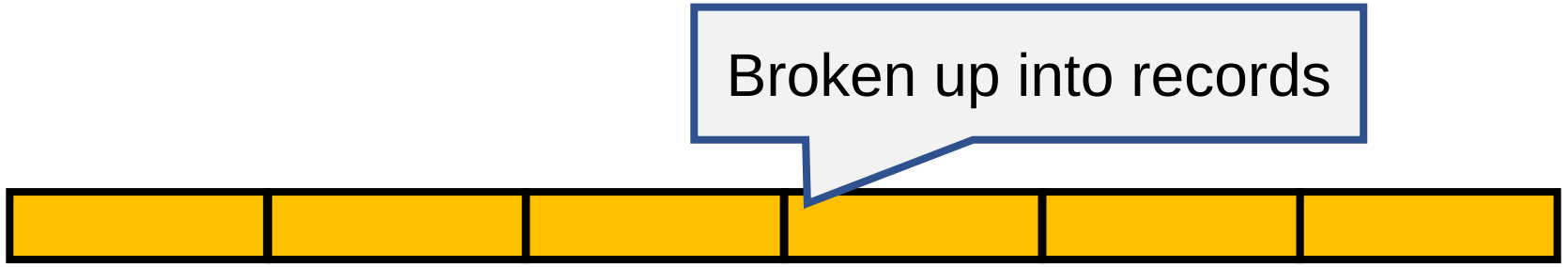
IBM Speak – Data Set

- **Data set**
  - Think "**File**"
  - Contains data in different structured formats
  - **Need to choose the format of data set and reserve space (allocate dataset) prior to using for the first time**
  - Resides on DASD, Tape
  - z/OS has types of data sets
    - Legacy Data set
    - Unix file

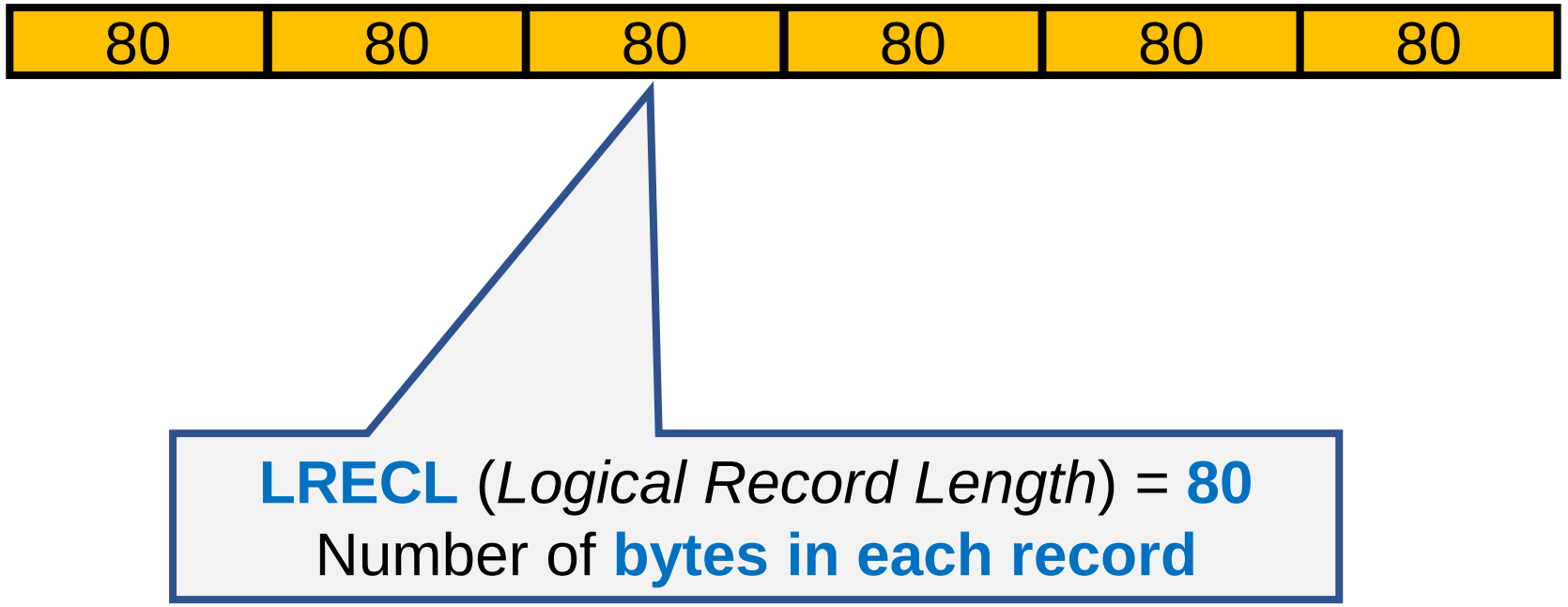
IBM prefers the spelling "data set" to "dataset"

IBM Speak – Data Set

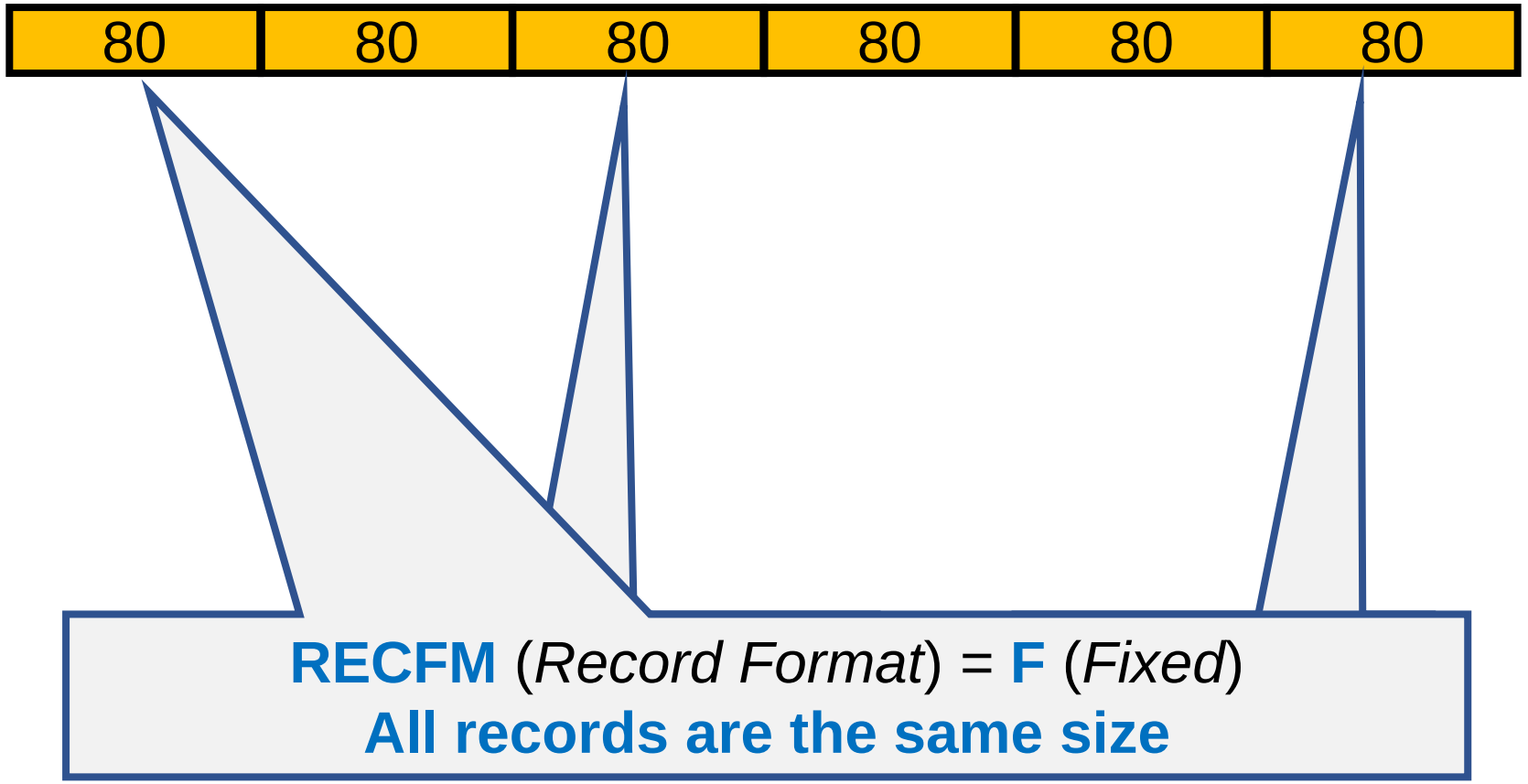
IBM Speak – Data Set



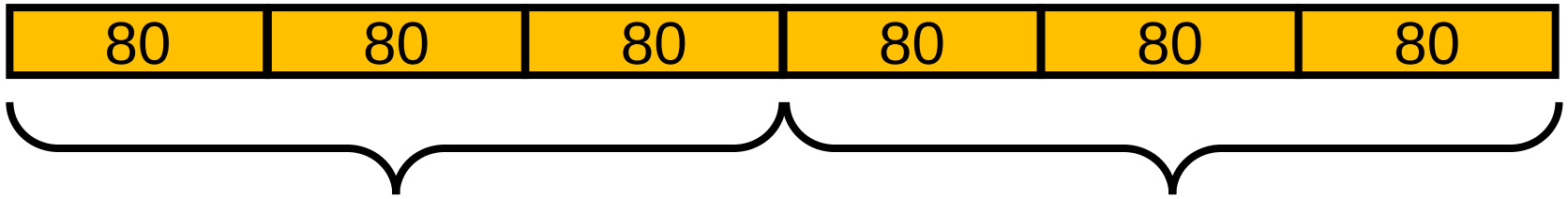
IBM Speak – Data Set



IBM Speak – Data Set



IBM Speak – Data Set

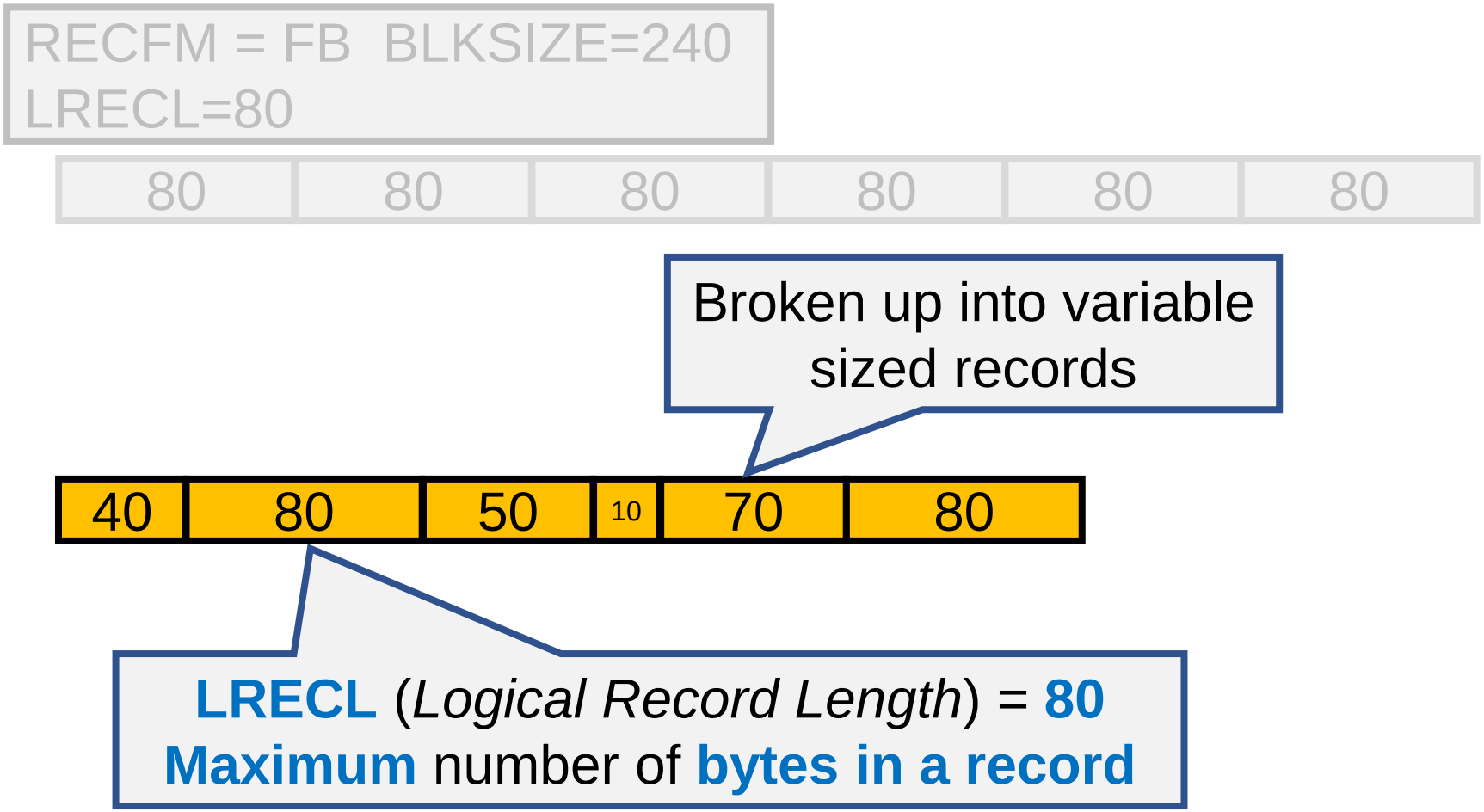


RECFM (*Record Format*) = **FB** (*Fixed Block*)
Records grouped into **blocks**

BLKSIZE (*Block Size*) = **240**

3 Records * **LRECL** = **BLKSIZE**

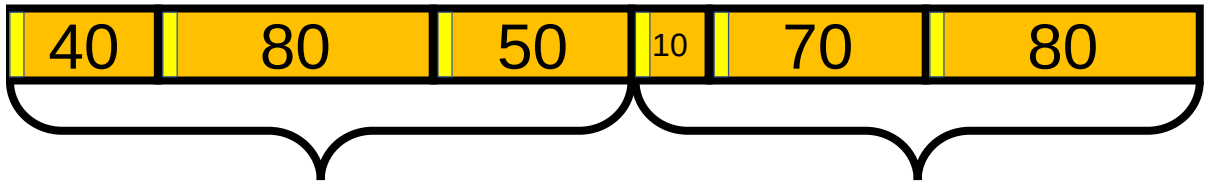
IBM Speak – Data Set



IBM Speak – Data Set

RECFM = FB BLKSIZE=240
LRECL=80

80	80	80	80	80	80



RECFM (*Record Format*) = **VB** (*Variable Block*)
Records grouped into **blocks**

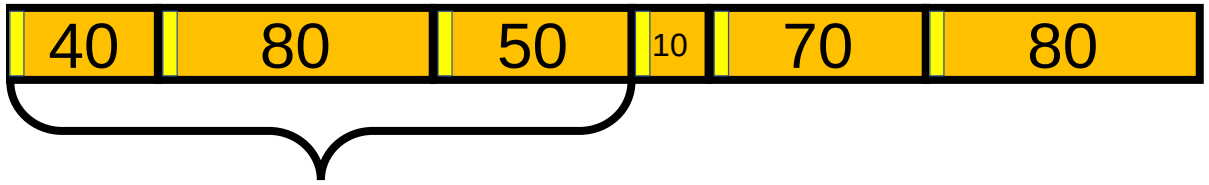
BLKSIZE (*Block Size*) = **174**

Find largest block: Total **LRECL** values + 4 byte descriptor = **BLKSIZE**

IBM Speak – Data Set

Record Format (**RECFM**) = Fixed Block (FB); Block Size (**BLKSIZE**)=240; Logical Record Length (**LRECL**)=80

Record Format (**RECFM**) = Variable Block (VB); Block Size (**BLKSIZE**)=174; Logical Record Length (**LRECL**)=80



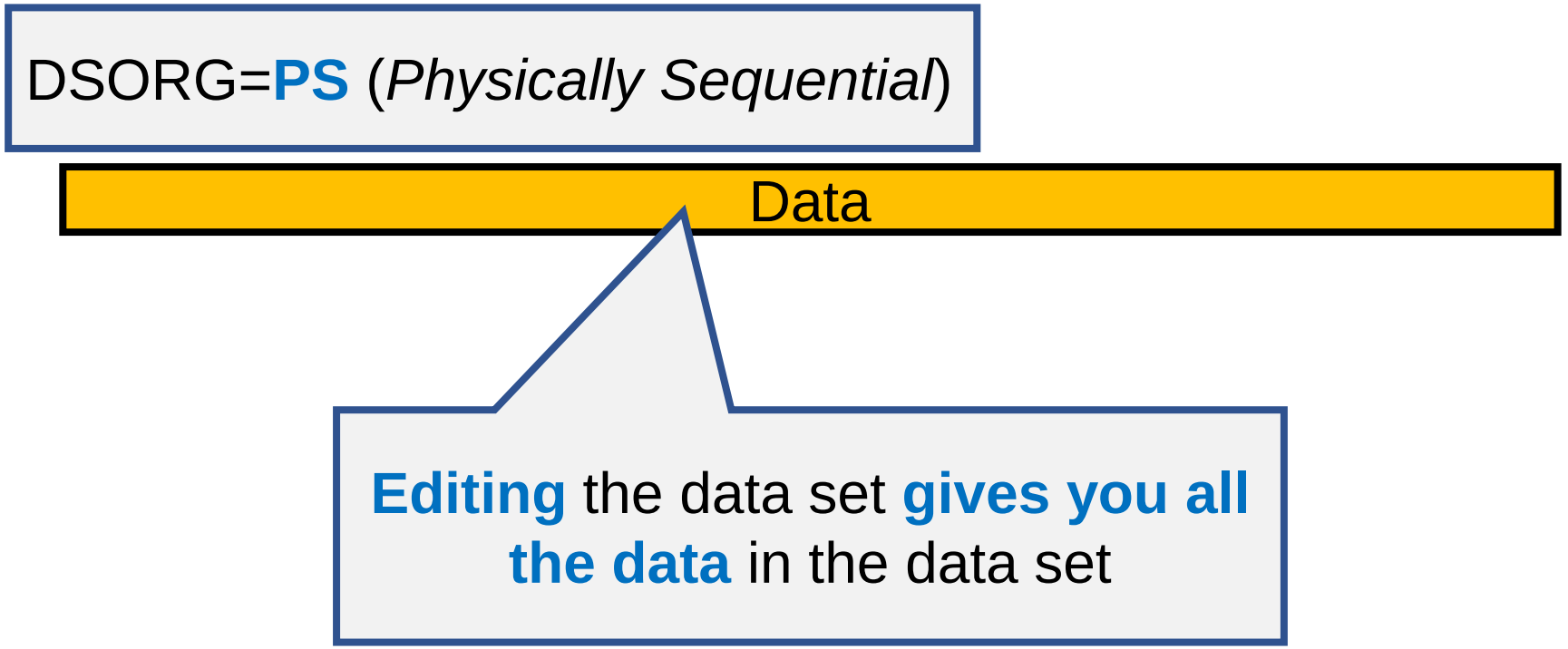
Block size =number of records to be read as a single I/O operation

Choose the appropriate settings for the type of data

IBM Speak – Data Set

One other data set attribute:
DSORG (*Data Set Organization*)

IBM Speak – Data Set



IBM Speak – Data Set

DSORG=**PS** (*Physically Sequential*)

Data

DSORG=**PO** (*Partitioned*)

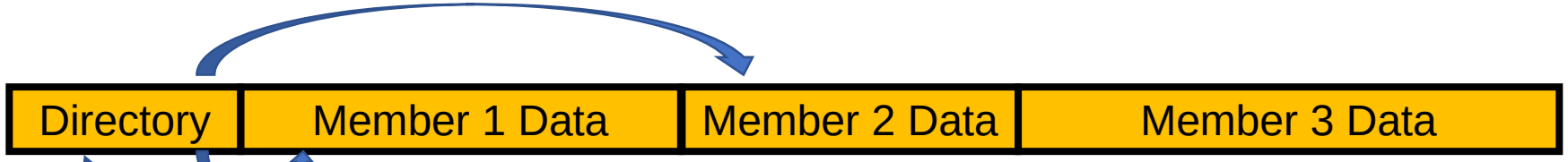
a.k.a. **PDS** (*Partitioned Data Set*)
or **PDSE** (*Partitioned Data Set Extended*)

IBM Speak – Data Set

DSORG=**PS** (*Physically Sequential*)

Data

DSORG=**PO** (*Partitioned*)



The PDS contains a **directory** to **locate** the members

The Naming of Data Sets

IBM Speak – Data Set Name

- ***Data set name*** (DSN or DSName)
  - Think "**File Name**"

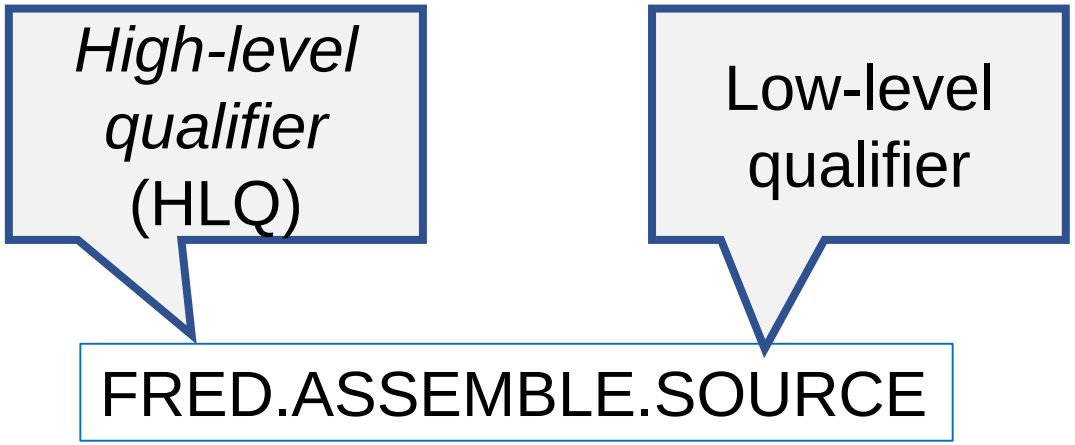
IBM Speak – Data Set Name

- ***Data set name*** (DSN or DSName)
  - Think "**File Name**"
  - 1 to 22 **segments**
    - segments **separated by a period**
    - each segment **limited to 8 characters**

FRED.ASSEMBLE.SOURCE

IBM Speak – Data Set Name

- ***Data set name*** (DSN or DSName)
  - Think "**File Name**"
  - 1 to 22 **segments**
    - segments **separated by a period**
    - each segment **limited to 8 characters**
- Up to **44 characters** (includes periods)
- **High-level qualifier** (HLQ) is typically your **userid**



Data Set Allocation

IBM Speak – Data Set Allocation

- **Data set allocation**
  - Think "**Create Space**" for a data set on disk

IBM Speak – Data Set Allocation

- **Data set allocation**
  - Think "**Create Space**" for a data set on disk
  - Need:
    - Data set **name**
    - **VOLSER** (i.e., the DASD volume) where the space should be allocated
    - Data set **attributes** (Size, RECFM, LRECL, BLKSZE, DSORG)

Check Your knowledge

Check your knowledge

- What is a VOLSER?

Check your knowledge

- What is a VOLSER? -
  - **Volume Serial Number , name of your disk drive**

Check your knowledge

- What is a VOLSER? -
  - **Volume Serial Number , name of your disk drive**
- What is the maximum length of a z/OS data set name (including periods)?

Check your knowledge

- What is a VOLSER? -
  - **Volume Serial Number , name of your disk drive**
- What is the maximum length of a z/OS data set name (including periods)?
  - **44 Characters**

TSO/E

Interacting with z/OS – TSO

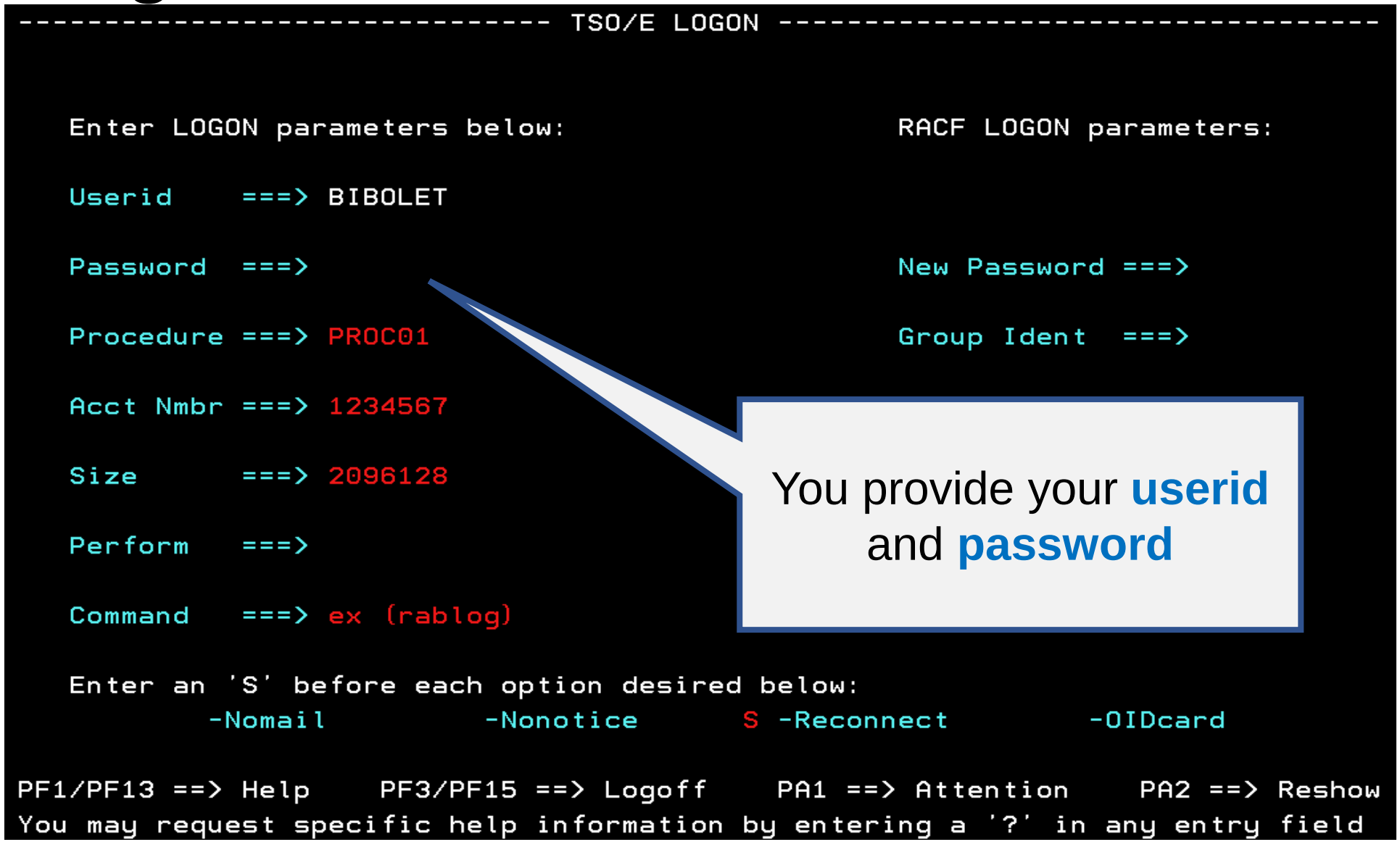
- **End users** (sometimes tens of thousands of them) use the system

Interacting with z/OS – TSO

- **End users** (sometimes tens of thousands of them) use the system

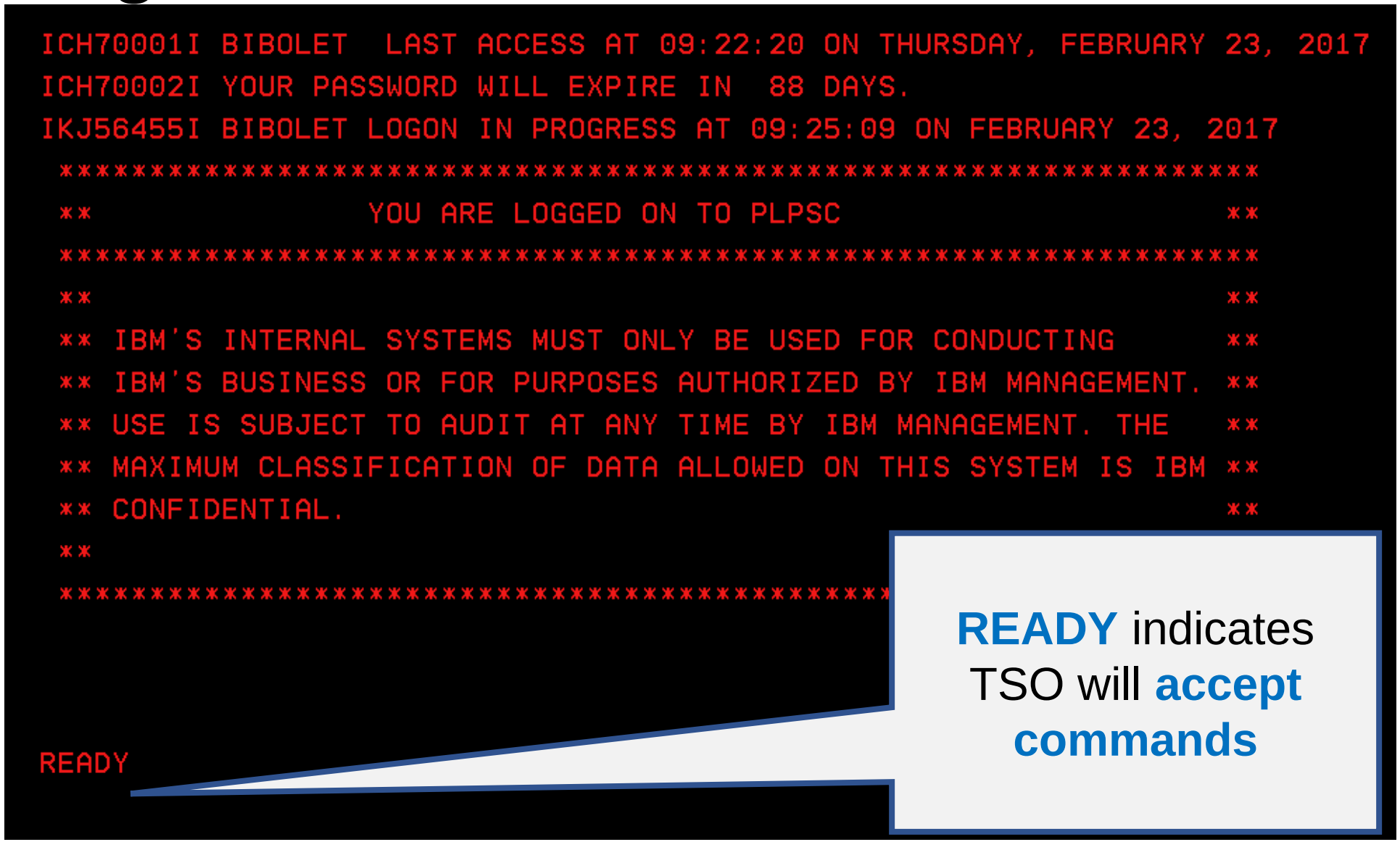
- TSO/E (*Time Sharing Option/Extensions*) allows **users to log on** and interactively **share resources**
  - Supports **limited set** of basic **commands**
  - Sometimes called using TSO in its "**native mode**"

Interacting with z/OS – TSO



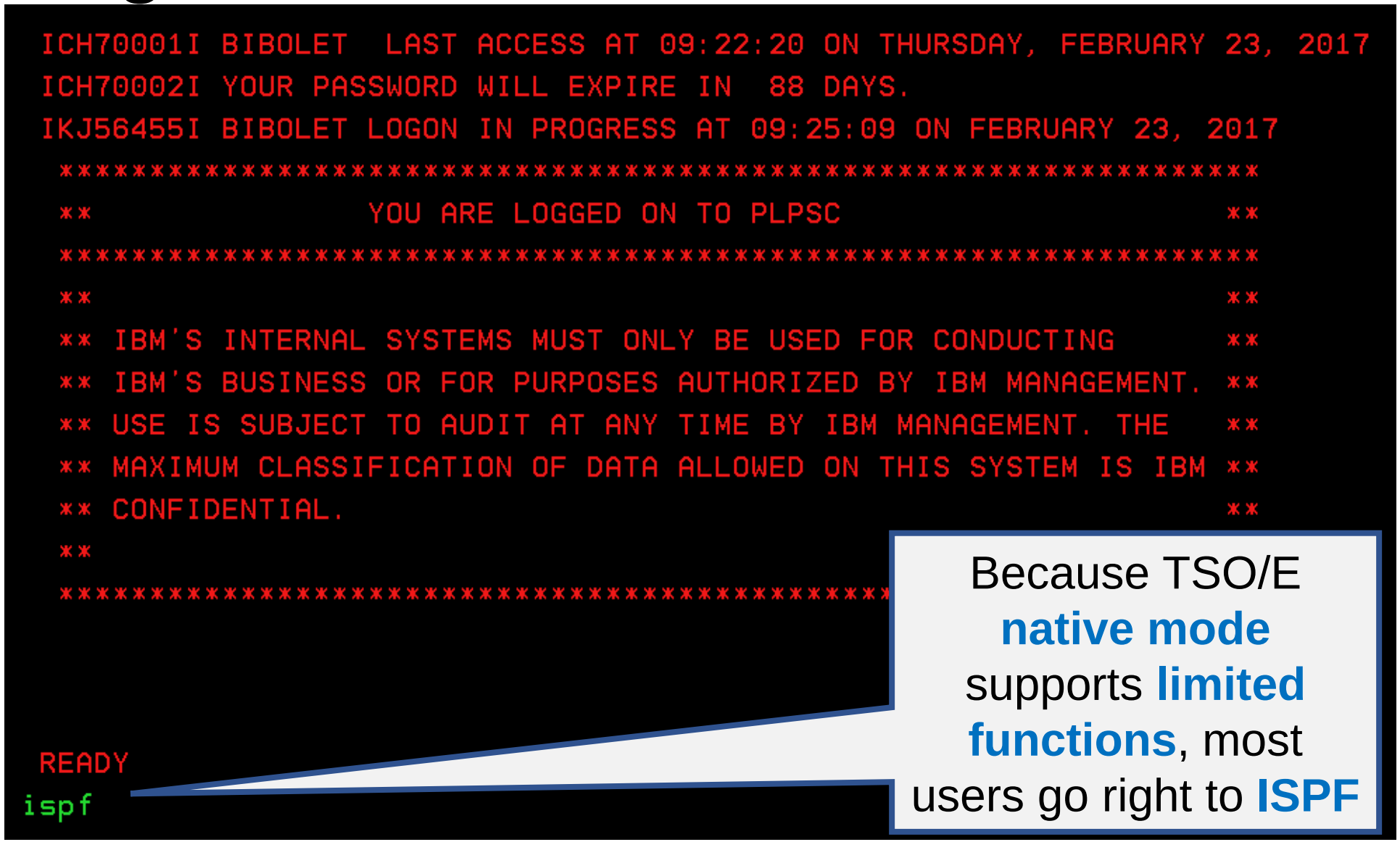
You provide your **userid** and **password**

Interacting with z/OS – TSO



ISPF

Interacting with z/OS – ISPF



Interacting with z/OS – ISPF

- **ISPF** (*Interactive System Productivity Facility*)
  - Full **panel application**
    - Panels are **hierarchical**

Interacting with z/OS – ISPF

- **ISPF** (*Interactive System Productivity Facility*)
  - Full **panel application**
    - Panels are **hierarchical**
  - **Navigated** via **keyboard**
    - **F7** and **F8 scroll up** (or backwards) and **down** (or forwards)
    - **Enter** *(not right-ctrl)* moves cursor to next input field
    - **Home** moves cursor to first input field
    - ISPF use of Function Keys – z/OS ISPF User's Guide Vol I

Table 1. Default Values

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

Link to ISPF publication

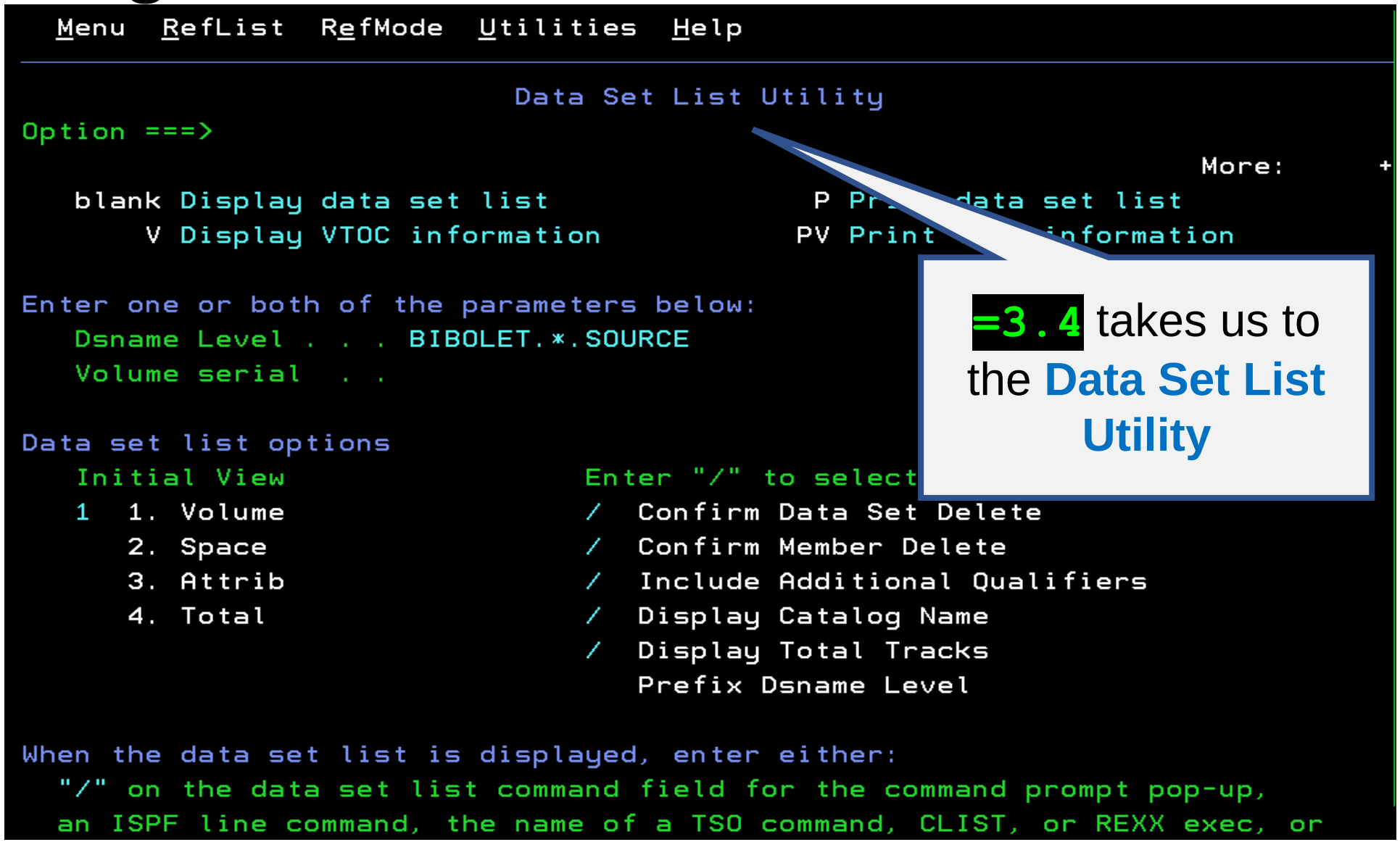
Interacting with z/OS – ISPF

- **ISPF** (*Interactive System Productivity Facility*)
  - Full **panel application**
    - Panels are **hierarchical**
  - **Navigated** via **keyboard**
    - **F7** and **F8 scroll backward** and **forward**
    - **Enter** *(not right-ctrl)* moves cursor to next input field
    - **Home** moves cursor to first input field
    - ISPF use of Function Keys – z/OS ISPF User's Guide Vol I
  - Provides text **editor** and **browser**

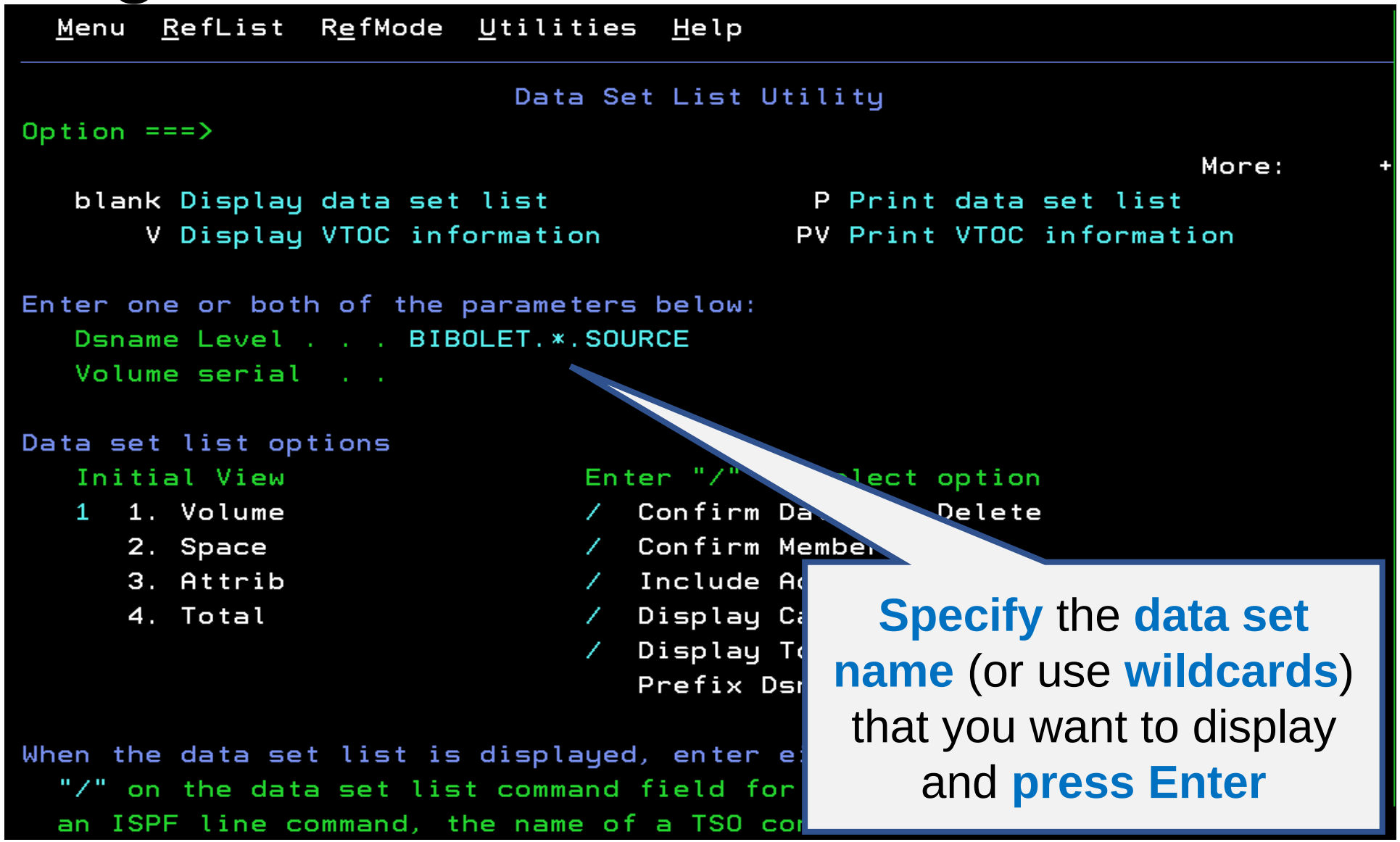
Interacting with z/OS – ISPF

- **ISPF** (*Interactive System Productivity Facility*)
  - Full **panel application**
    - Panels are **hierarchical**
  - **Navigated** via **keyboard**
    - **F7** and **F8 scroll backward** and **forward**
    - **Enter** *(not right-ctrl)* moves cursor to next input field
    - **Home** moves cursor to first input field
    - ISPF use of Function Keys
  - Provides text **editor** and **browser**
  - Data set **utilities**
    - Allocation
    - Deletion
    - Locating and Listing
    - etc.

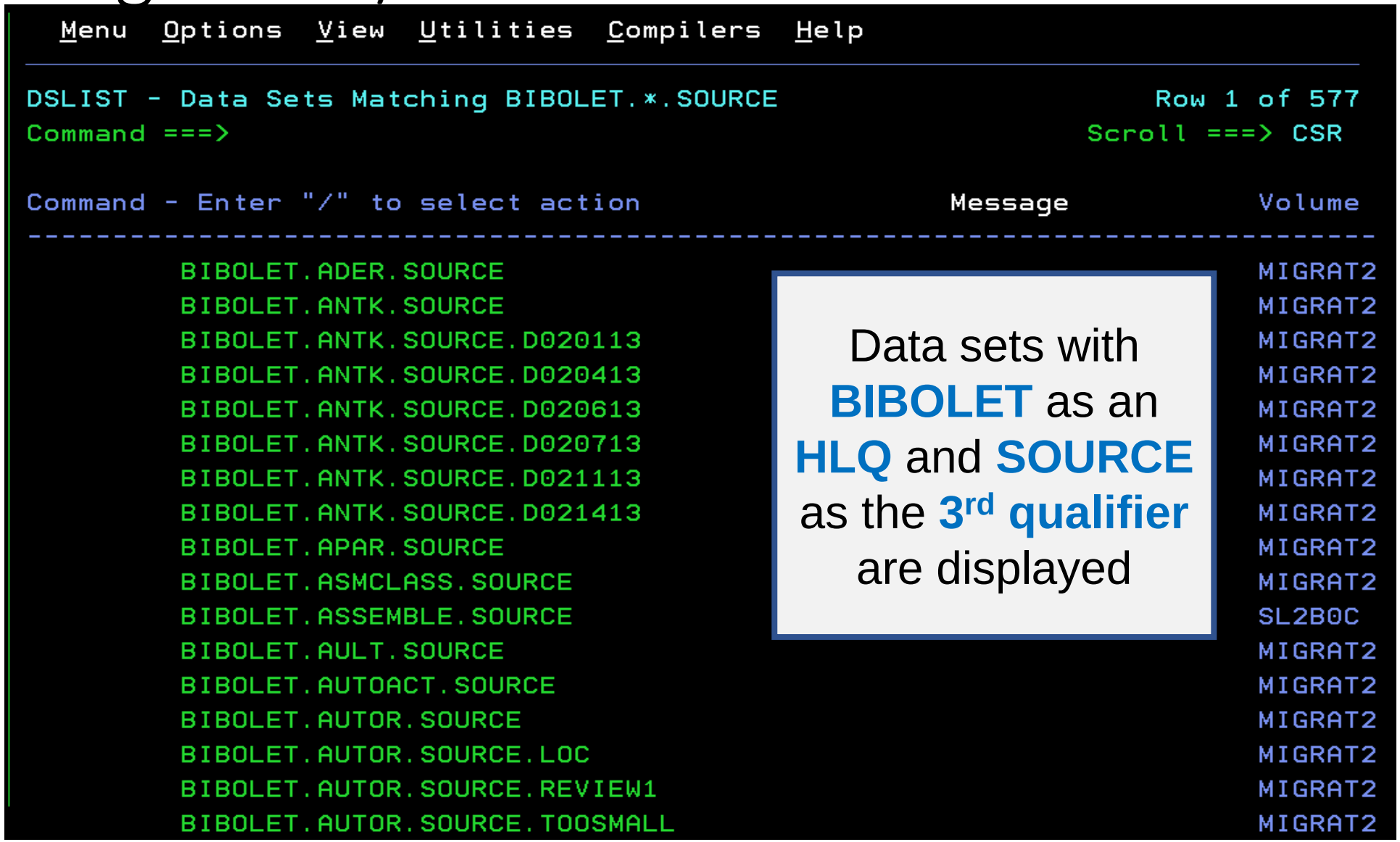
Interacting with z/OS – ISPF



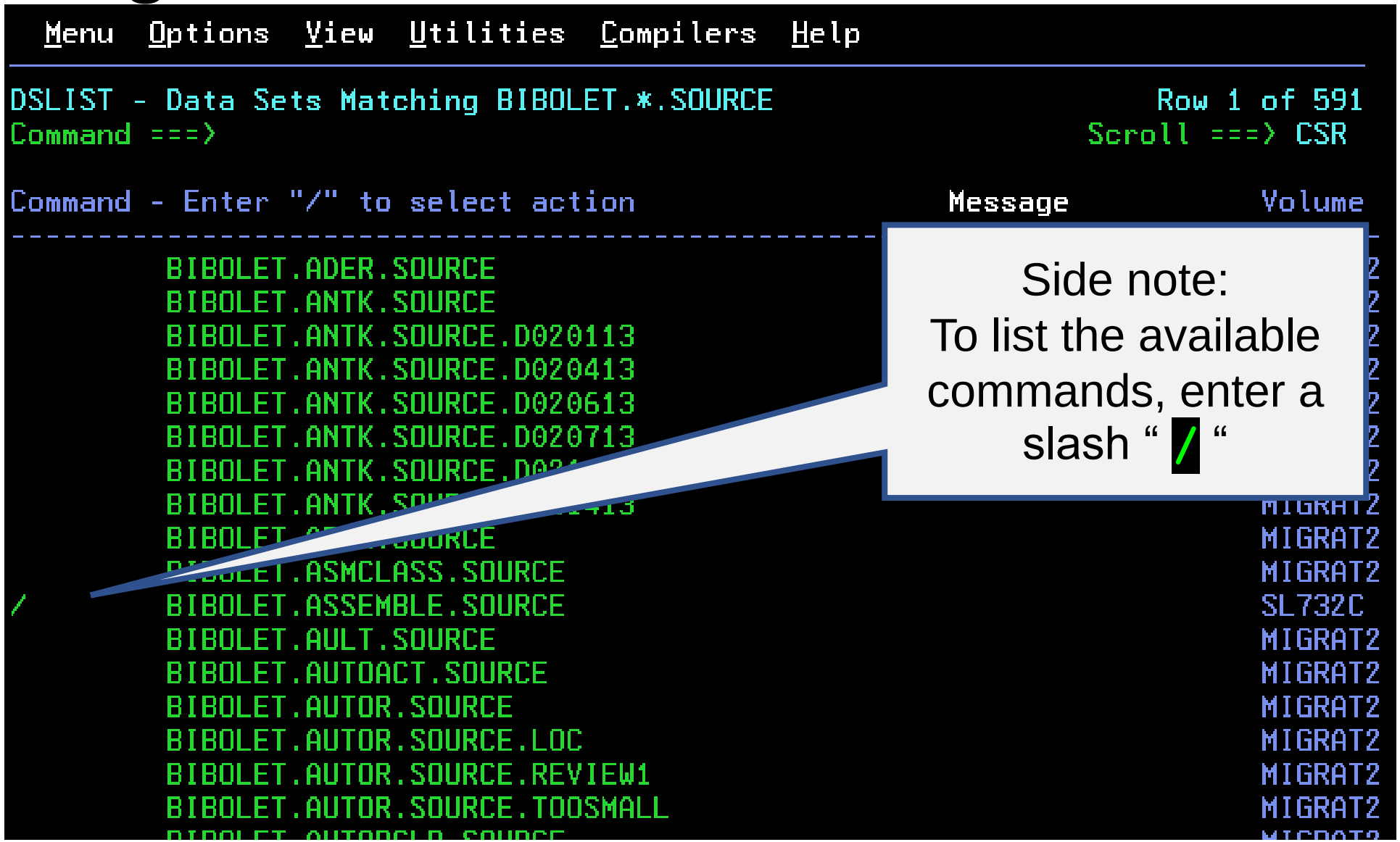
Interacting with z/OS – ISPF



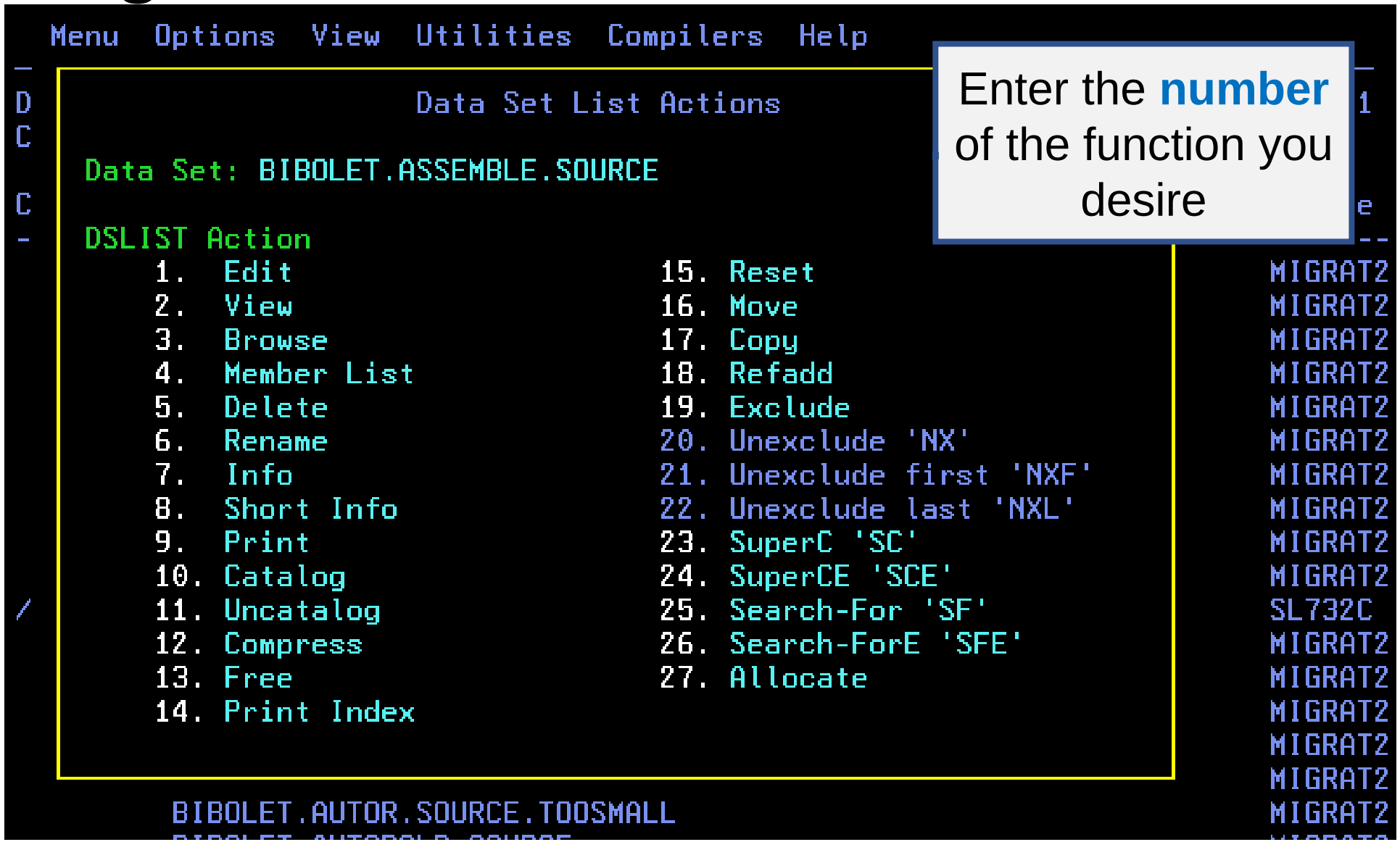
Interacting with z/OS – ISPF



Interacting with z/OS – ISPF



Interacting with z/OS – ISPF



Enter the **number** of the function you desire

Interacting with z/OS – ISPF

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
EDIT       BIBOLET.ASSEMBLE.SOURCE(HELOWRLD) - 01.00       Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ******************************** Top of Data ********************************
000001           Title 'Hello World Program For The Assembler Class'
000002 HeloWrld AMODE 31                    Addressing mode is 31-bit
000003 HeloWrld RMODE 31                    Residency mode is 31-bit
000004 HeloWrld CSECT ,
000005          SAVE  (14,12)               Save caller's registers
000006          BASR  R12,0                 Obtain addressability address
000007          USING *,R12                 Establish addressability
000008
000009 ***********************************************************
000010 *        Chain our savearea to the caller's saveare
000011 ***********************************************************
000012
000013          LA    R2,SaveArea           Get address of ou
000014          ST    R2,8(,R13)            Make caller SA point to our SA
000015          ST    R13,SaveArea+4        Make our SA point to our          X
000016                                      caller's SA
000017          LR    R13,R2                Setup SA to be used by code       X
000018                                      that we call
000019
```

You **write** your **program**!

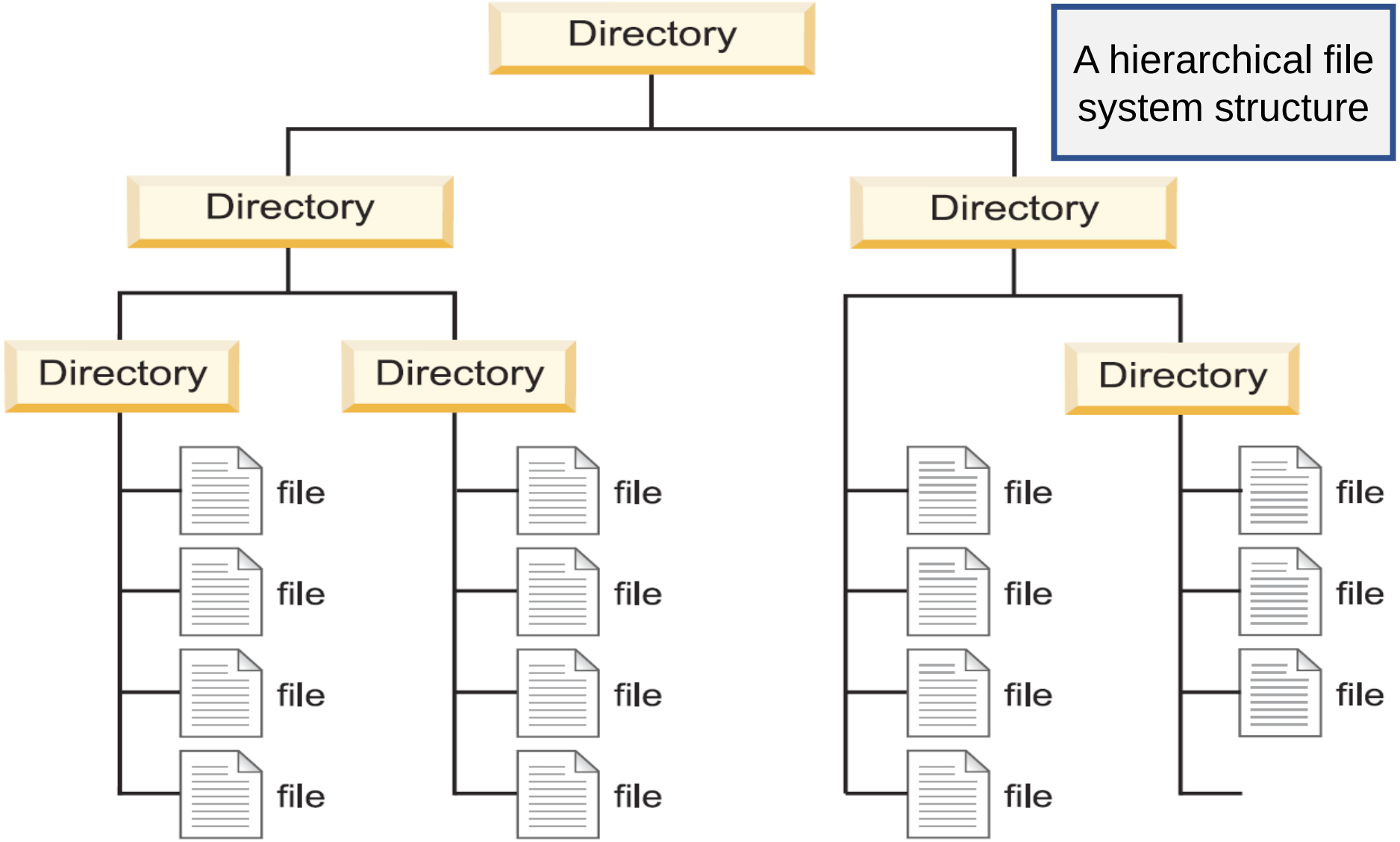
z/OS UNIX

Interacting with z/OS – UNIX

- z/OS UNIX a certified UNIX operating system by the Open Software Foundation
- **z/OS UNIX®** provides **another interactive** way to access z/OS
- Before we examine some UNIX functions, we **need to understand** the z/OS UNIX file system
  - In particular, **the zFS** (*zSeries File System*)

UNIX is a registered trademark of The Open Group

Interacting with z/OS – UNIX zFS



A hierarchical file system structure

Interacting with z/OS – UNIX zFS

- **Path name** identifies a file
  - **Consists** of **directory** names and a **file** name

Interacting with z/OS – UNIX zFS

- **Path name** identifies a file
  - **Consists** of **directory** names and a **file** name
  - Up to **1023** characters

Interacting with z/OS – UNIX zFS

- **Path name** identifies a file
  - **Consists** of **directory** names and a **file** name
  - Up to **1023** characters
  - Directories and file name **separated by** a forward-slash (**/**)

```
/dir1/dir2/dir3/MyFile
```

Interacting with z/OS – UNIX zFS

- **Path name** identifies a file
  - **Consists** of **directory** names and a **file** name
  - Up to **1023** characters
  - Directories and file name **separated by** a forward-slash (**/**)

    `/dir1/dir2/dir3/MyFile`
  - Names ***are*** **case sensitive**

Interacting with z/OS – UNIX zFS

- **Path name** identifies a file
  - **Consists** of **directory** names and a **file** name
  - Up to **1023** characters
  - Directories and file name **separated by** a forward-slash (**/**)

    `/dir1/dir2/dir3/MyFile`
  - Names ***are*** **case sensitive**
  - All files are **sequential files**

Interacting with z/OS – UNIX

- z/OS UNIX provides **three main environments**
  - **ISHELL**
    - **ISPF panel interface** to z/OS UNIX System Services
    - Good for **users familiar with** TSO and ISPF

Interacting with z/OS – UNIX

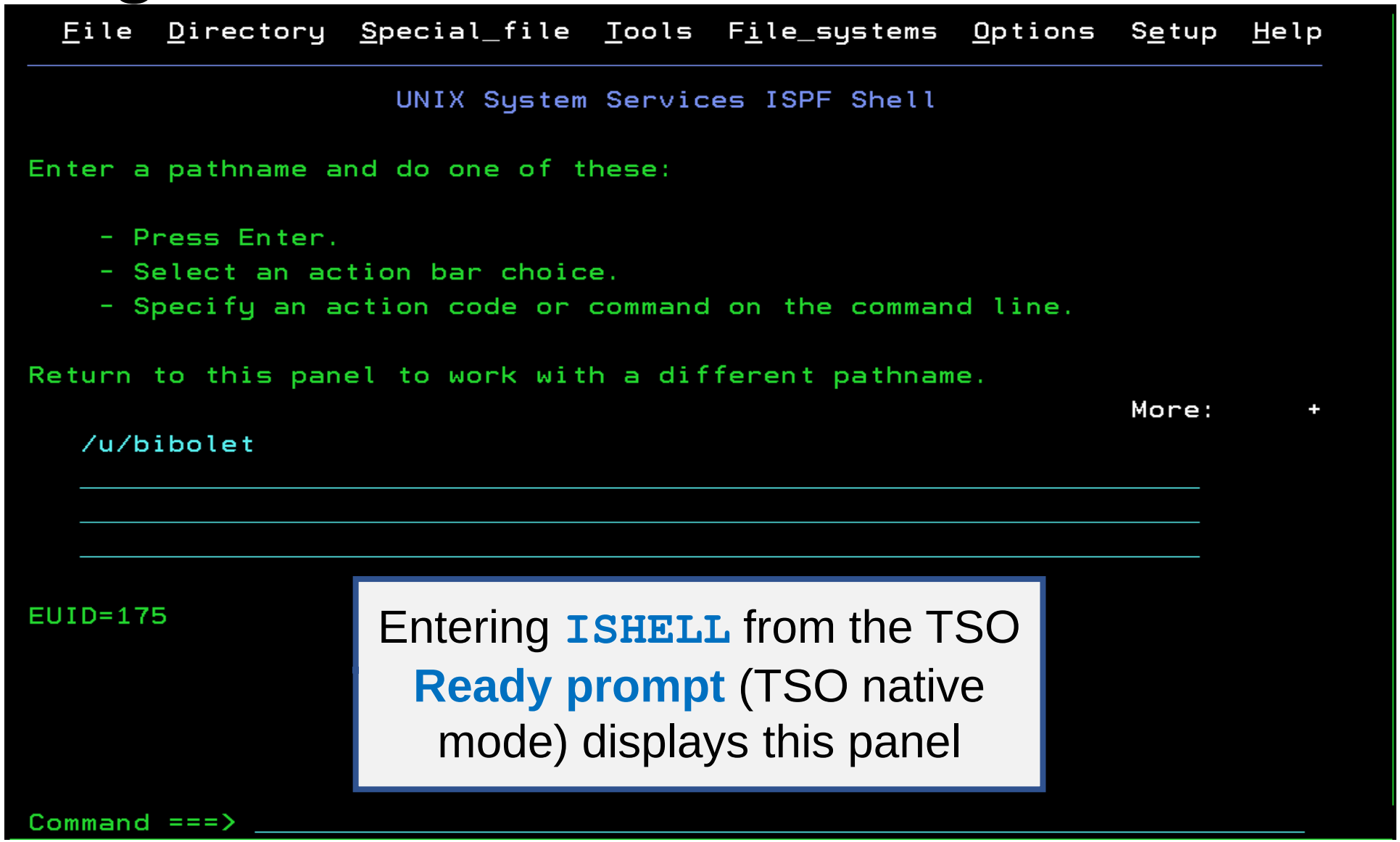
- z/OS UNIX provides **three main environments**
  - **ISHELL**
    - **ISPF panel interface** to z/OS UNIX System Services
    - Good for **users familiar with** TSO and ISPF
  - **OMVS**
    - The z/OS UNIX **shell**
    - Users of **current UNIX systems** find the z/OS UNIX **shell environment familiar**

Interacting with z/OS – UNIX

- z/OS UNIX provides **three main environments**
  - **ISHELL**
    - **ISPF panel interface** to z/OS UNIX System Services
    - Good for **users familiar with** TSO and ISPF
  - **OMVS**
    - The z/OS UNIX **shell**
    - Users of **current UNIX systems** find the z/OS UNIX **shell environment familiar**
  - **Remote UNIX interfaces**
    - z/OS UNIX can be accessed thru standard UNIX interfaces remotely
      - **Telnet, SSH, NFS**

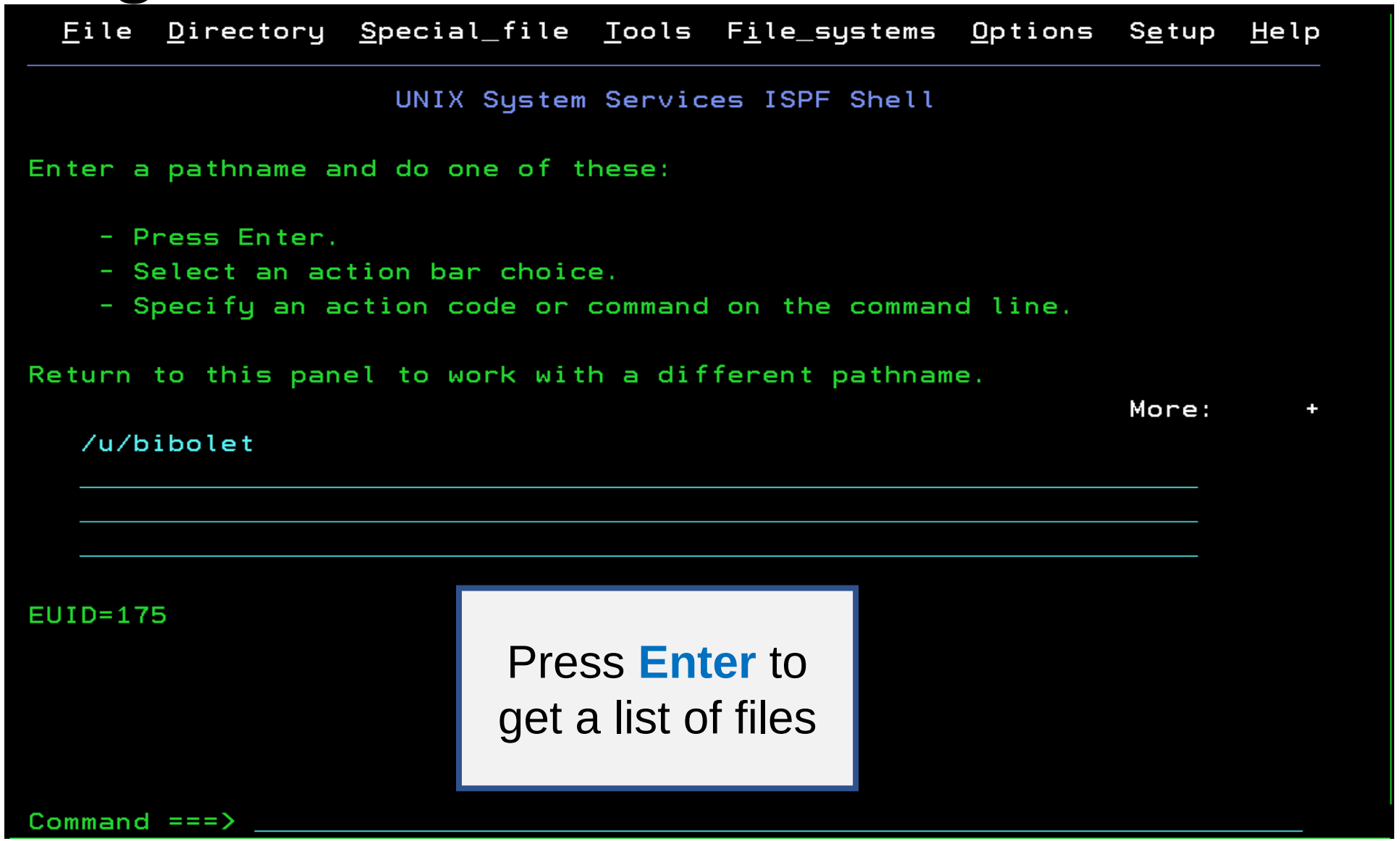
ISHELL

Interacting with z/OS – ISHELL



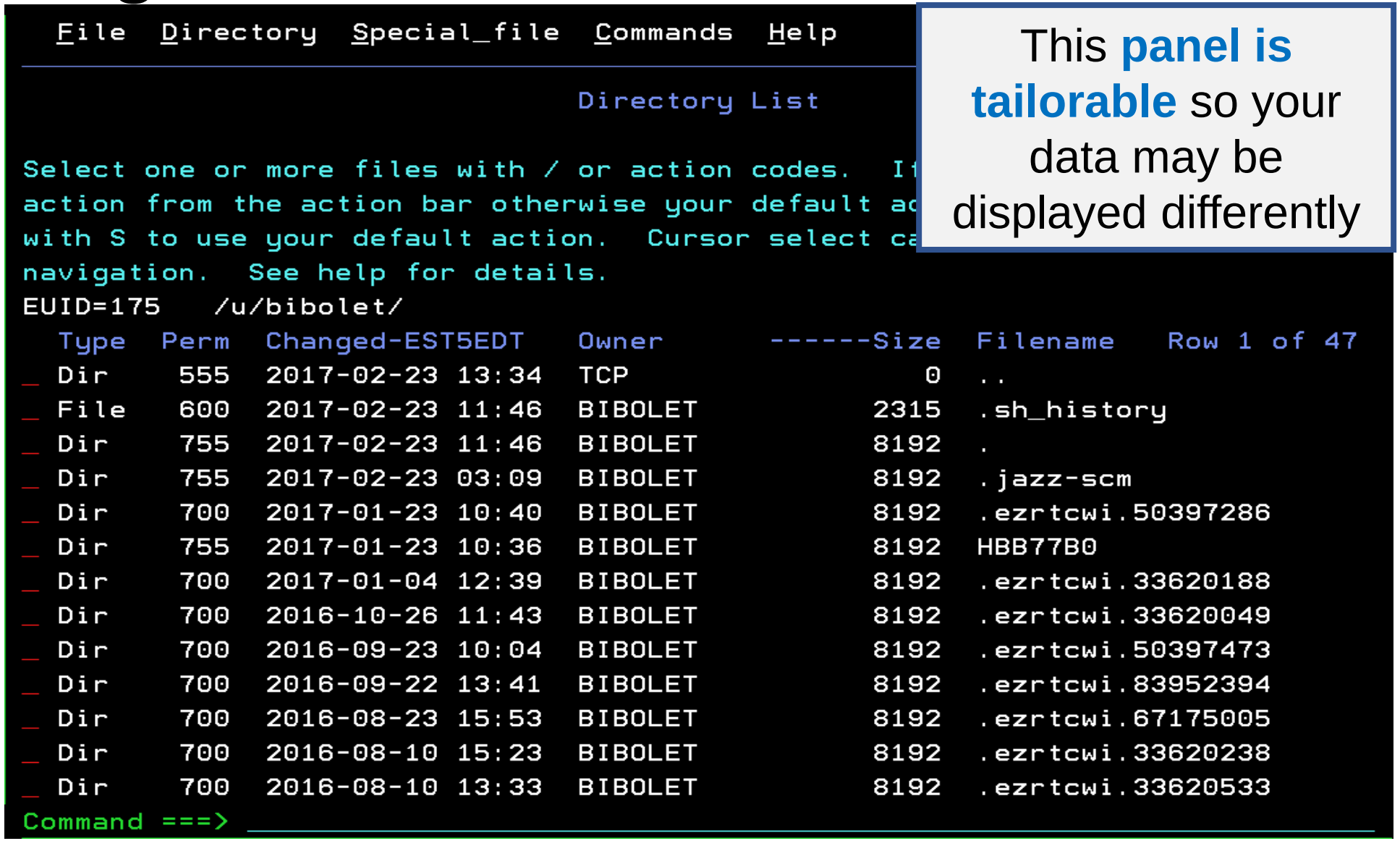
Entering **ISHELL** from the TSO **Ready prompt** (TSO native mode) displays this panel

Interacting with z/OS – ISHELL



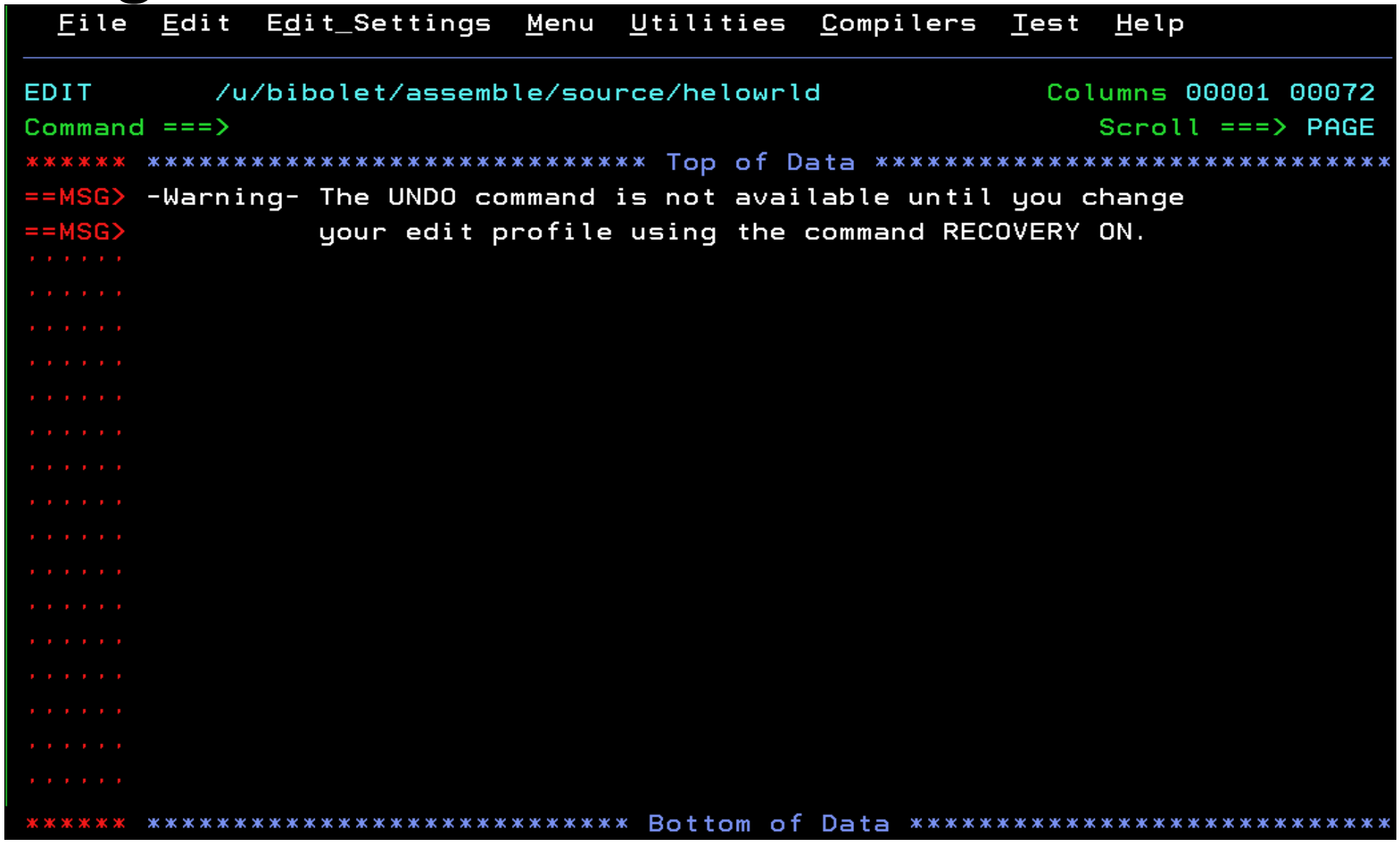
Press **Enter** to get a list of files

Interacting with z/OS – ISHELL



This **panel is tailorable** so your data may be displayed differently

Interacting with z/OS – ISHELL

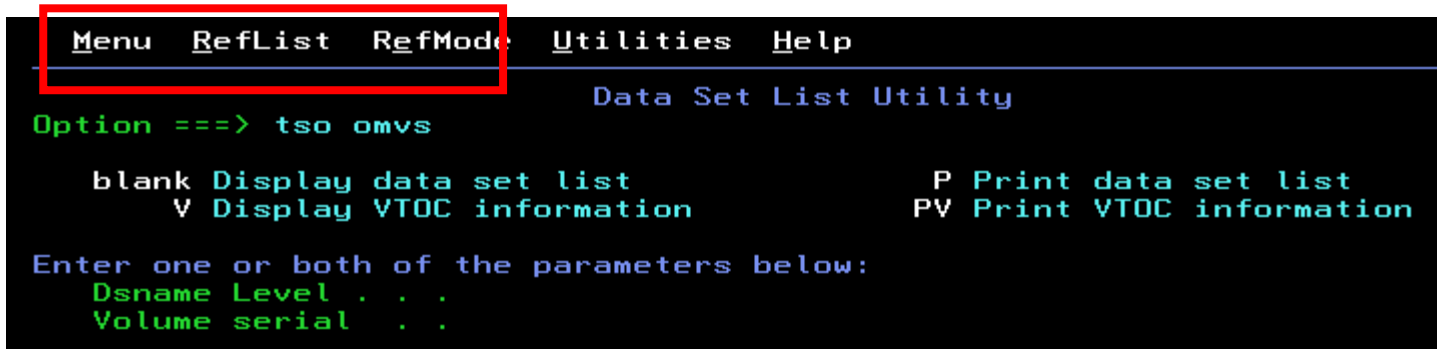


OMVS

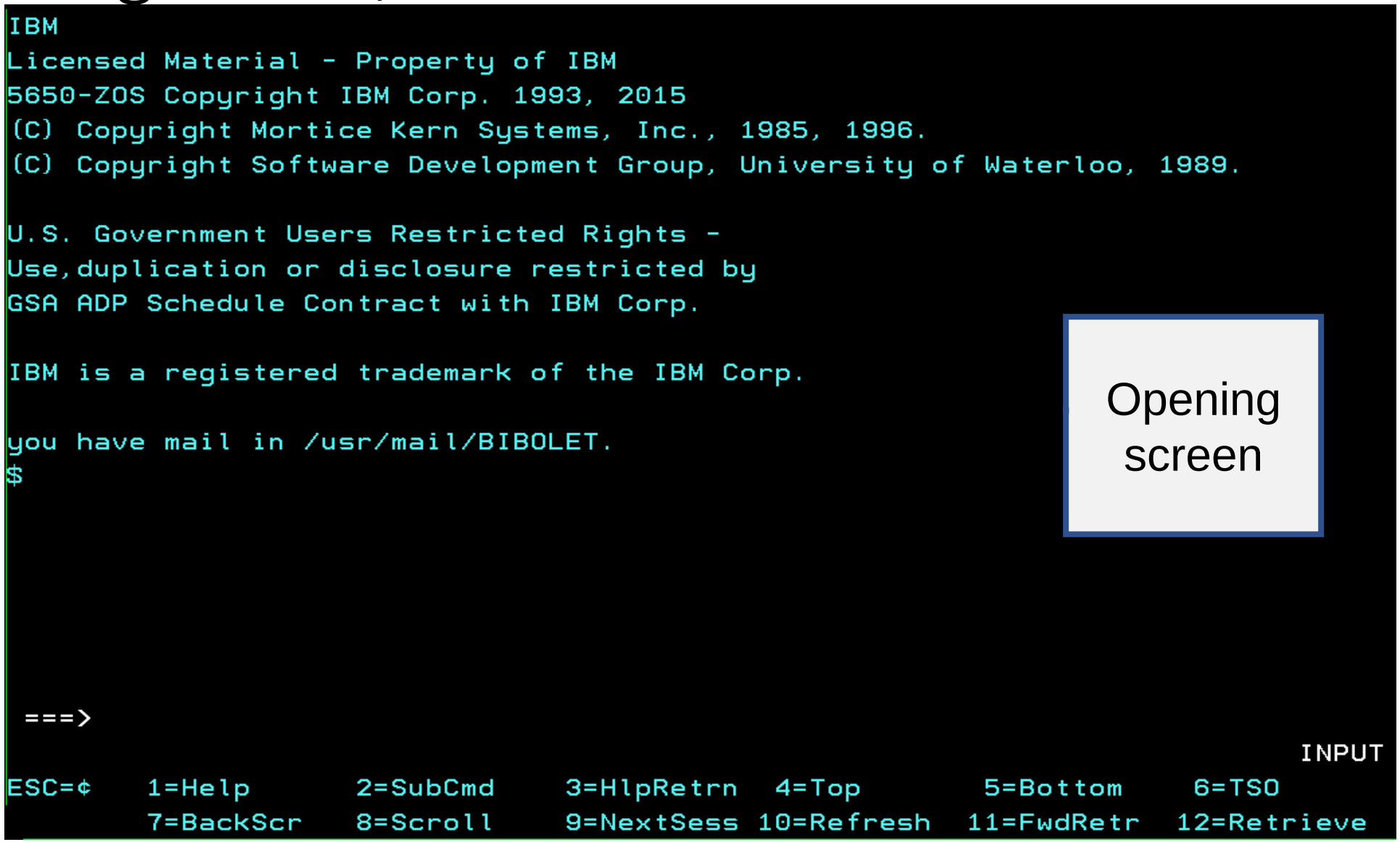
Interacting with z/OS – OMVS

- **OMVS** is the command to invoke the z/OS UNIX shell

- **Specify:**
  - At the TSO **READY prompt**
  - From the ISPF **Command Shell** (option 6)
  - From an ISPF **panel input field**

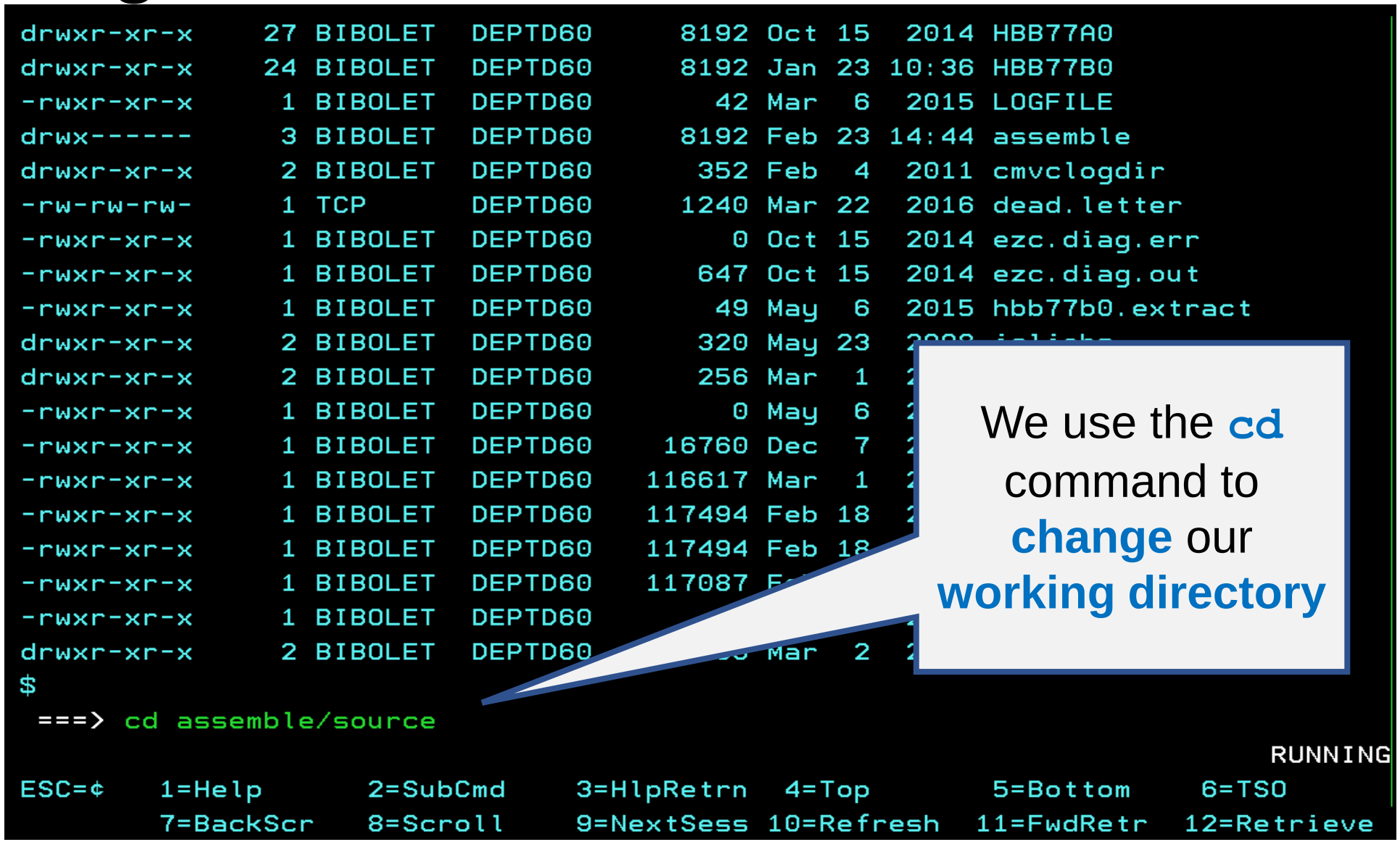


Interacting with z/OS – OMVS



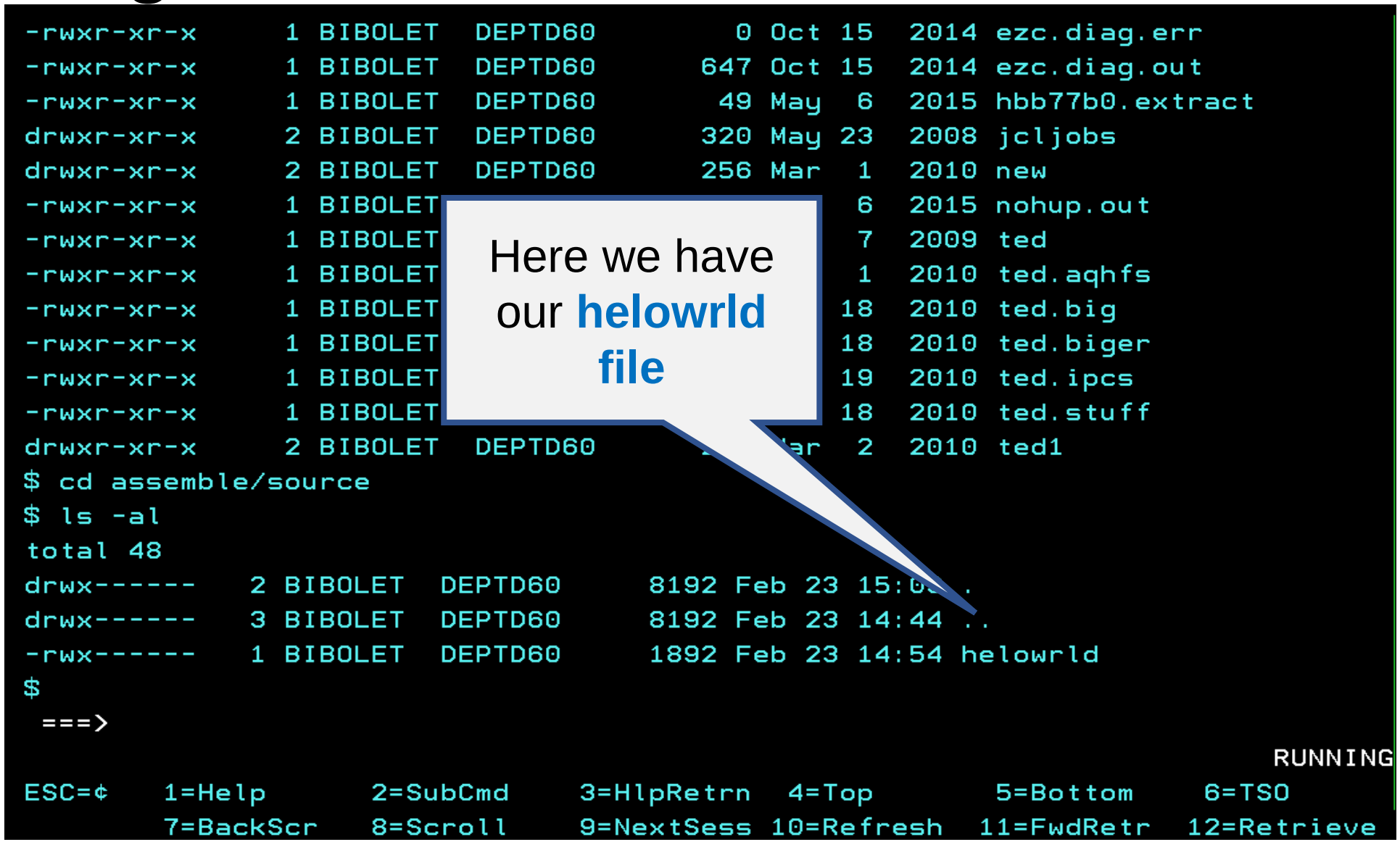
Opening screen

Interacting with z/OS – OMVS

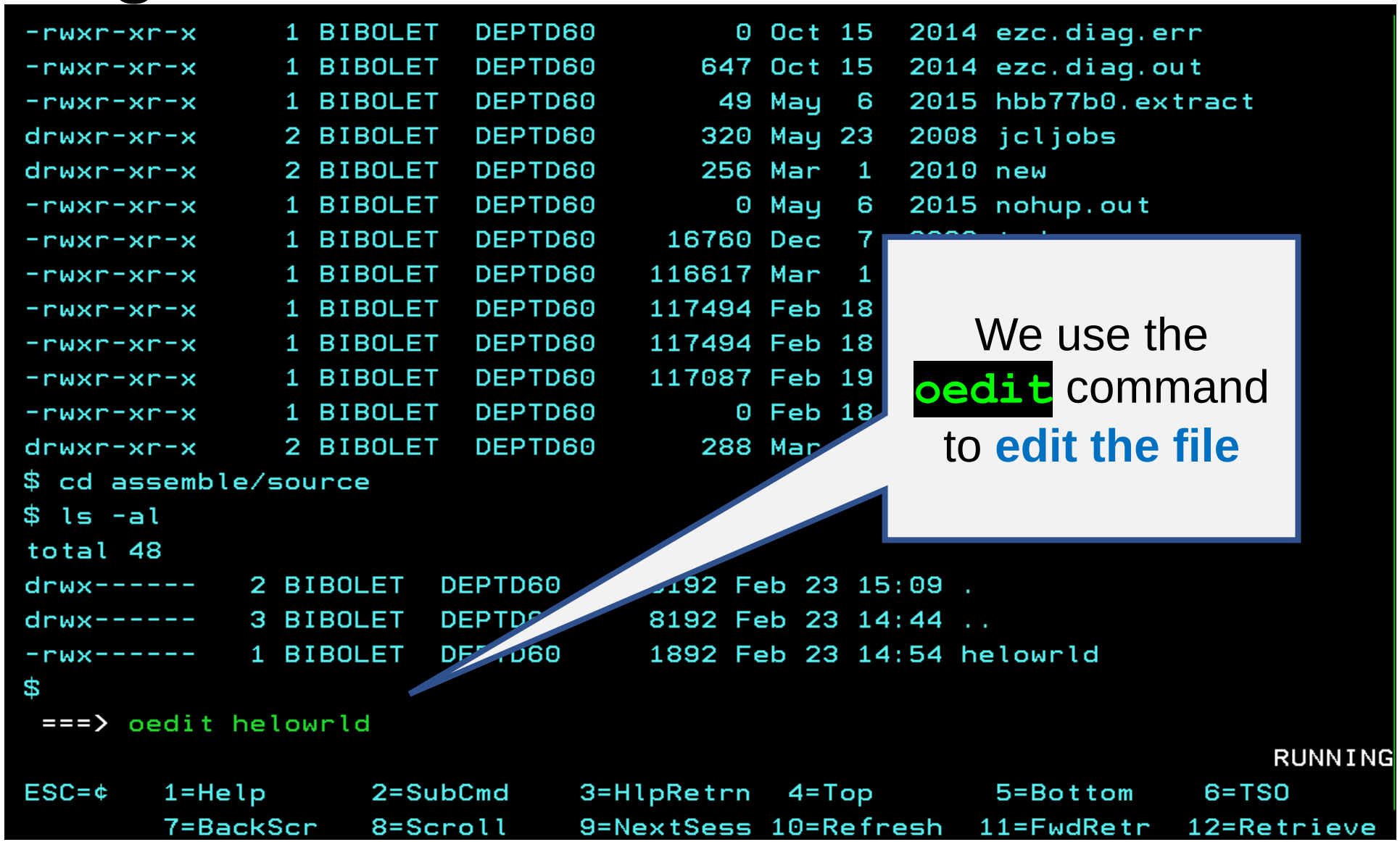


Interacting with z/OS – OMVS

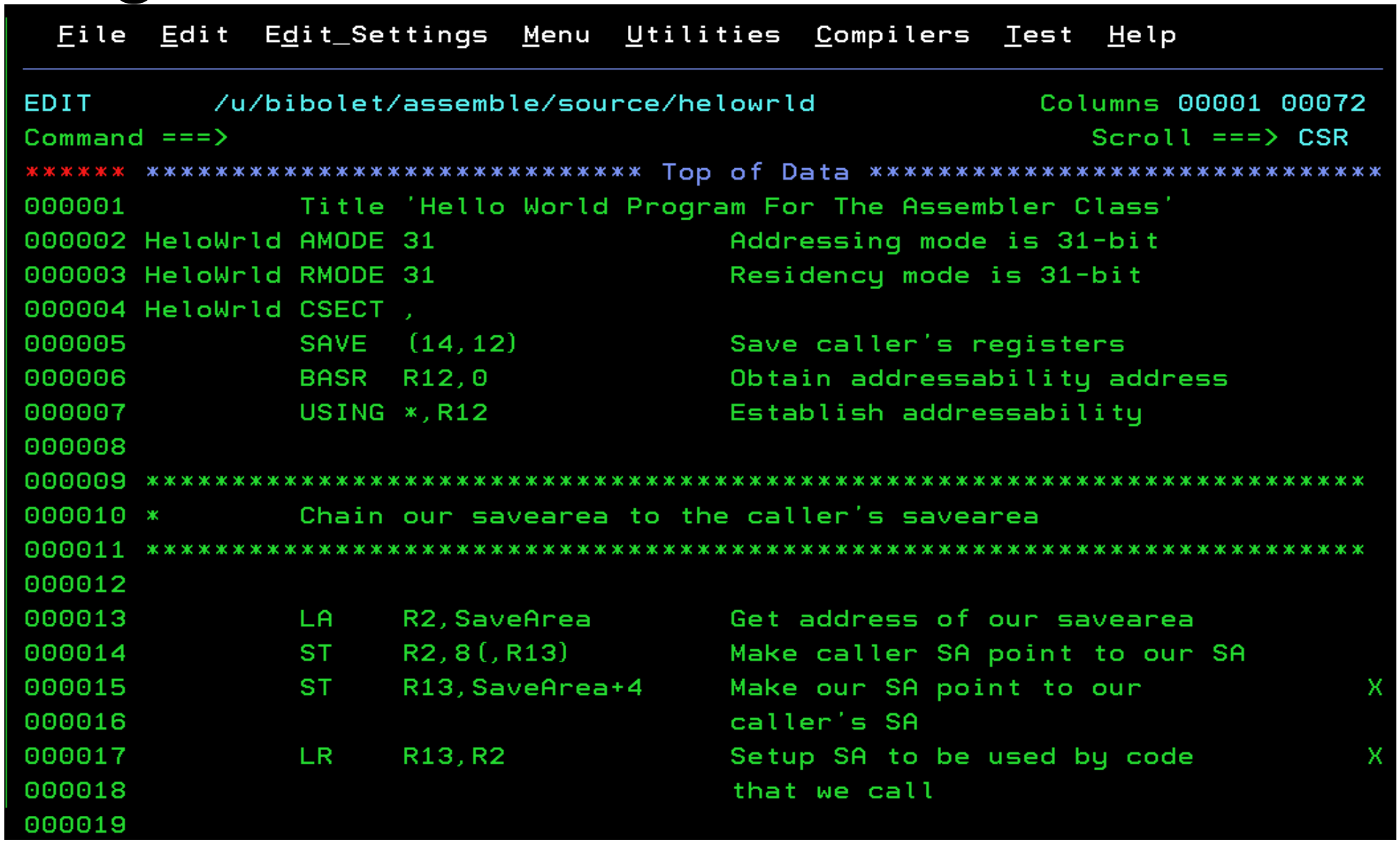
Interacting with z/OS – OMVS



Interacting with z/OS – OMVS



Interacting with z/OS – OMVS

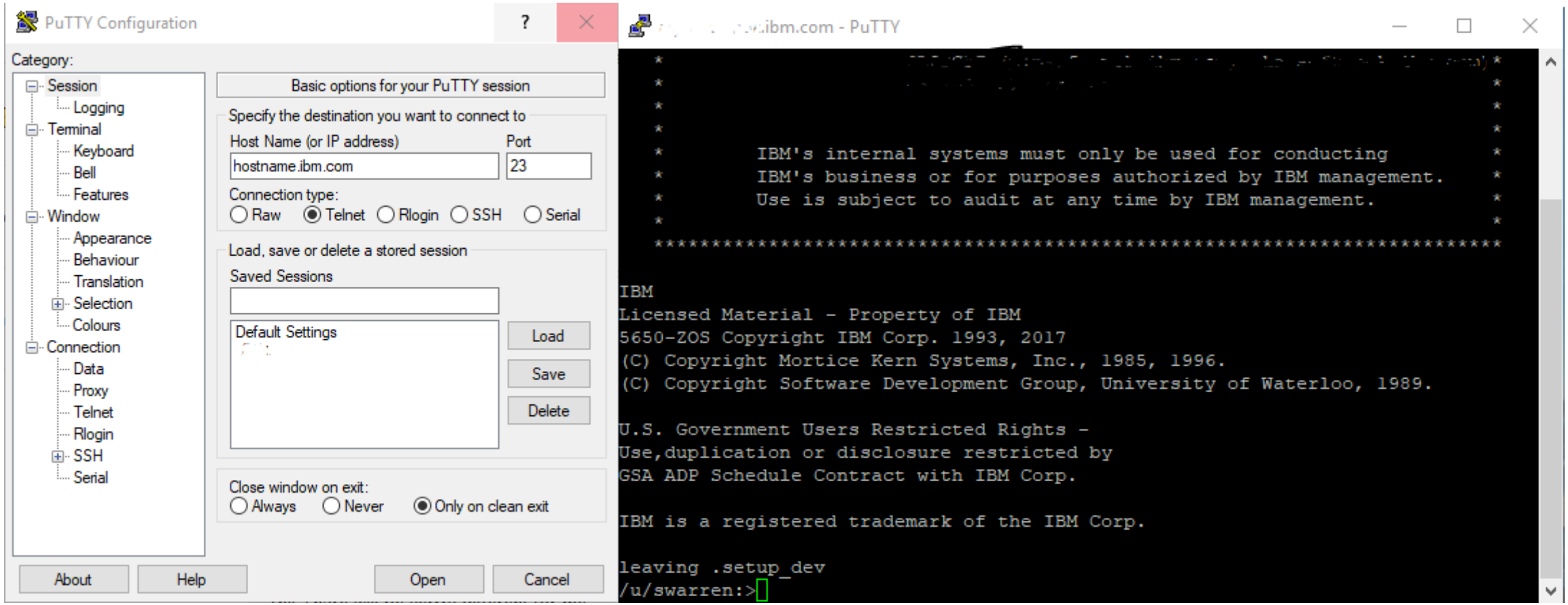


Remote UNIX Interfaces

Interacting with z/OS - Telnet

- Remote command execution
- Connection may be encrypted
- Automatically converts EBCDIC on the mainframe side to ASCII on the user side
- Telnet client is needed
- Userid/password of valid z/OS userid required

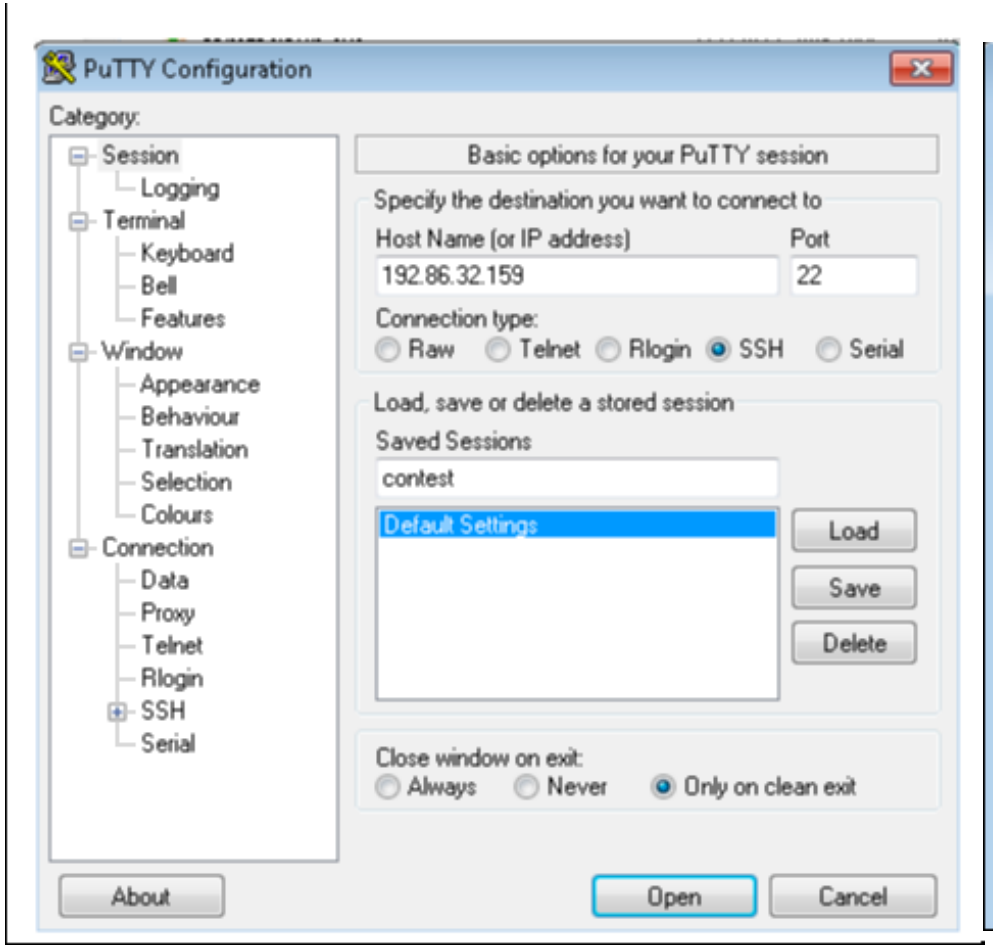
Interacting with z/OS – Telnet

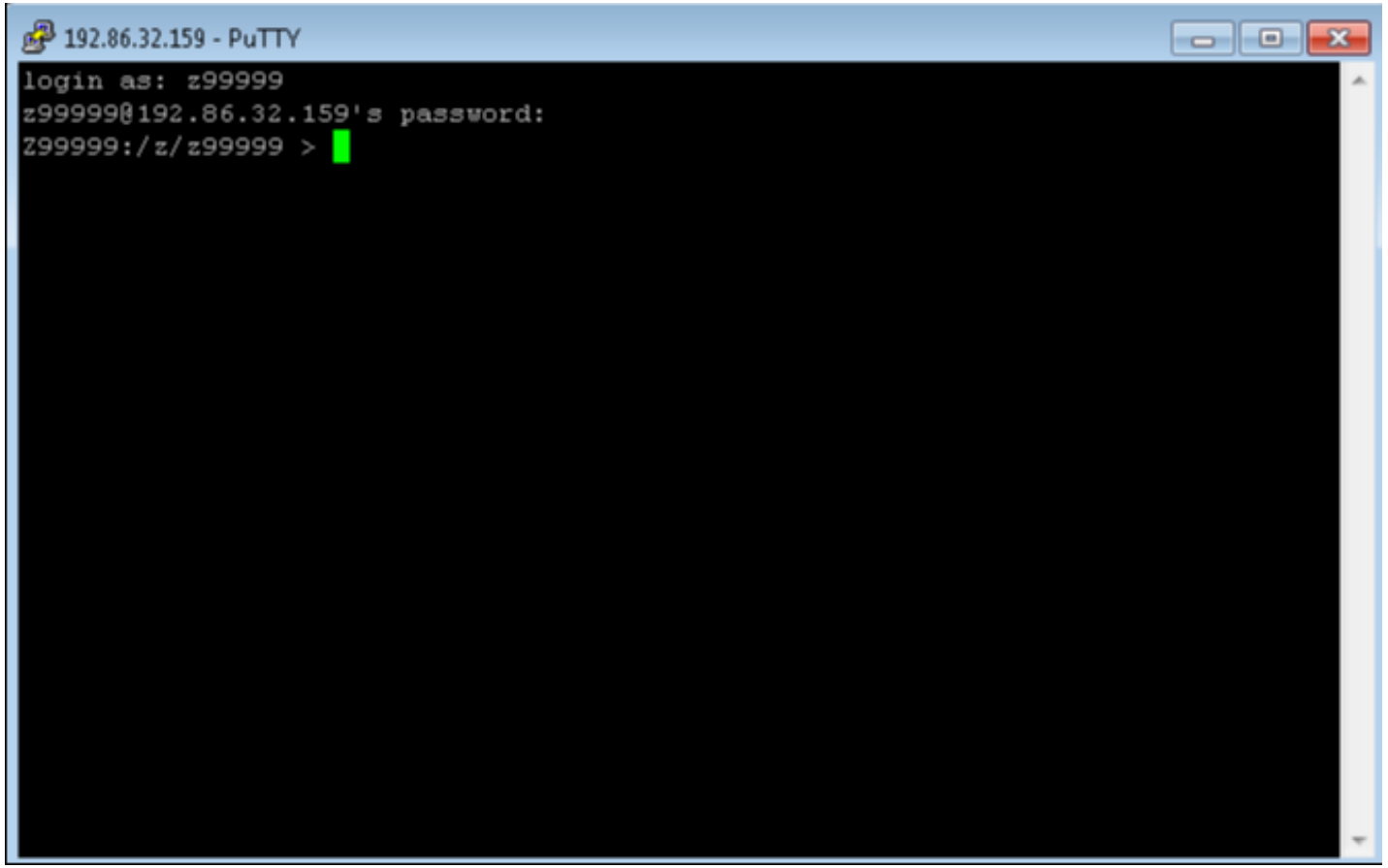


Interacting with z/OS – SSH

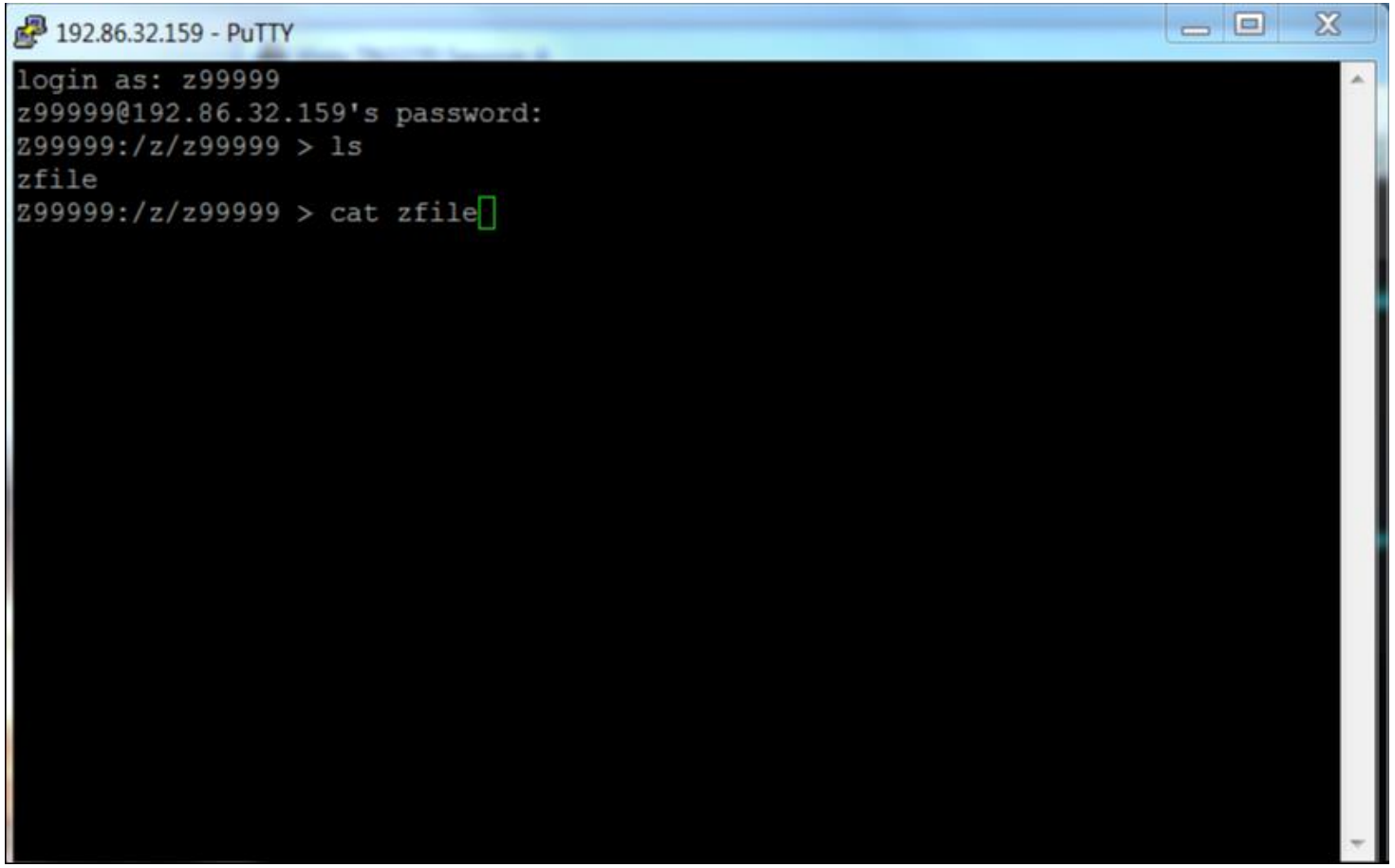
- Remote command execution
- Connection is encrypted
- Automatically converts EBCDIC on the mainframe side to ASCII on the user side
- SSH client is needed
- Public/private encryption key pair needs to be generated
  - No userid/password needs to be specified
  - Public key needs to be stored on z/OS

Interacting with z/OS – SSH





Interacting with z/OS – SSH



Interacting with z/OS – NFS

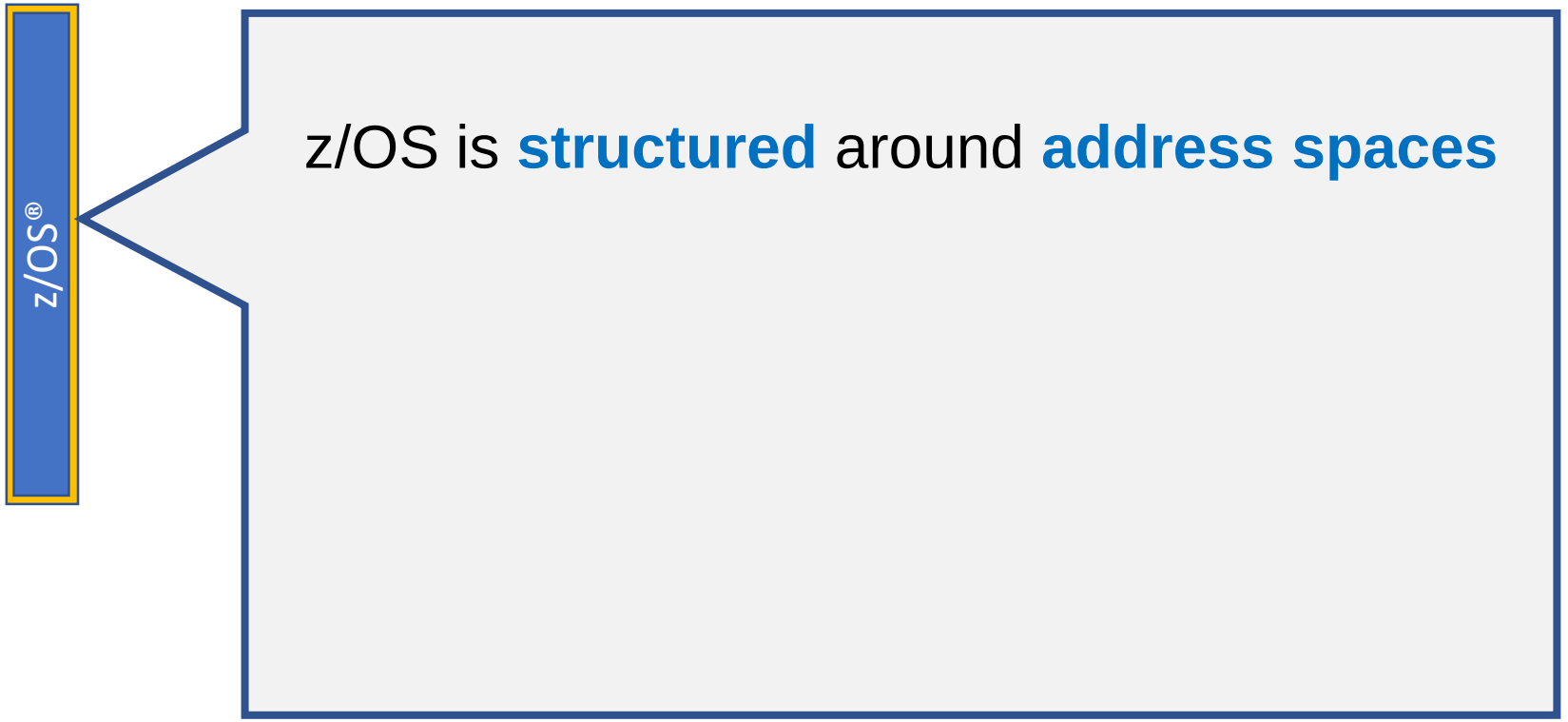
- Network file system
  - Acts as a file server to remote systems
- Client sees data sets or files as if they are local resources
- Access to both traditional z/OS data sets and z/OS UNIX files
- Connection may be encrypted
- Automatically converts EBCDIC on the mainframe side to ASCII on the user side
- NFS client is needed
- mvslogin command required to logon to z/OS
- mount command used to make a connection between a drive letter of local system and z/OS data sets or z/OS UNIX directories

Interacting with z/OS – NFS

```
c:\znfs-client-utils>mvslogin mvshost smith
GFSA968I UNIX uid=502/gid=1000 for user JSmith obtained
from local passwd file.
Password required
GFSA973A Enter MVS password for SMITH: ********
GFSA955I SMITH logged in ok.

c:\znfs-client-utils>mount \\mvshost\mvs\smith J:
J: is now successfully connected to \\mvshost\mvs\smith
The command completed successfully.

c:\znfs-client-utils> J:

j:\>
```

Address Spaces

z/OS Address Spaces



z/OS Address Spaces

z/OS is **structured** around **address spaces**

Address spaces **are ranges of addresses** in virtual storage

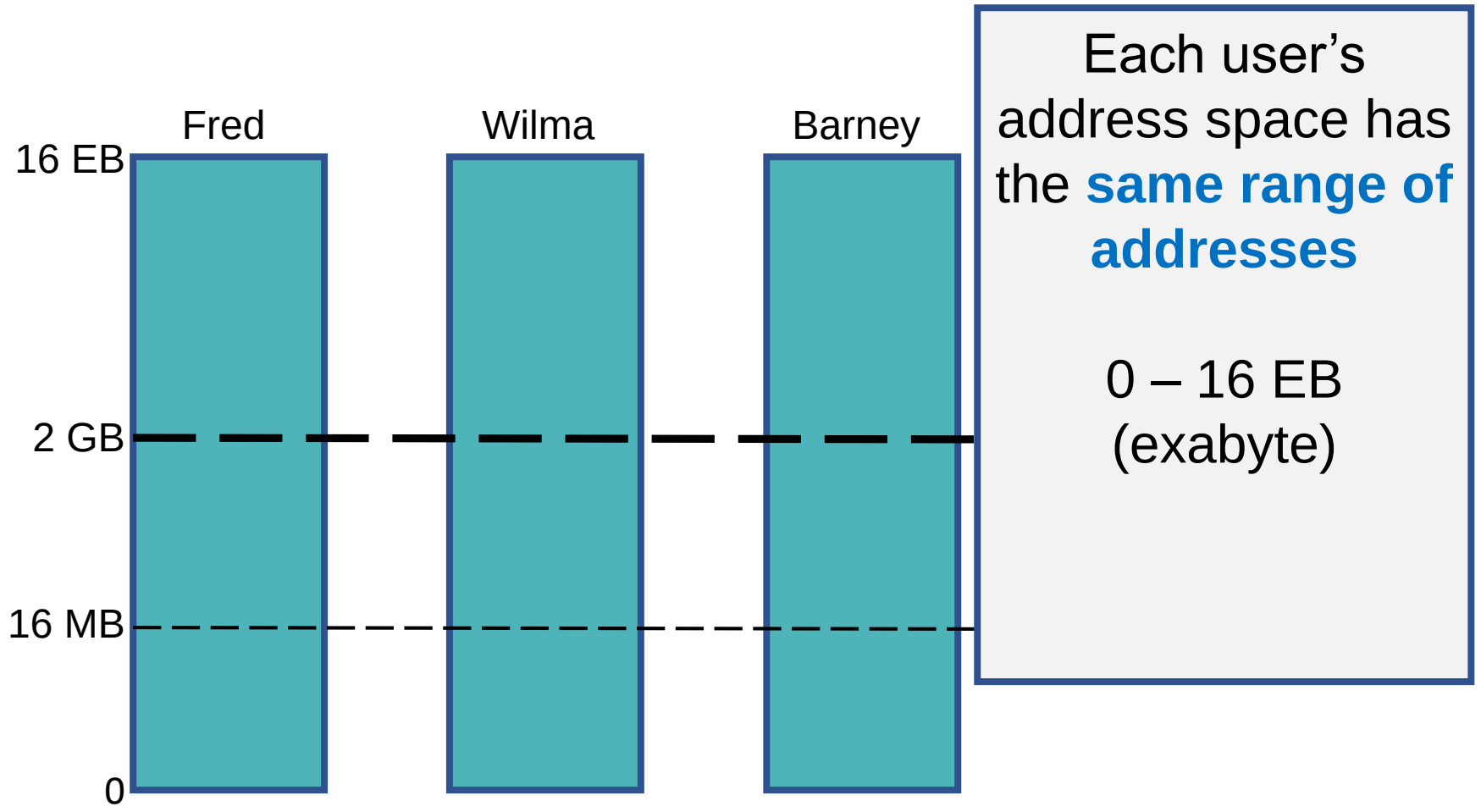
z/OS Address Spaces

z/OS is **structured** around **address spaces**

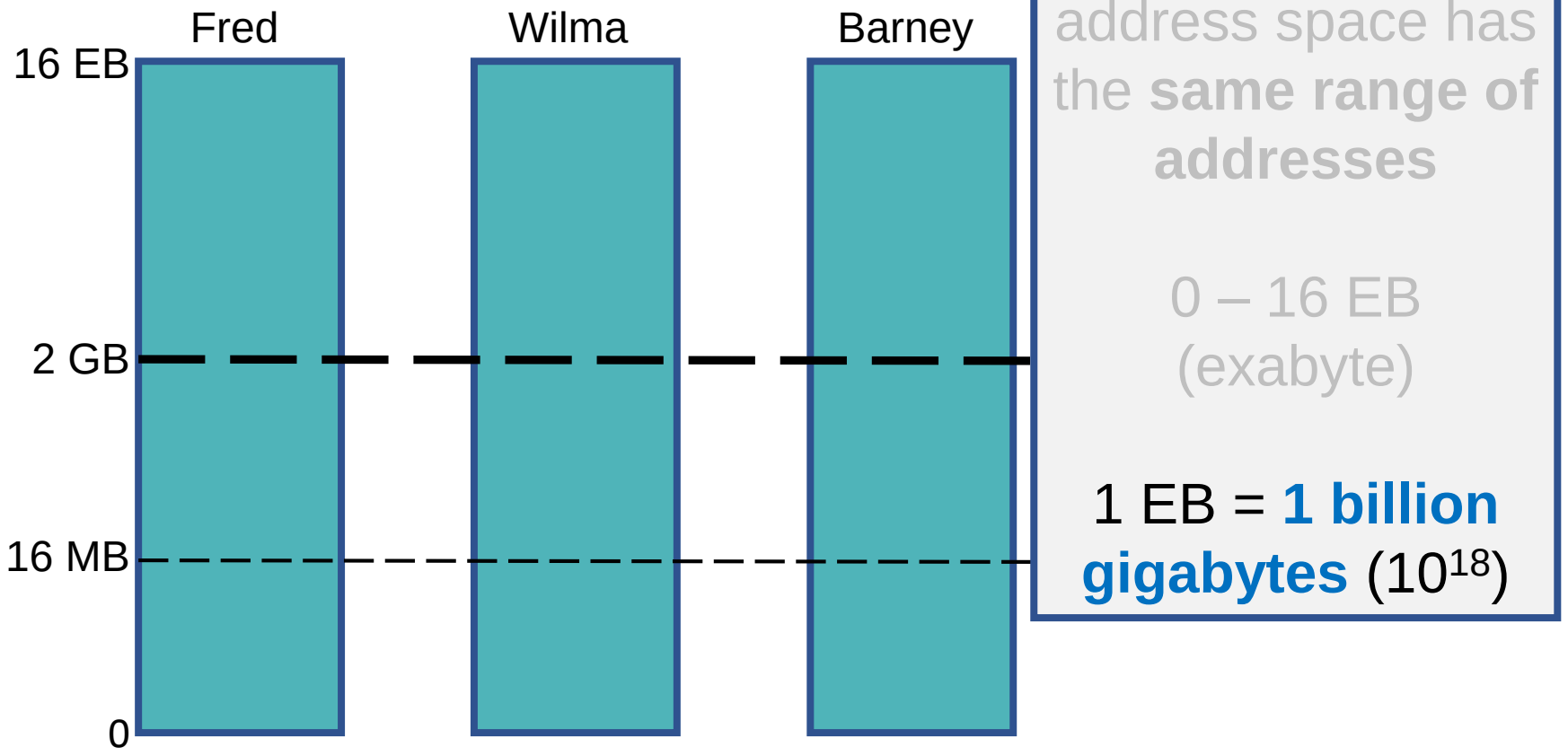
Address spaces **are ranges of addresses** in virtual storage

Each user gets an **address space** containing the **same range of addresses**

z/OS Address Spaces



z/OS Address Spaces

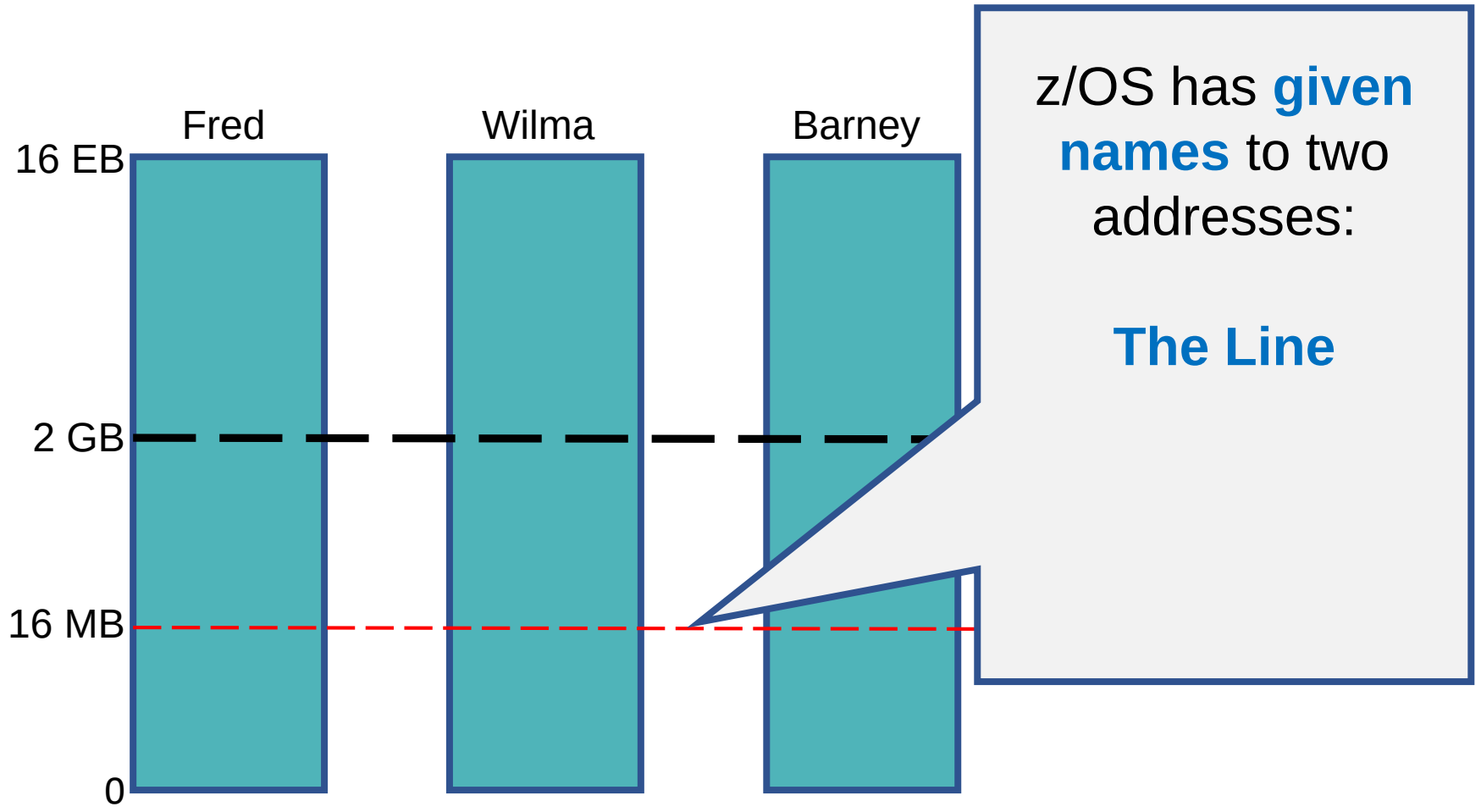


Each user's address space has the **same range of addresses**

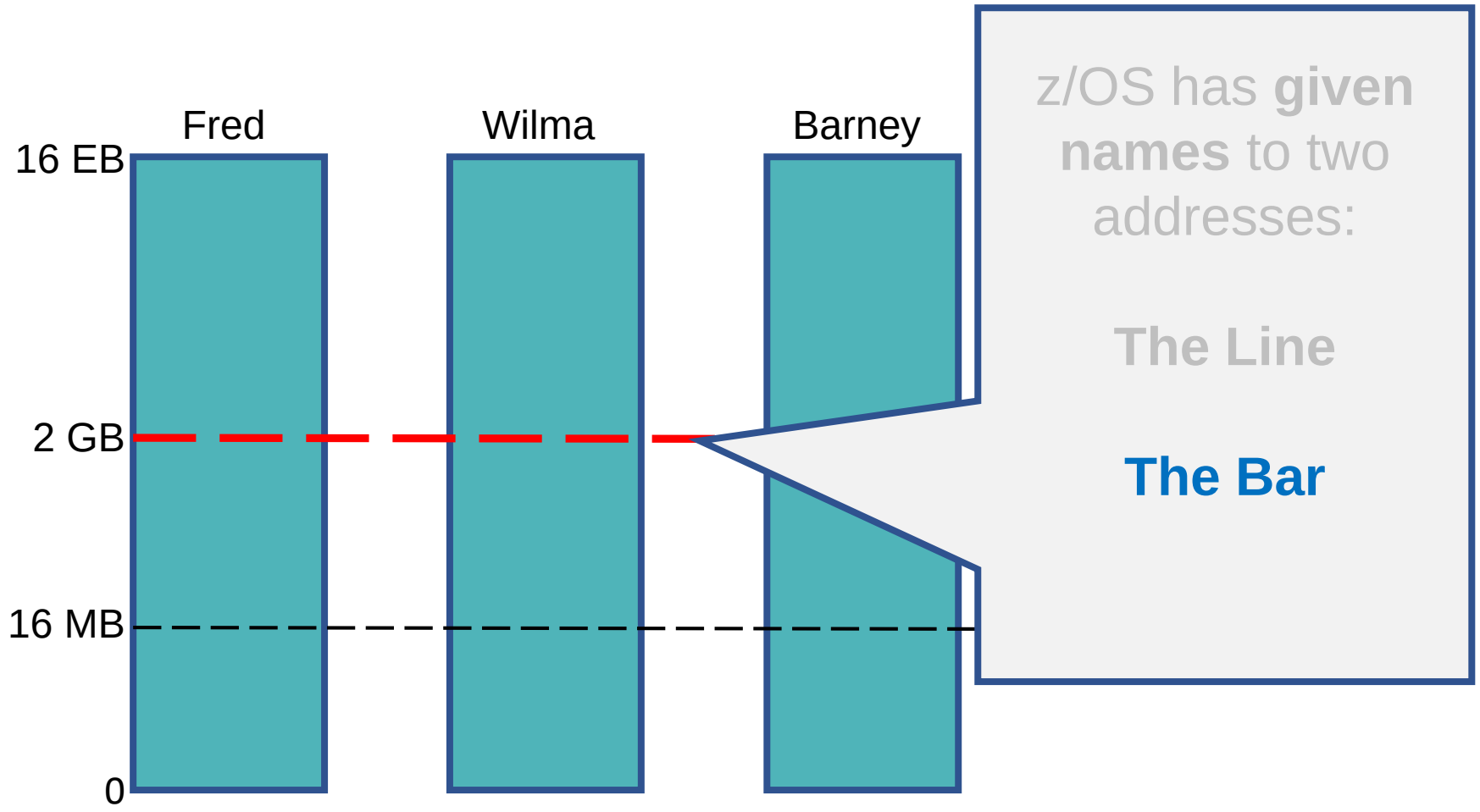
0 – 16 EB (exabyte)

1 EB = **1 billion gigabytes** (10^{18})

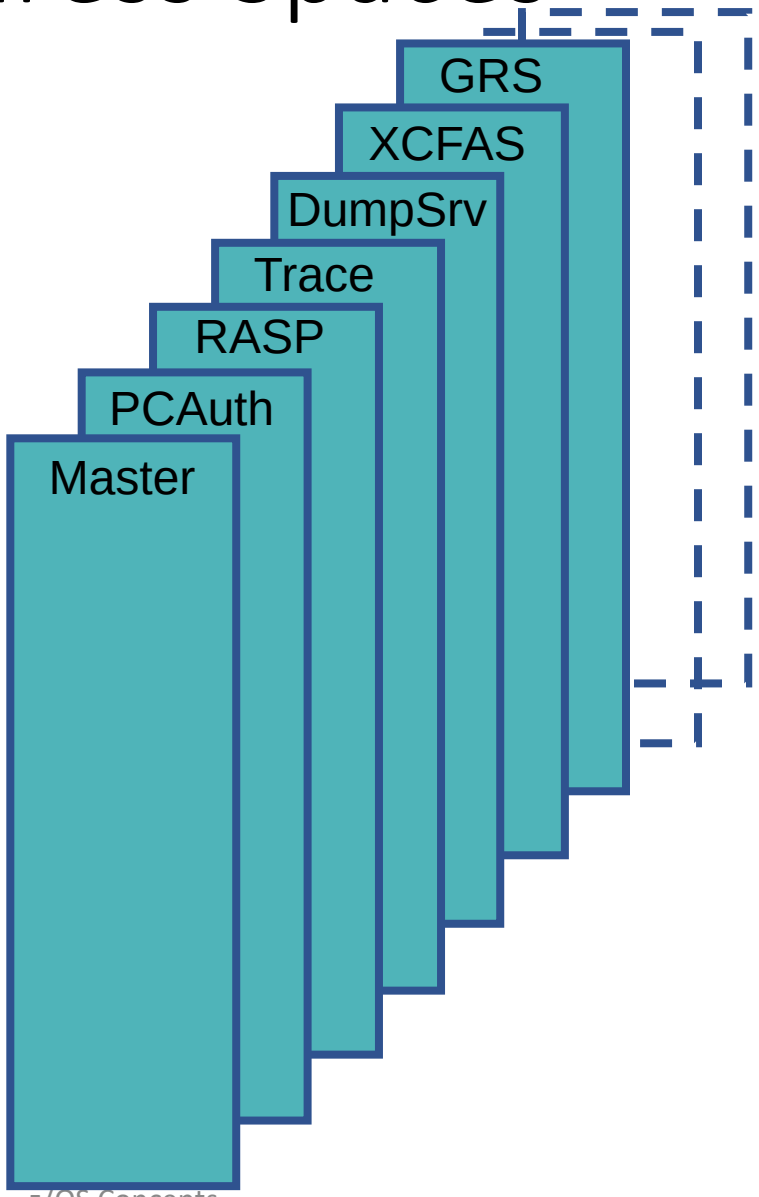
z/OS Address Spaces



z/OS Address Spaces

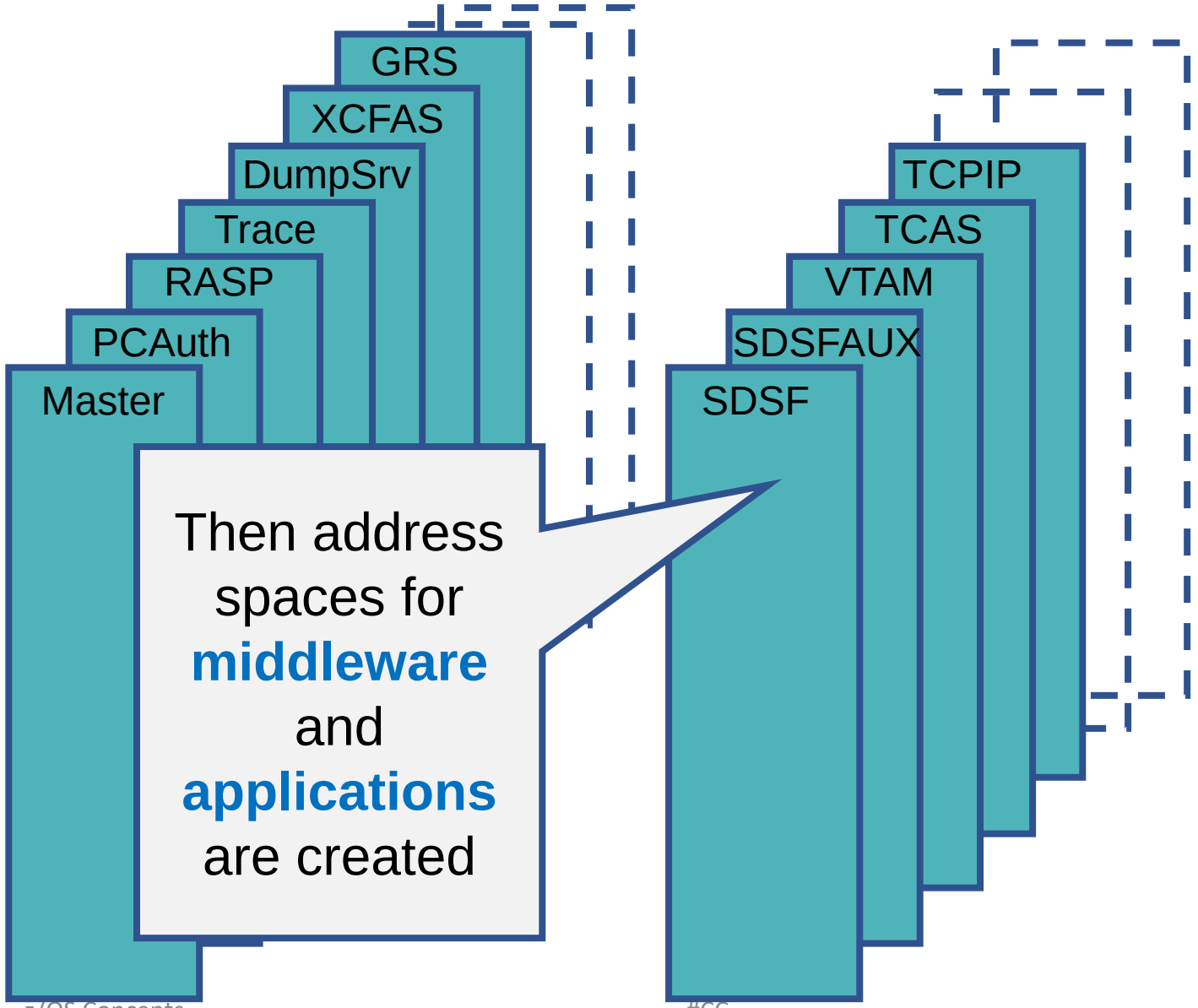


z/OS Address Spaces



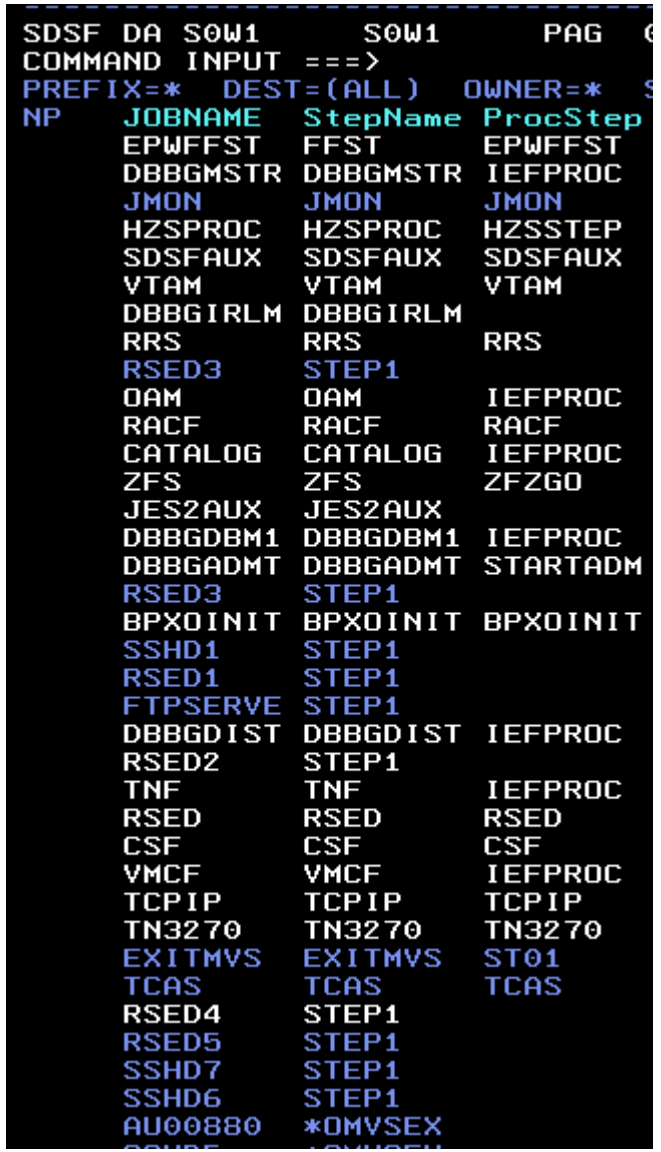
Besides an address space for each user, **z/OS creates address spaces (~35)** for some of its **internal components**

z/OS Address Spaces



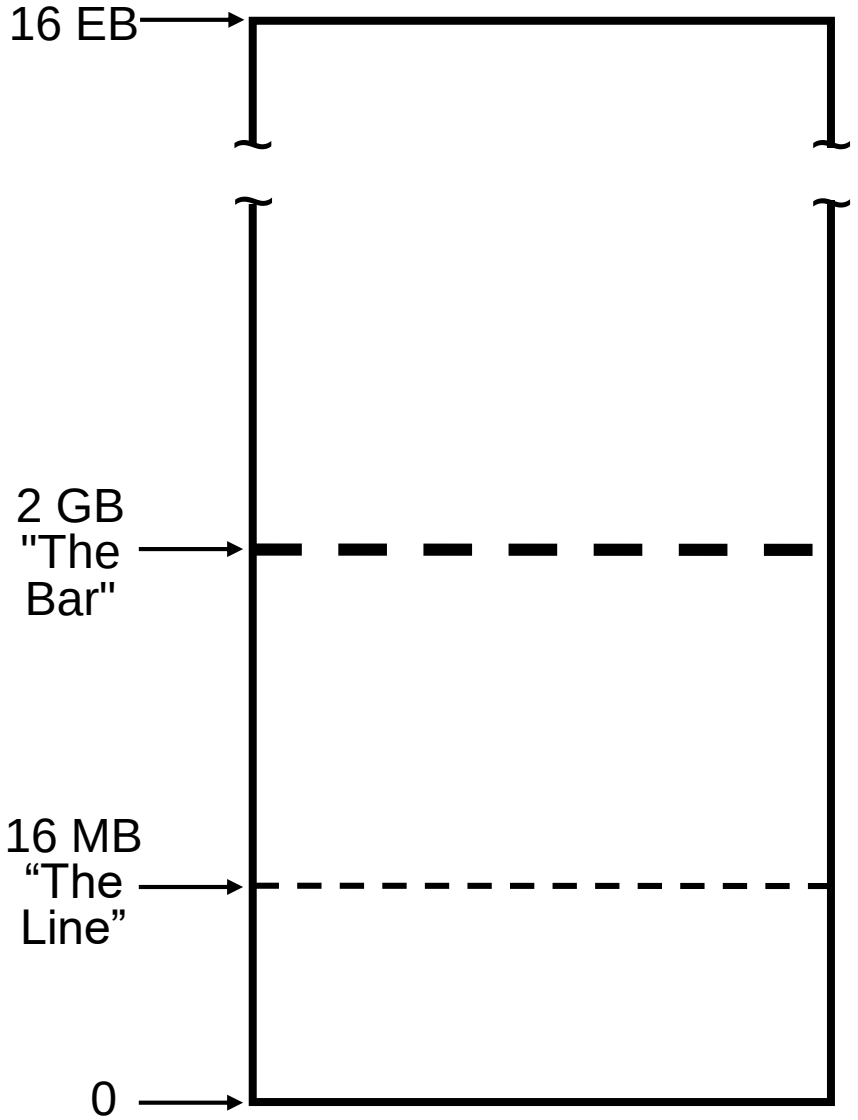
z/OS Address Spaces

Address spaces listed in SDSF



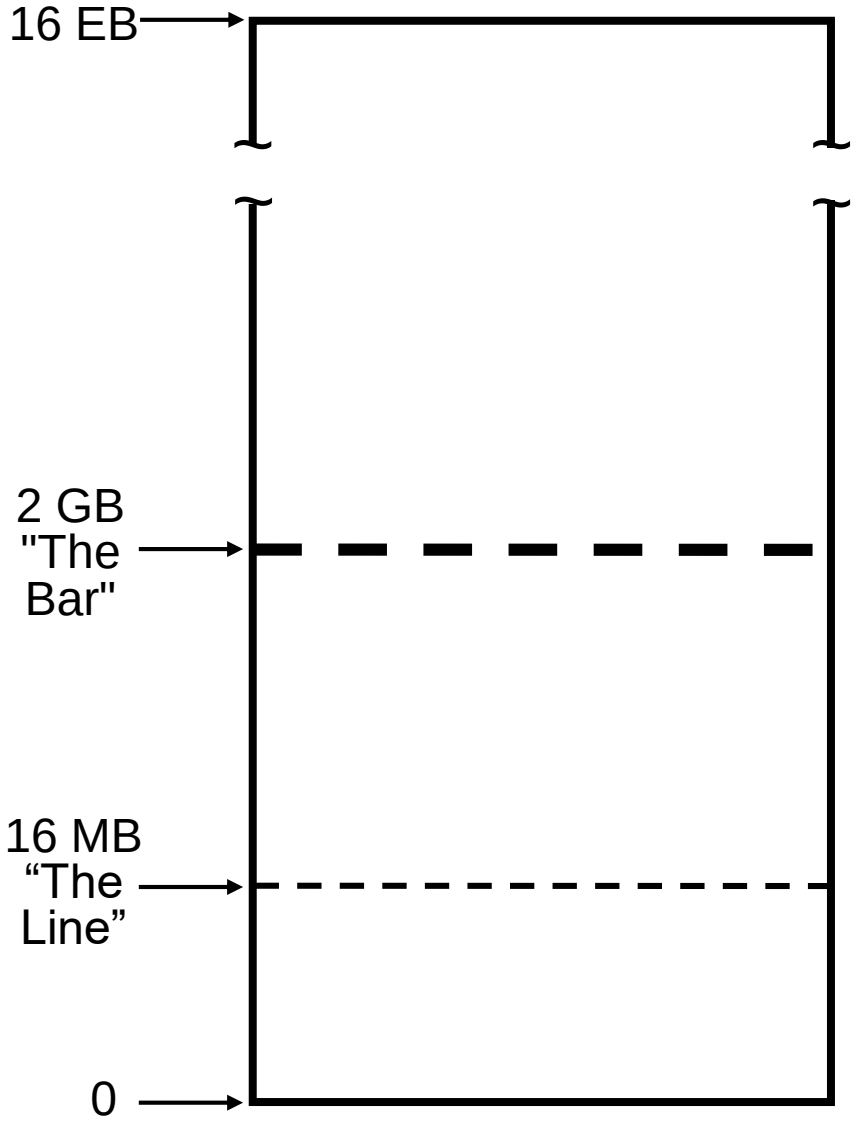
What makes up an Address Space?

What Makes Up an Address Space?



So you have 16 exabytes **all for your use. Right**?

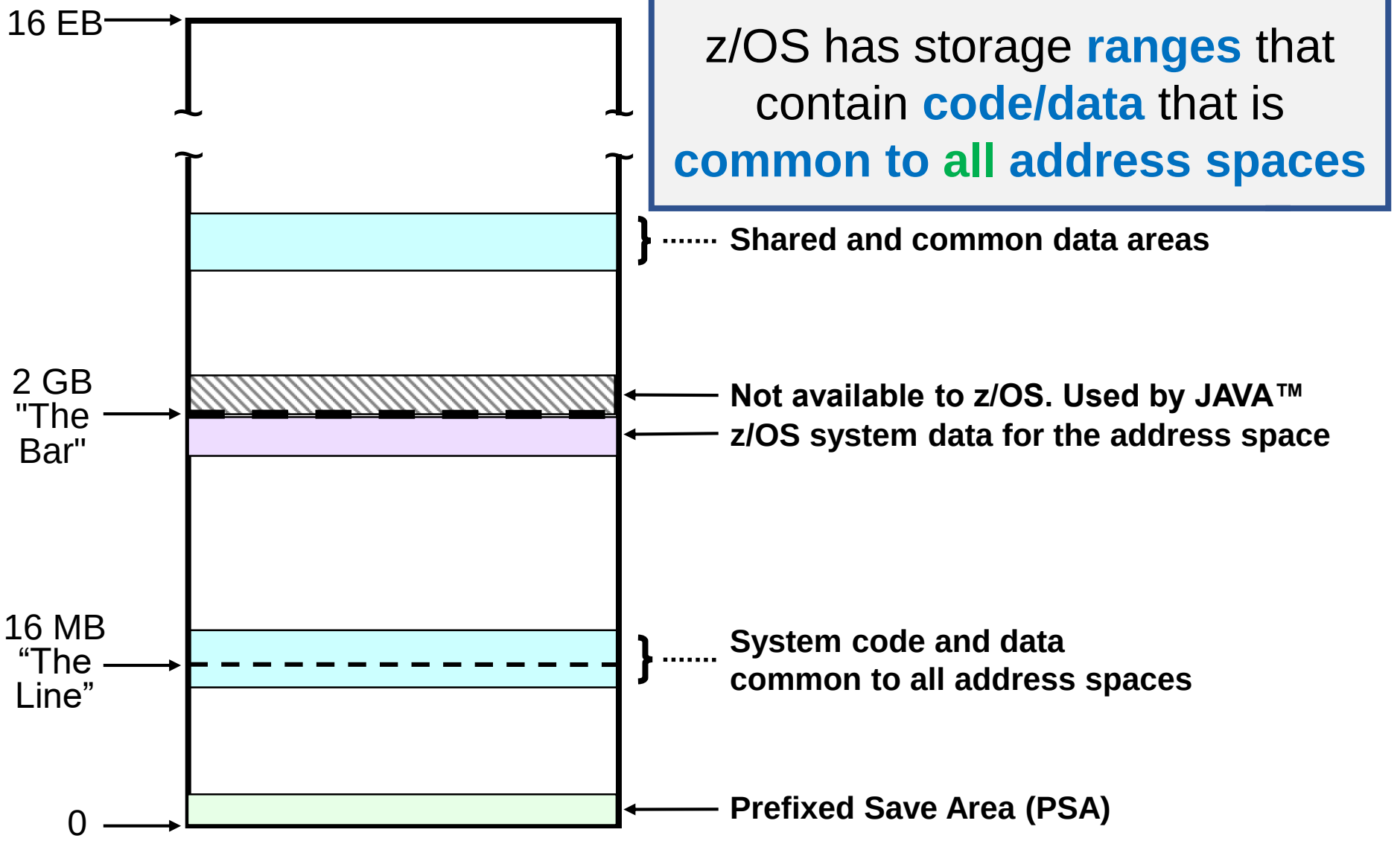
What Makes Up an Address Space?



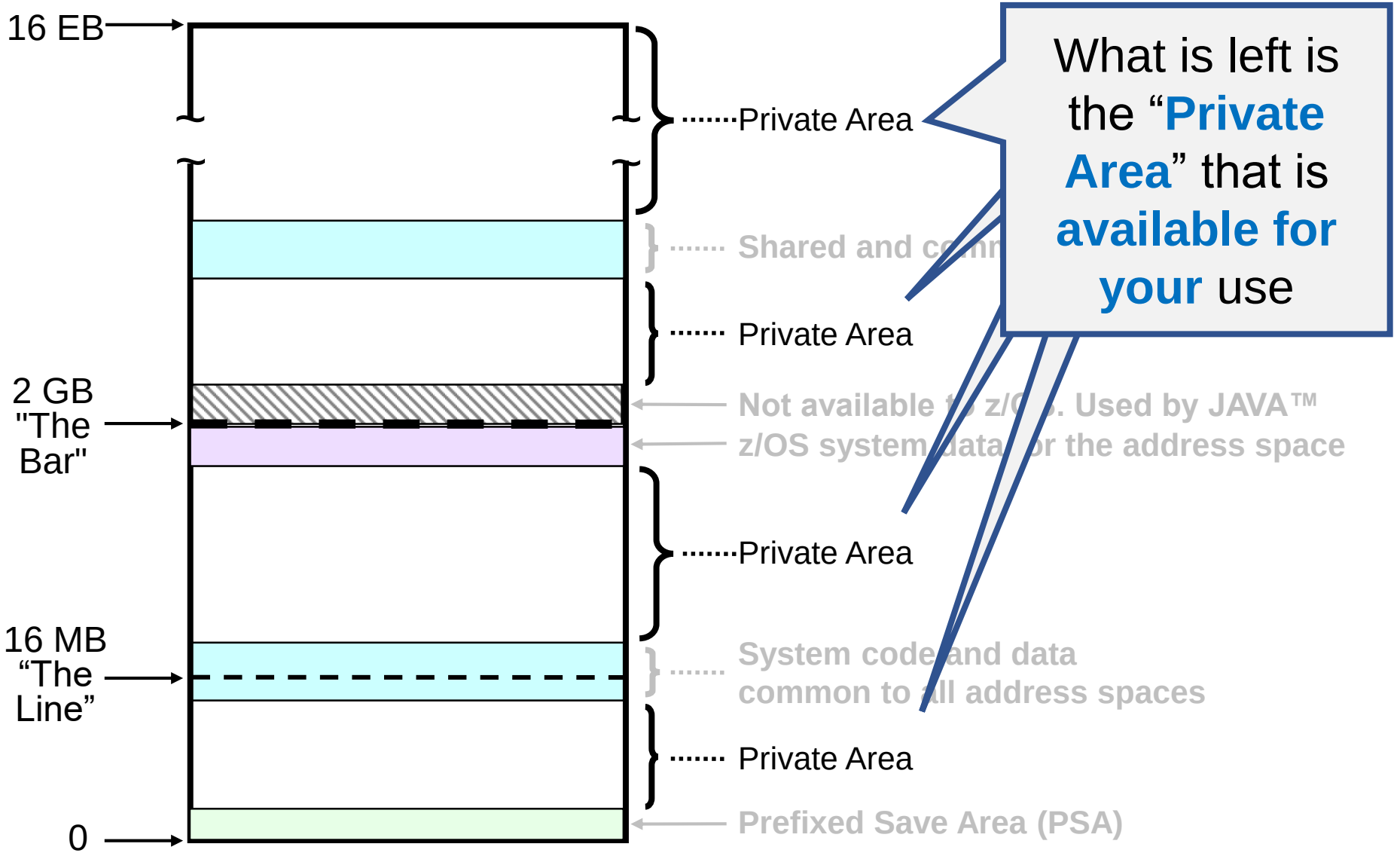
So you have 16 exabytes **all for your use. Right**?

Wrong!

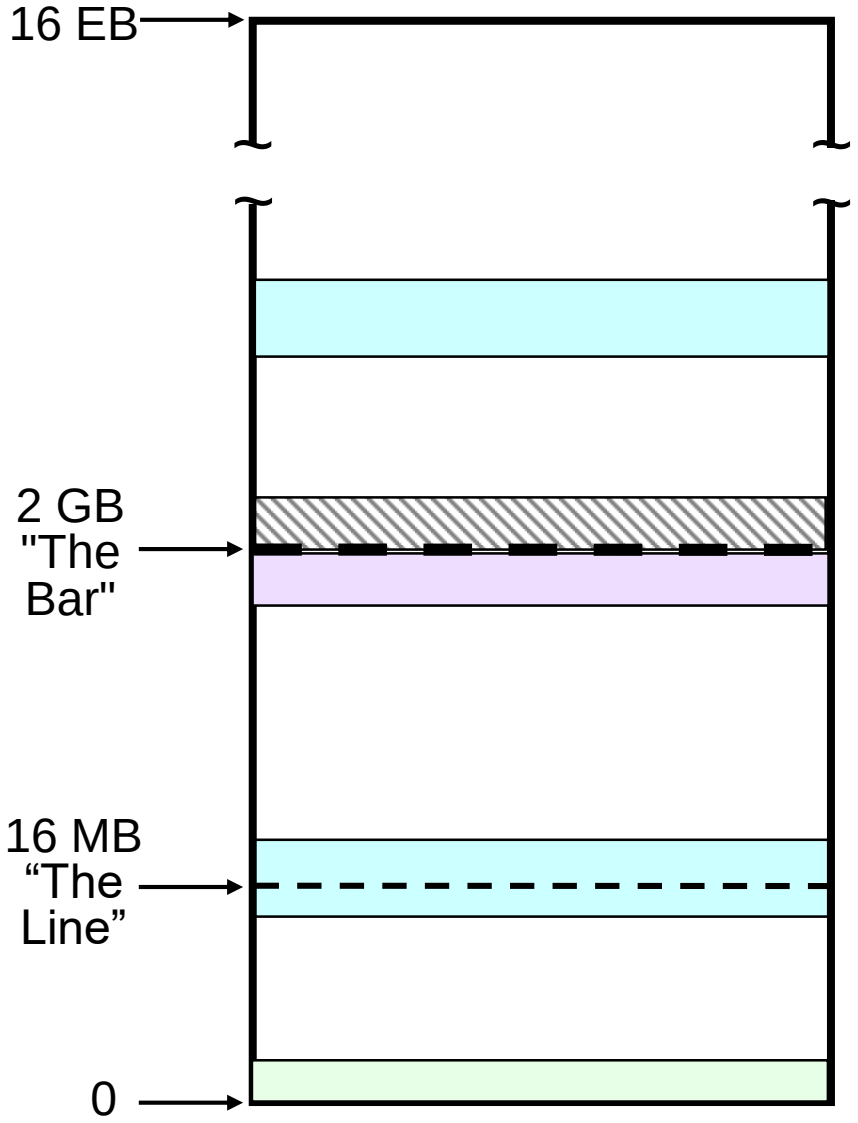
What Makes Up an Address Space?



What Makes Up an Address Space?



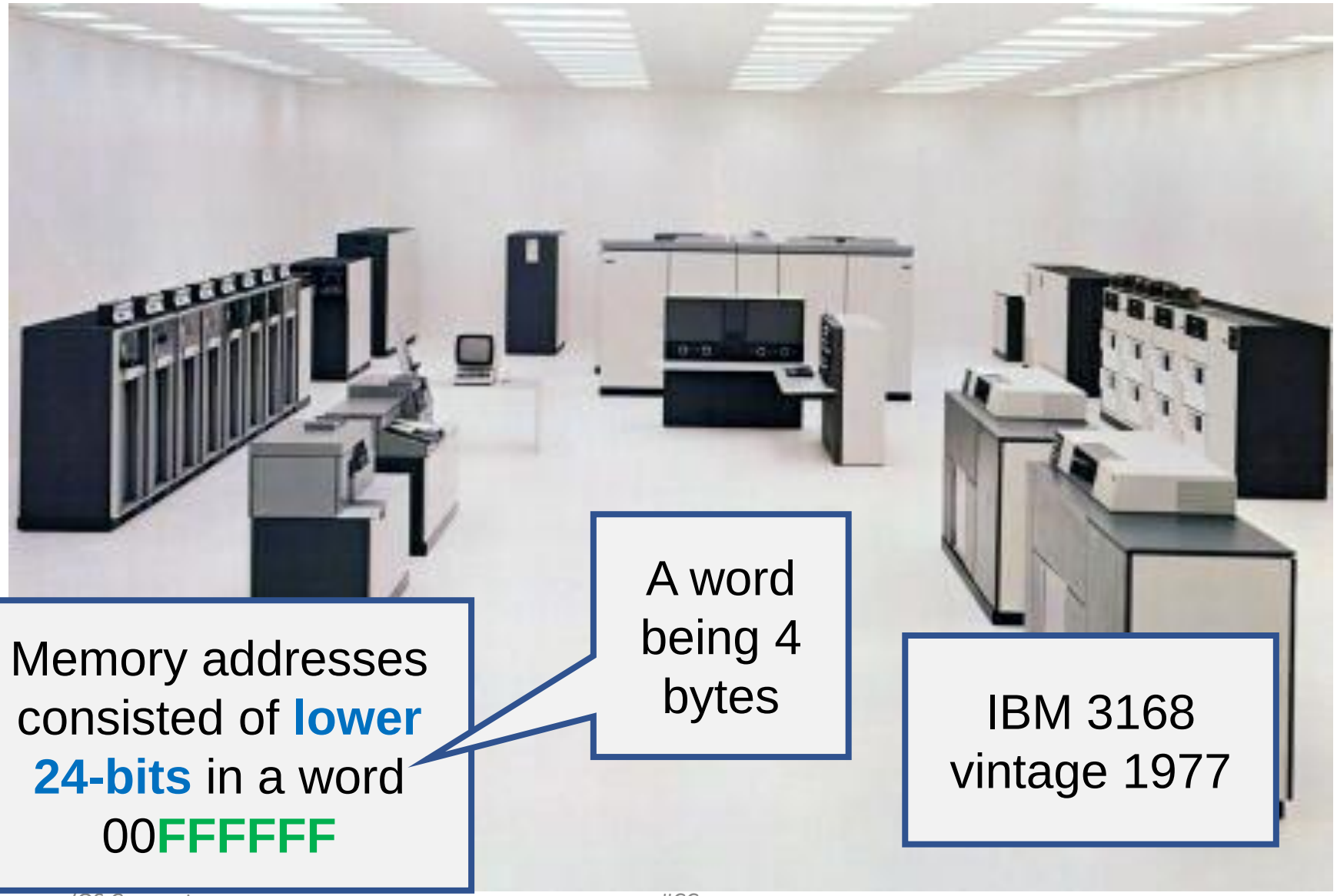
What Makes Up an Address Space?



But **how much** can your program see (i.e., **address**)?

Addressing Modes

Addressing Modes

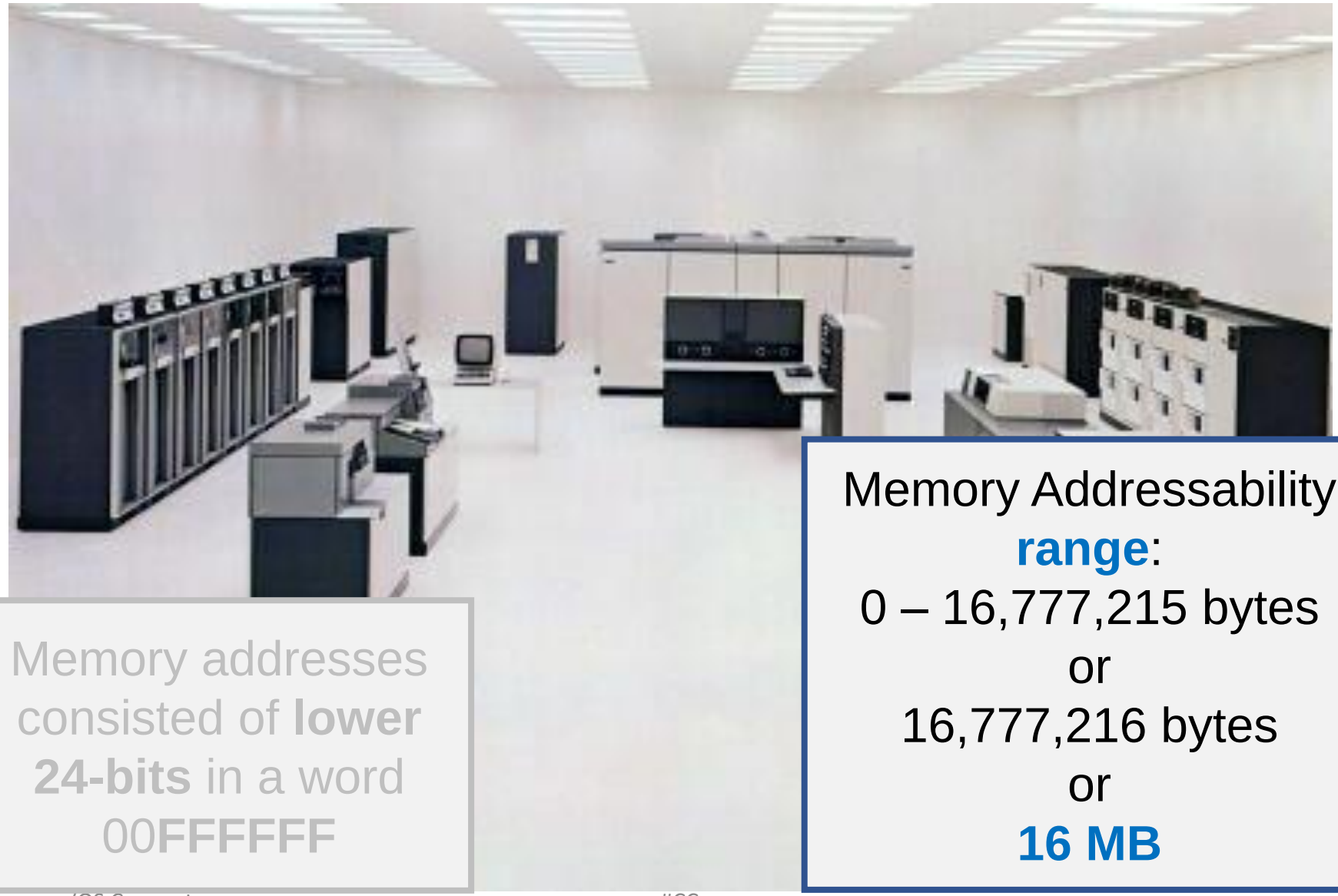


Memory addresses consisted of **lower 24-bits** in a word 00**FFFFFF**

A word being 4 bytes

IBM 3168 vintage 1977

Addressing Modes



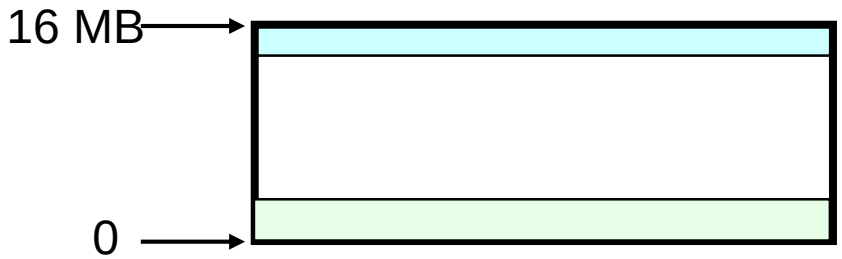
Memory addresses consisted of **lower 24-bits** in a word 00**FFFFFF**

Memory Addressability **range**:
0 – 16,777,215 bytes
or
16,777,216 bytes
or
16 MB

Addressing Modes

An address space **size** is **limited to** the amount of **memory that can be addressed**

In 1977, an address space was **limited to 16 MB**



Addressing Modes

Memory addresses consisted of **lower 31-bits** in a word **7FFFFFFF**

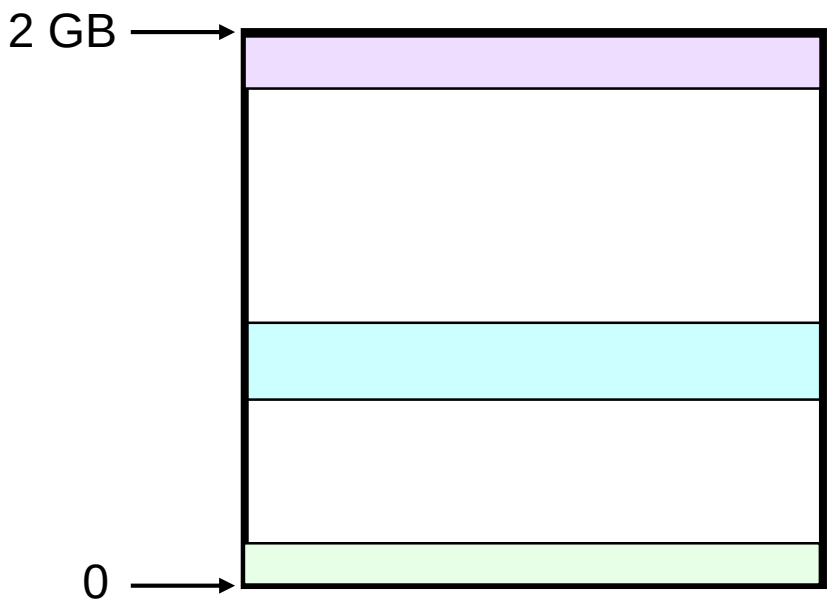
IBM 3081 vintage 1981

Addressing Modes

Memory addresses consisted of **lower 31-bits** in a word **7FFFFFFF**

Memory Addressability **range**:
0 – 2,147,483,647 bytes
or
2,147,483,648 bytes
or
2 GB

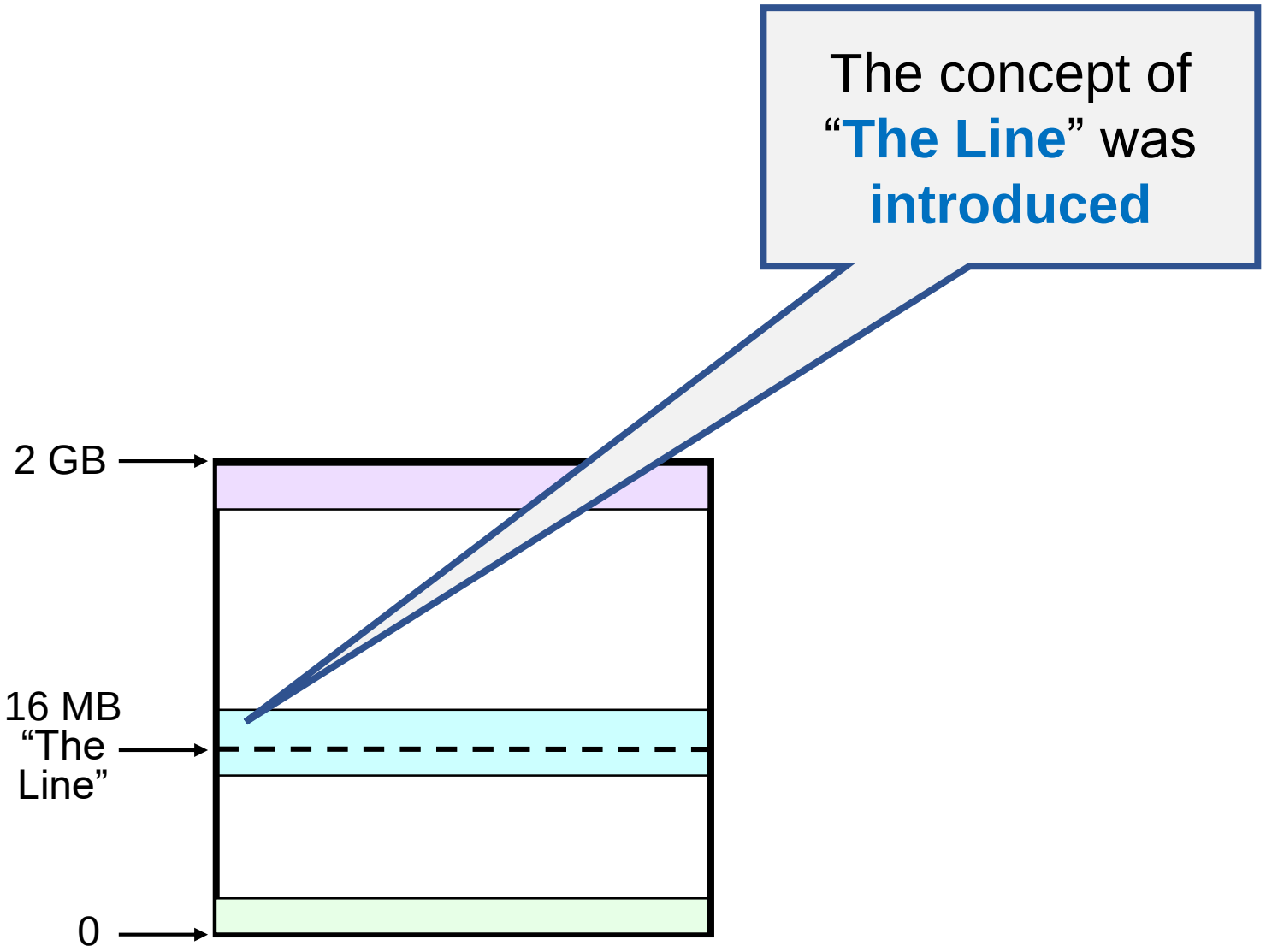
Addressing Modes



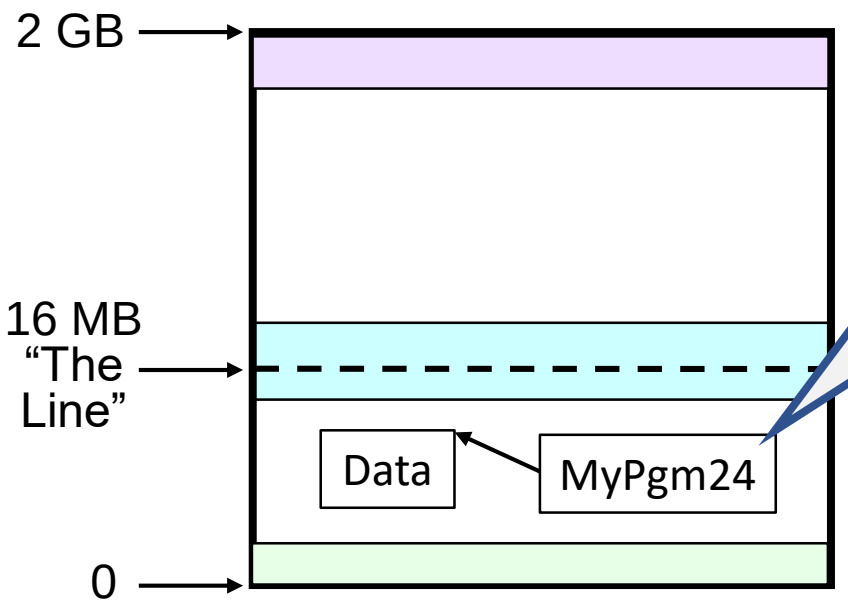
An address space **size** is **limited to** the amount of **memory that can be addressed**

In 1981, an address space was **limited to 2 GB**

Addressing Modes



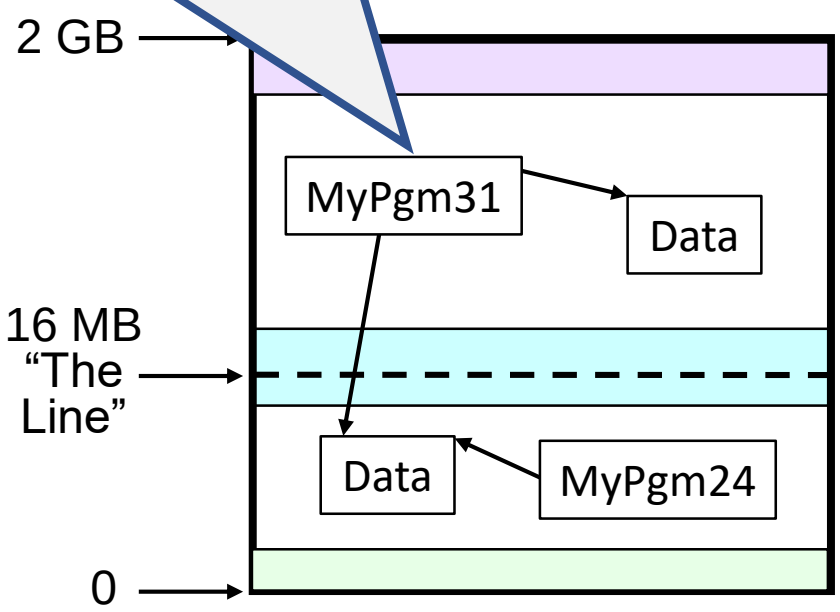
Addressing Modes



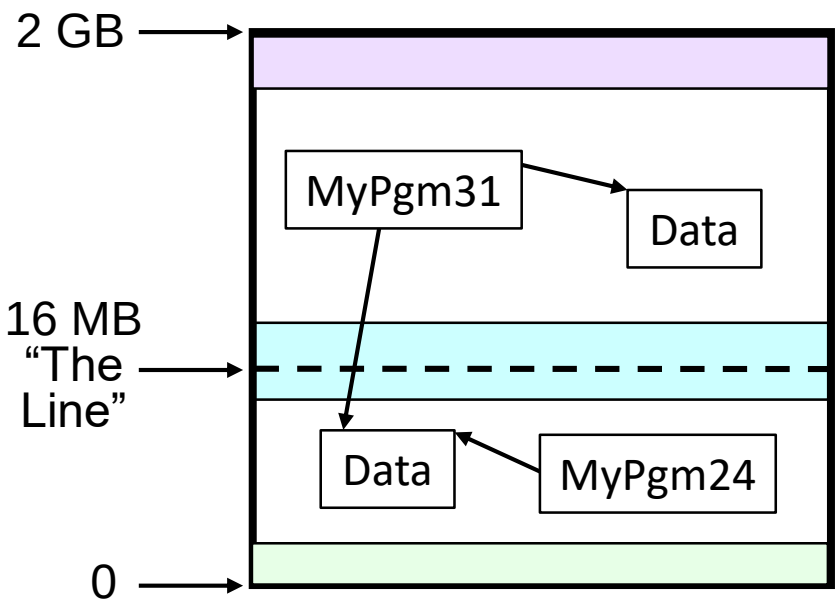
Programs **not changed** to support 31-bit addressing **had to run** and could **only access data** that was "**Below the line**"

Addressing Modes

Programs **changed** to support 31-bit addressing **could run** and could also **access data** that was "**Above the line**"



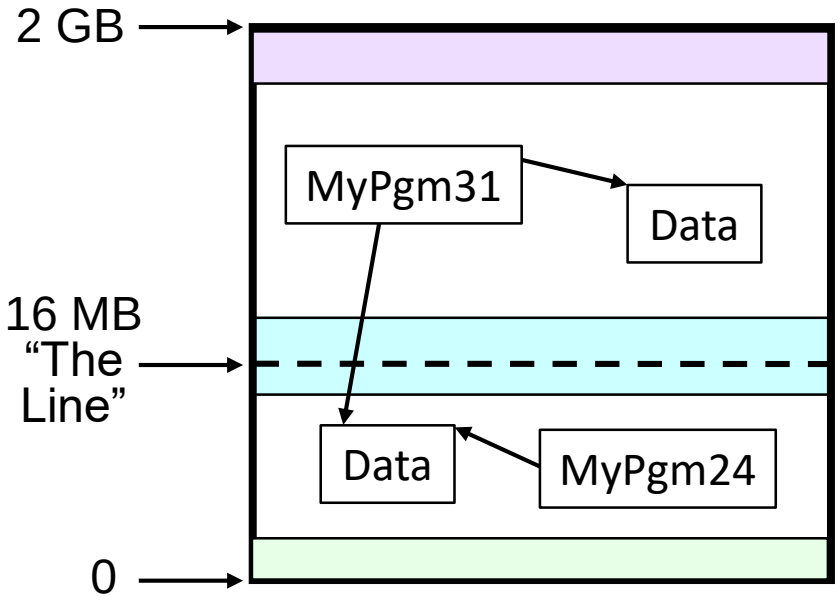
Addressing Modes



Needed a way to **indicate which addressing mode** the program **supported**

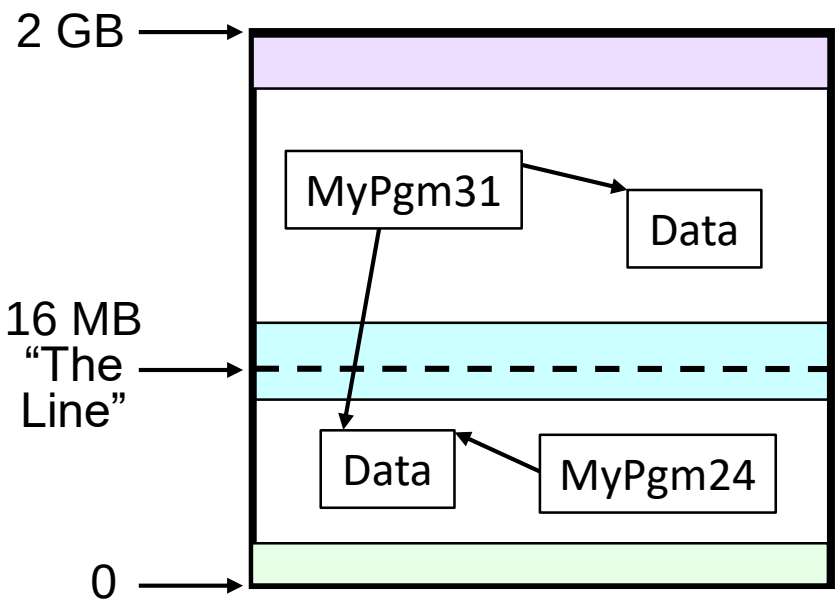
Programs now have to specify:
AMODE 24 or
AMODE 31

Addressing Modes



Since AMODE 31 programs **could run** (i.e., be loaded) **above or below the line**, a way was **needed to request** where the program should **reside**

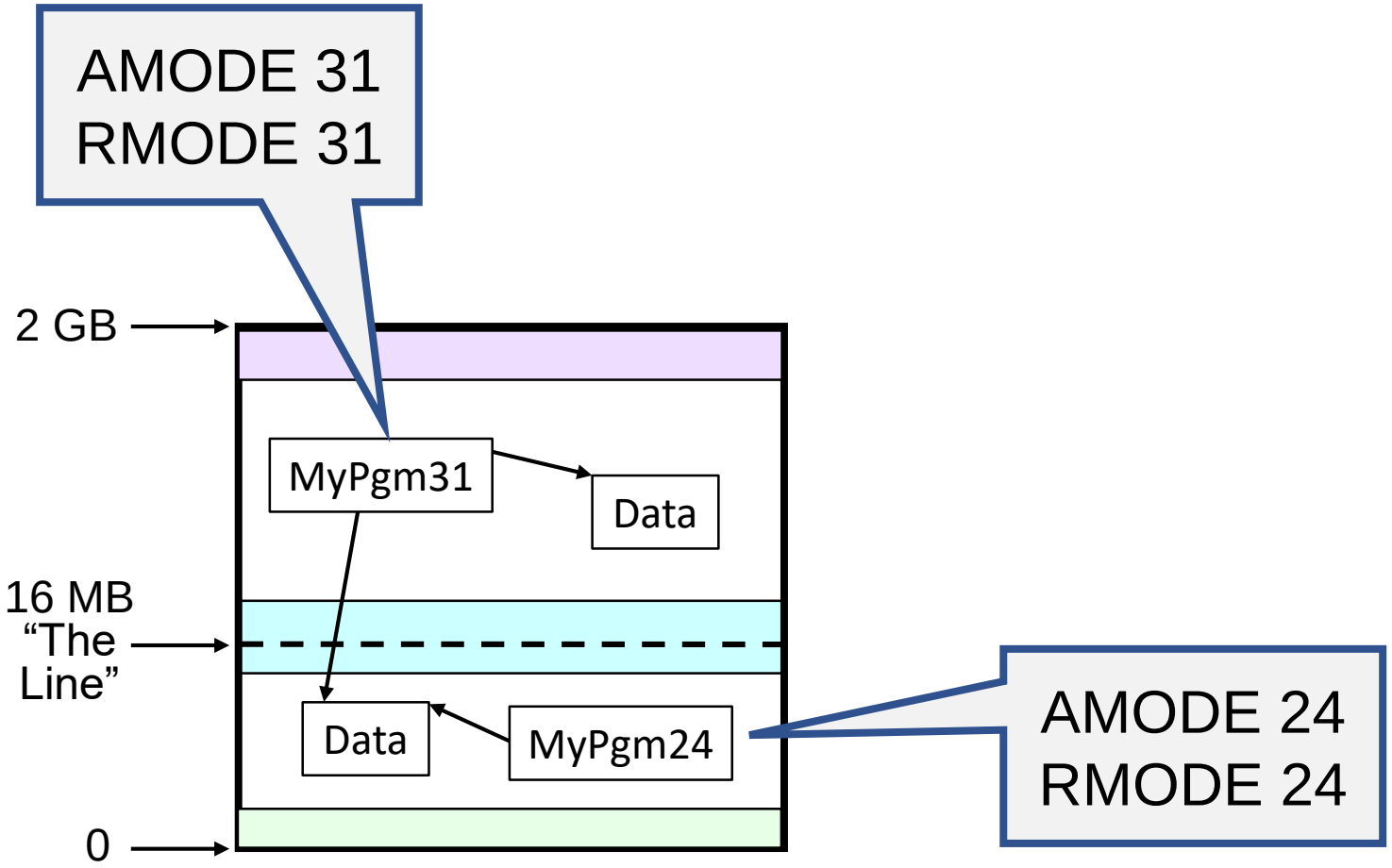
Addressing Modes



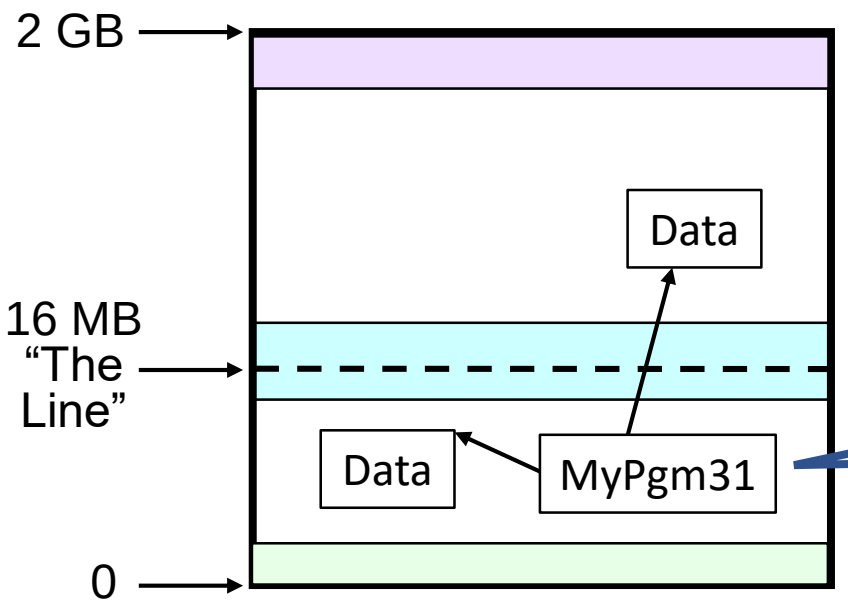
Programs now have to **specify a residency mode**:
RMODE 24 or
RMODE 31

so **z/OS knows where** to **load** the program

Addressing Modes



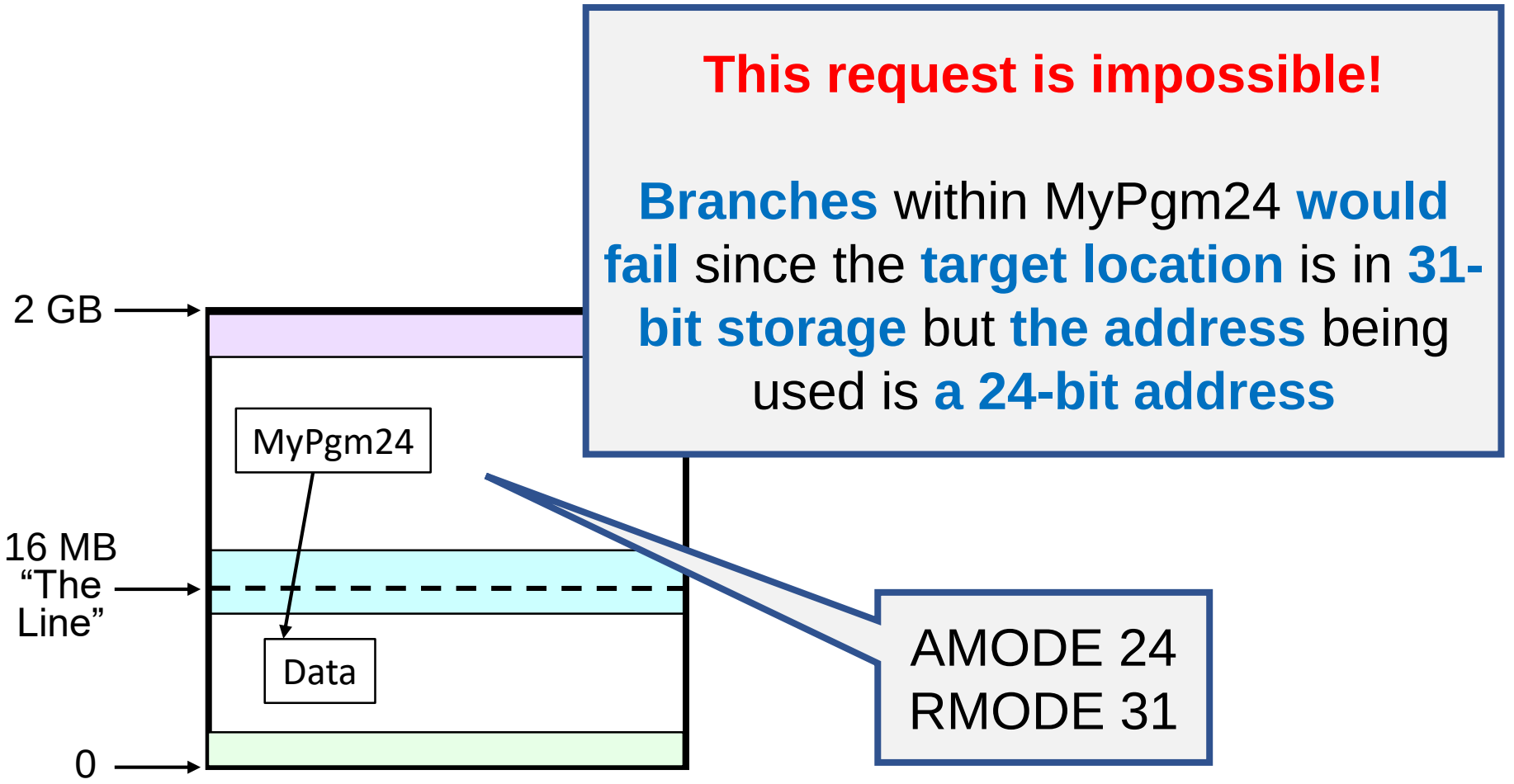
Addressing Modes



This request is also possible

AMODE 31
RMODE 24

Addressing Modes



Addressing Modes

Memory addresses consisted of **all 64-bits** of a doubleword
FFFFFFFF FFFFFFFF

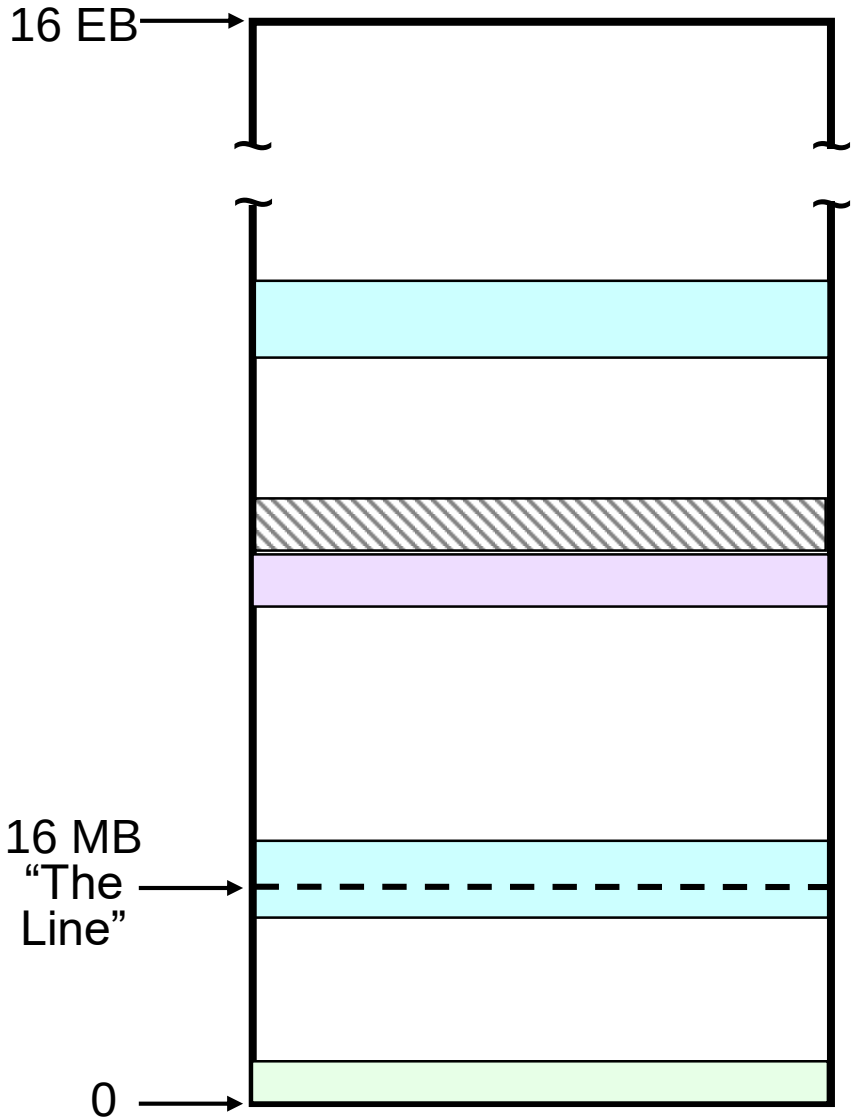
Addressing Modes

IBM z900
vintage 2000

Memory addresses consisted of **all 64-bits** of a doubleword
FFFFFFFF FFFFFFFF

Memory Addressability **range**:
0 – 18,446,744,073,709,551,615 bytes
or
18,446,744,073,709,551,616 bytes
or
16 EB

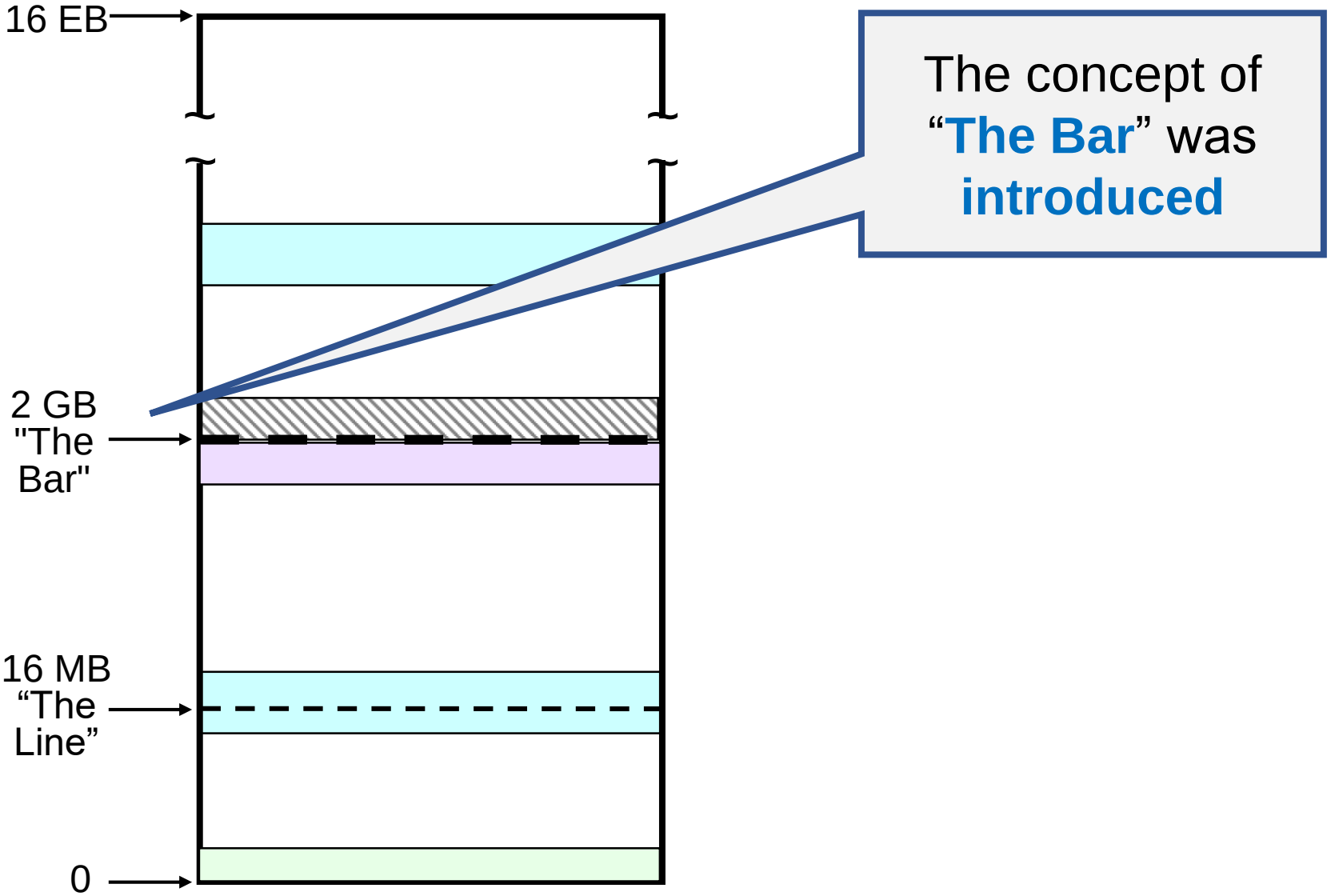
Addressing Modes



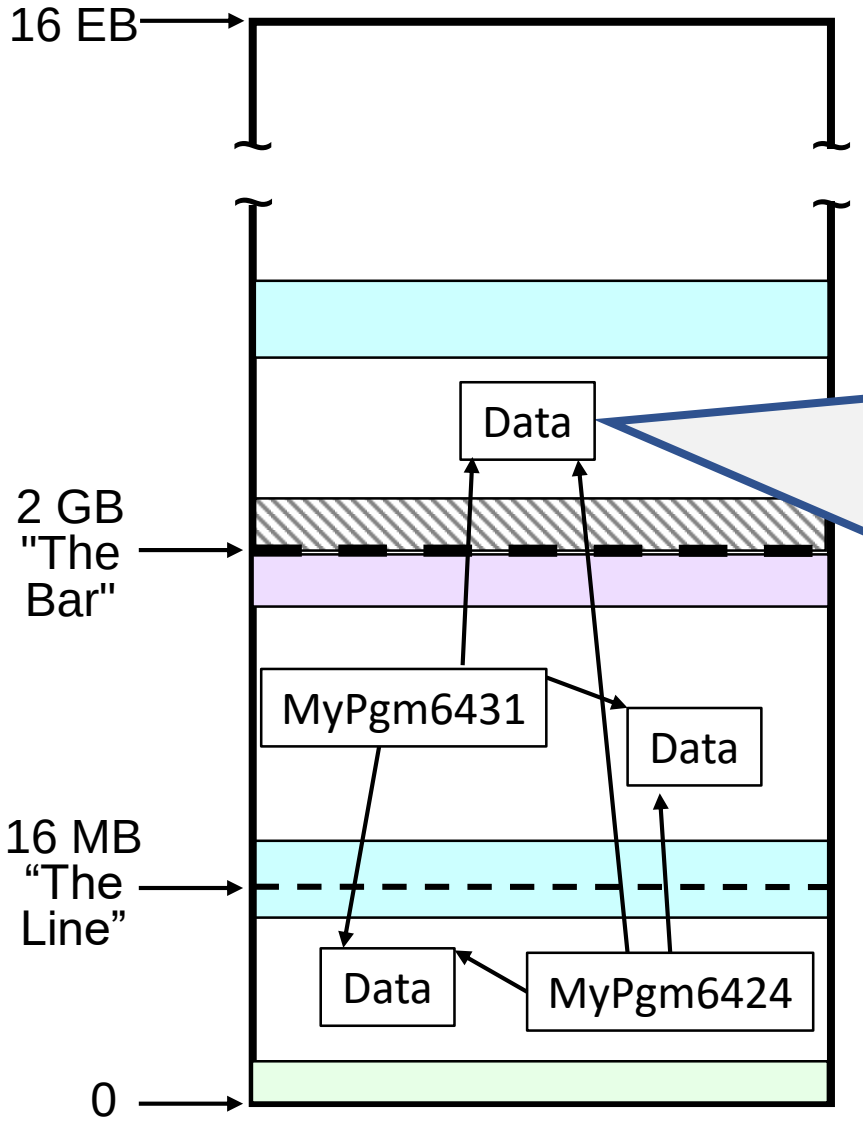
An address space **size** is **limited to** the amount of **memory that can be addressed**

In 2000, an address space is **limited to 16 EB**

Addressing Modes

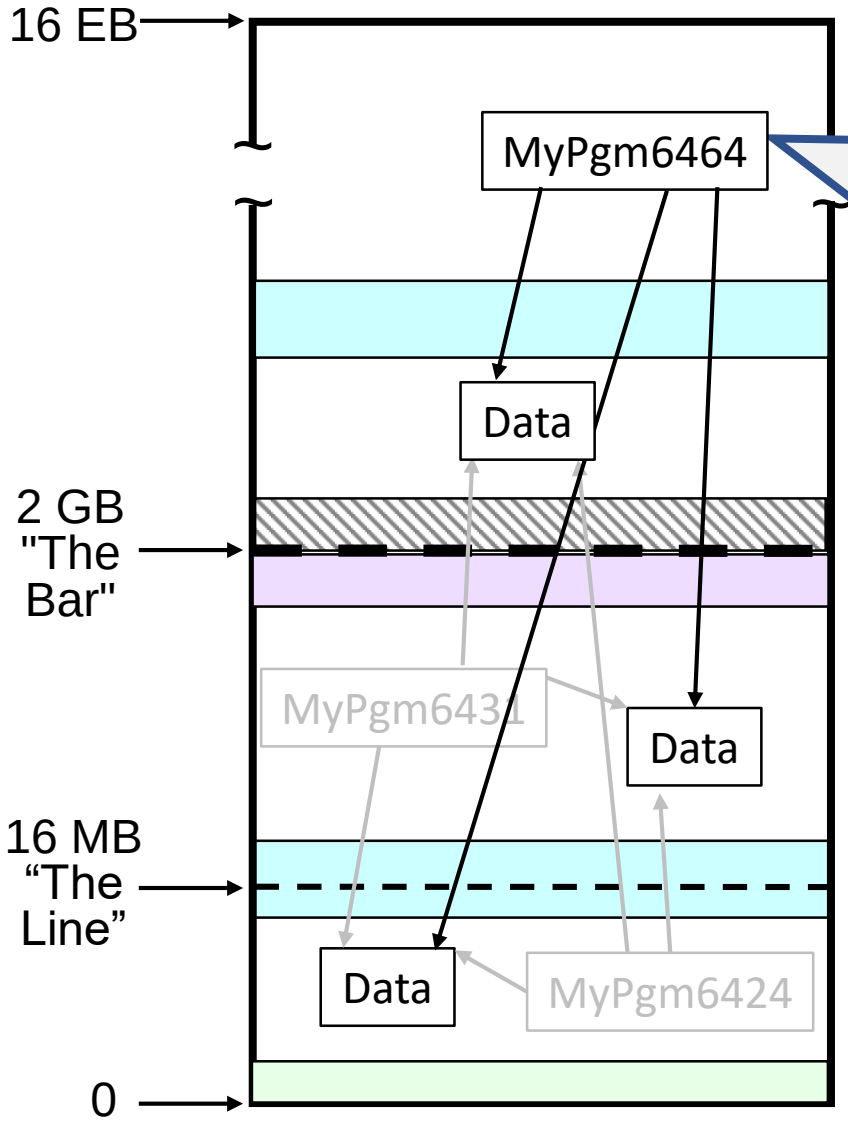


Addressing Modes



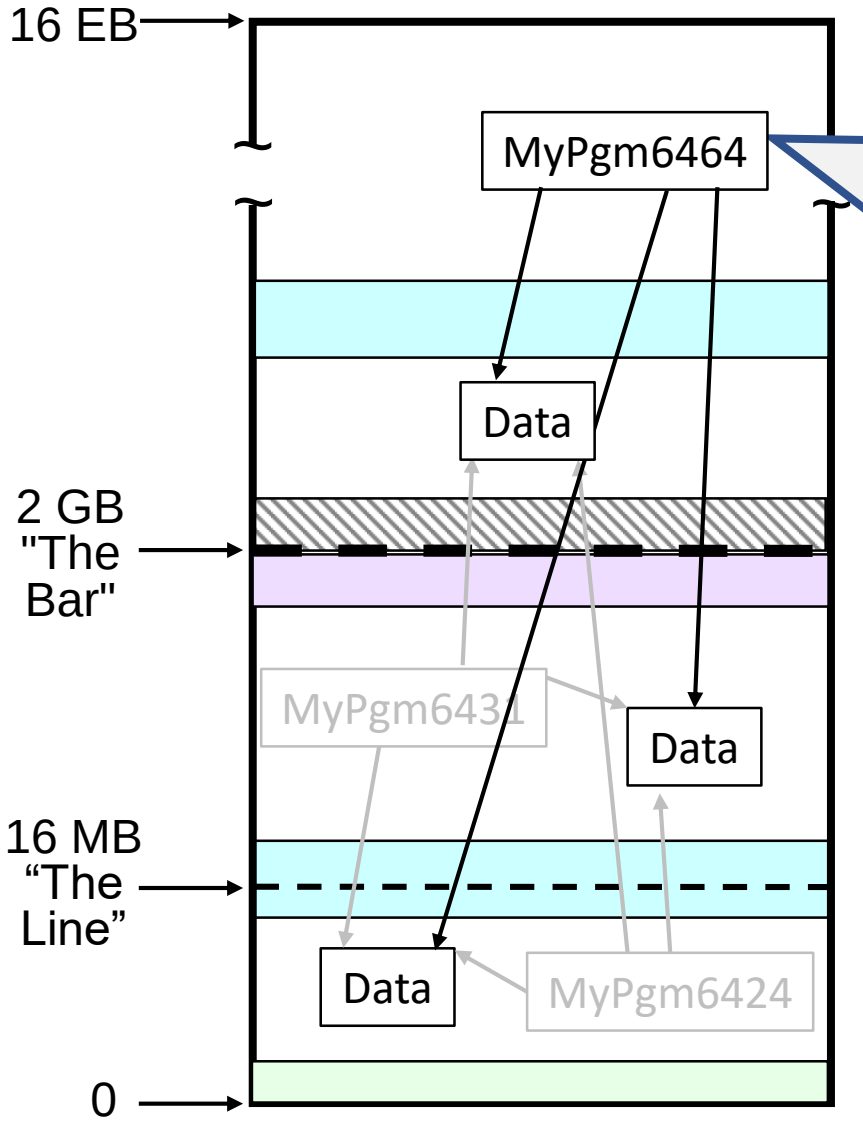
AMODE 64 is introduced and **data** can now be accessed “**Above the Bar**”

Addressing Modes



With z/OS V2R3 (2017),
z/OS supports
RMODE 64
With restrictions

Addressing Modes



With z/OS V2R3 (2017),
z/OS supports
RMODE 64
With restrictions

Programs will **be loaded**
"Above the Bar"

Check your Knowledge

Check Your Knowledge

- **(T or F)** Each user gets an address space containing the same range of Addresses?

Check Your Knowledge

- **(T or F)** Each user gets an address space containing the same range of Addresses? **TRUE**
- **(T or F)** A module's RMODE value can be greater than its AMODE value.

Check Your Knowledge

- **(T or F)** Each user gets an address space containing the same range of Addresses? **TRUE**
- **(T or F)** A module's RMODE value can be greater than its AMODE value. **FALSE**
- What is the address range for area below the bar?

Check Your Knowledge

- **(T or F)** Each user gets an address space containing the same range of Addresses? **TRUE**
- **(T or F)** A module's RMODE value can be greater than its AMODE value. **FALSE**
- What is the address range for area below the bar? - **0 - 2GB**

Check Your Knowledge

- **(T or F)** Each user gets an address space containing the same range of Addresses? **TRUE**
- **(T or F)** A module's RMODE value can be greater than its AMODE value. **FALSE**
- What is the address range for area below the bar? - **0 - 2GB**
- To address data that resides "above the bar", what mode is required?

Check Your Knowledge

- **(T or F)** Each user gets an address space containing the same range of Addresses? **TRUE**
- **(T or F)** A module's RMODE value can be greater than its AMODE value. **FALSE**
- What is the address range for area below the bar? - **0 - 2GB**
- To address data that resides "above the bar", what mode is required? – **AMODE 64**

Storage

Storage

- Programs are written to **utilize** a given amount of **memory**.

Storage

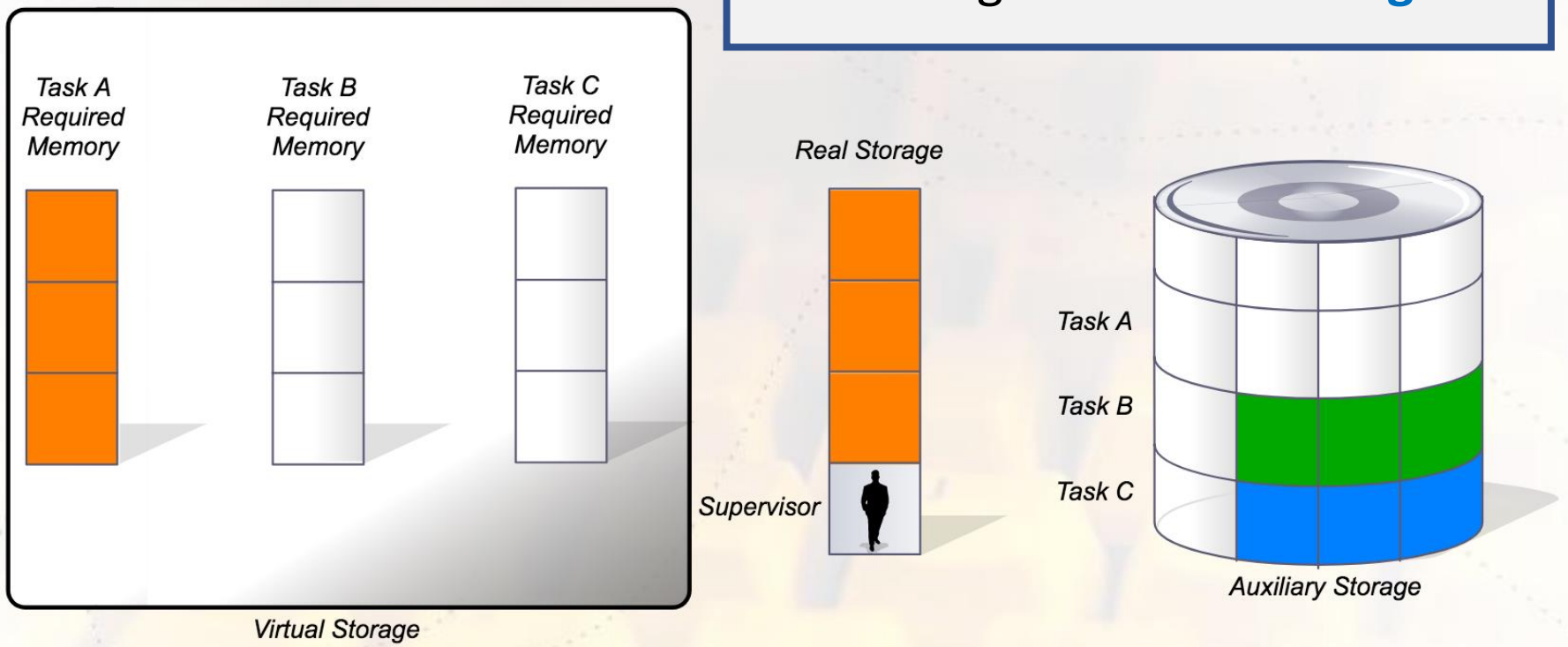
- Programs are written to **utilize** a given amount of **memory**.
- The amount of **memory** required by **all** the running **tasks** is usually much **greater** than the amount of **real storage available**.

Storage

- Programs are written to **utilize** a given amount of **memory**.
- The amount of **memory** required by **all** the running **tasks** is usually much **greater** than the amount of **real storage available**.
- This is facilitated by the use of **virtual storage**.

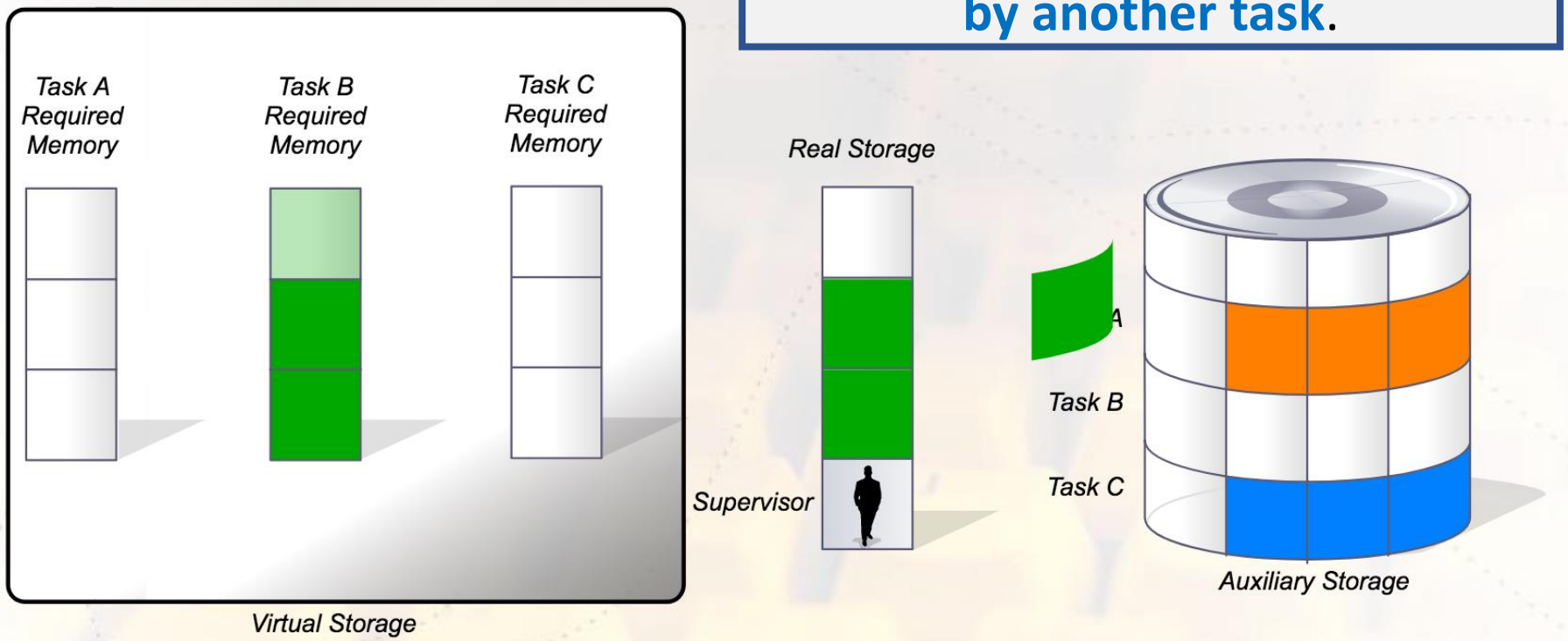
Storage

When a task is **active**, its memory is brought into **real storage**.



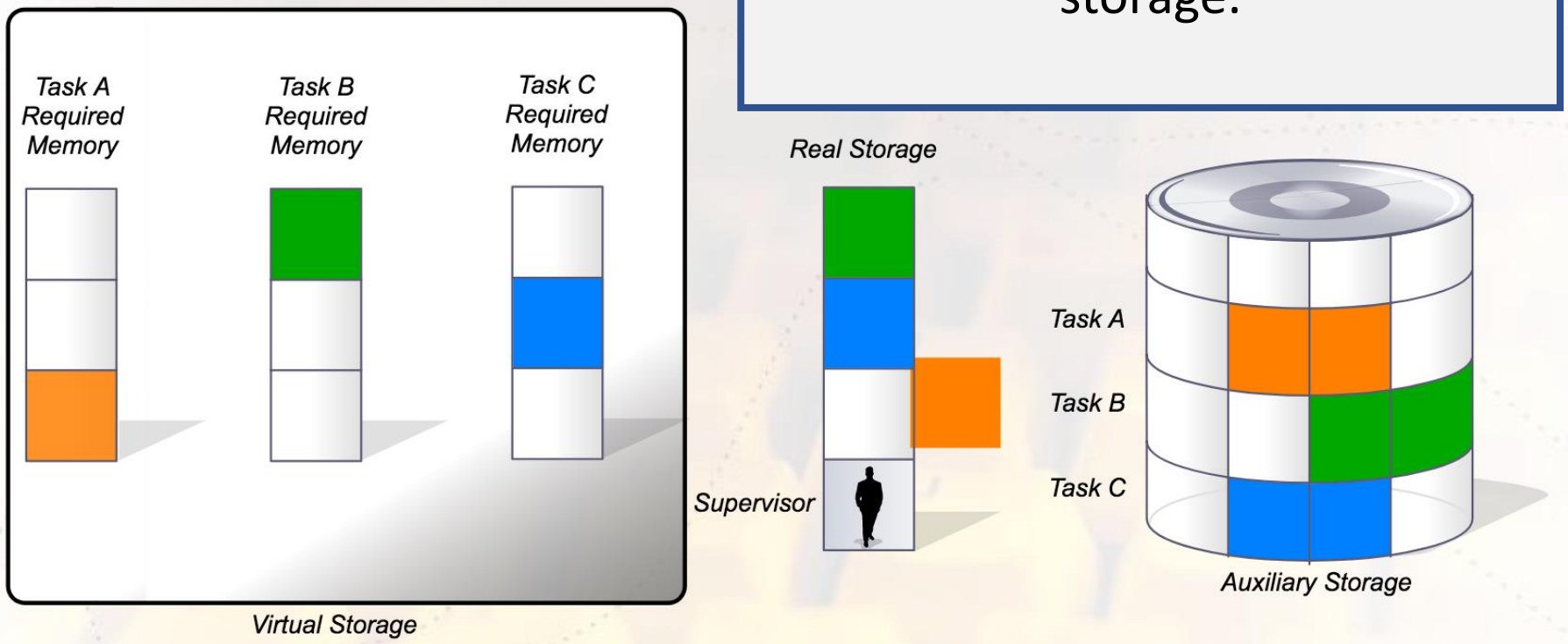
Storage

When a task is **not active**, the **real storage** that it used can be **used by another task**.



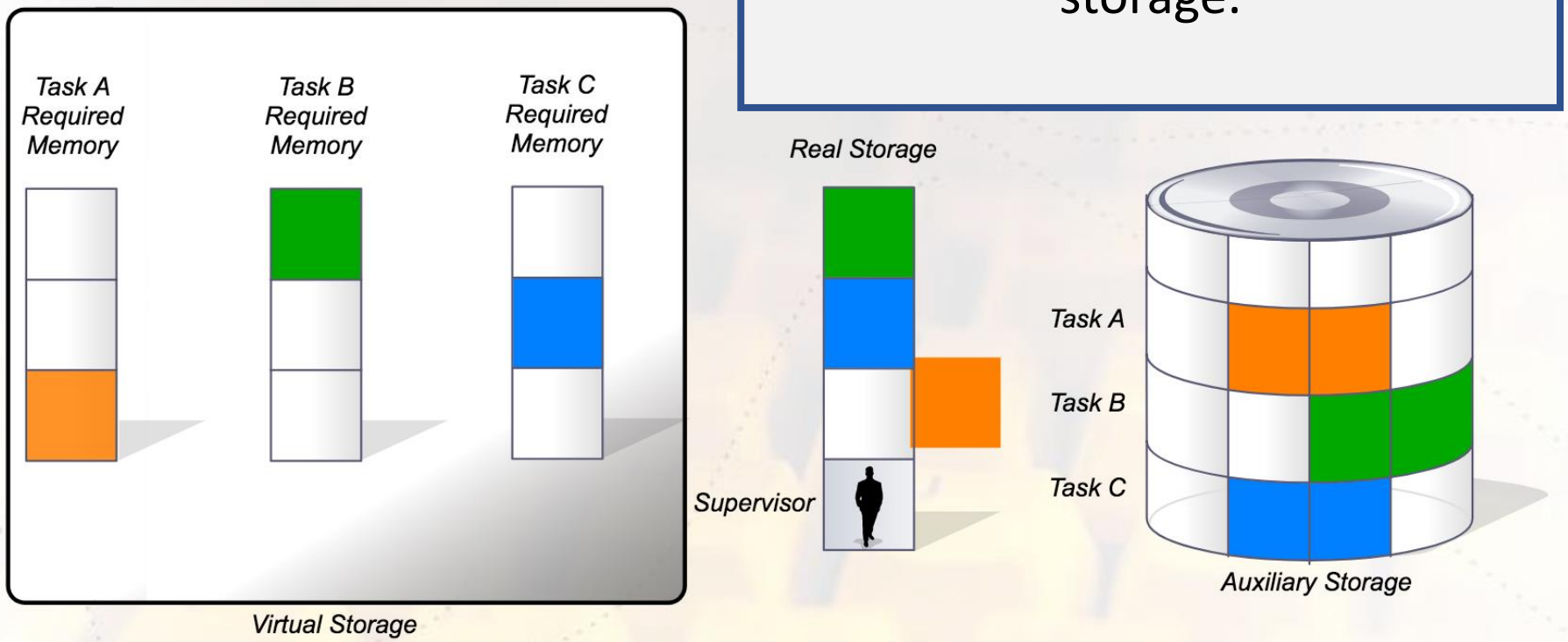
Storage

When the task is **reactivated**, its memory is **restored** from auxiliary storage.



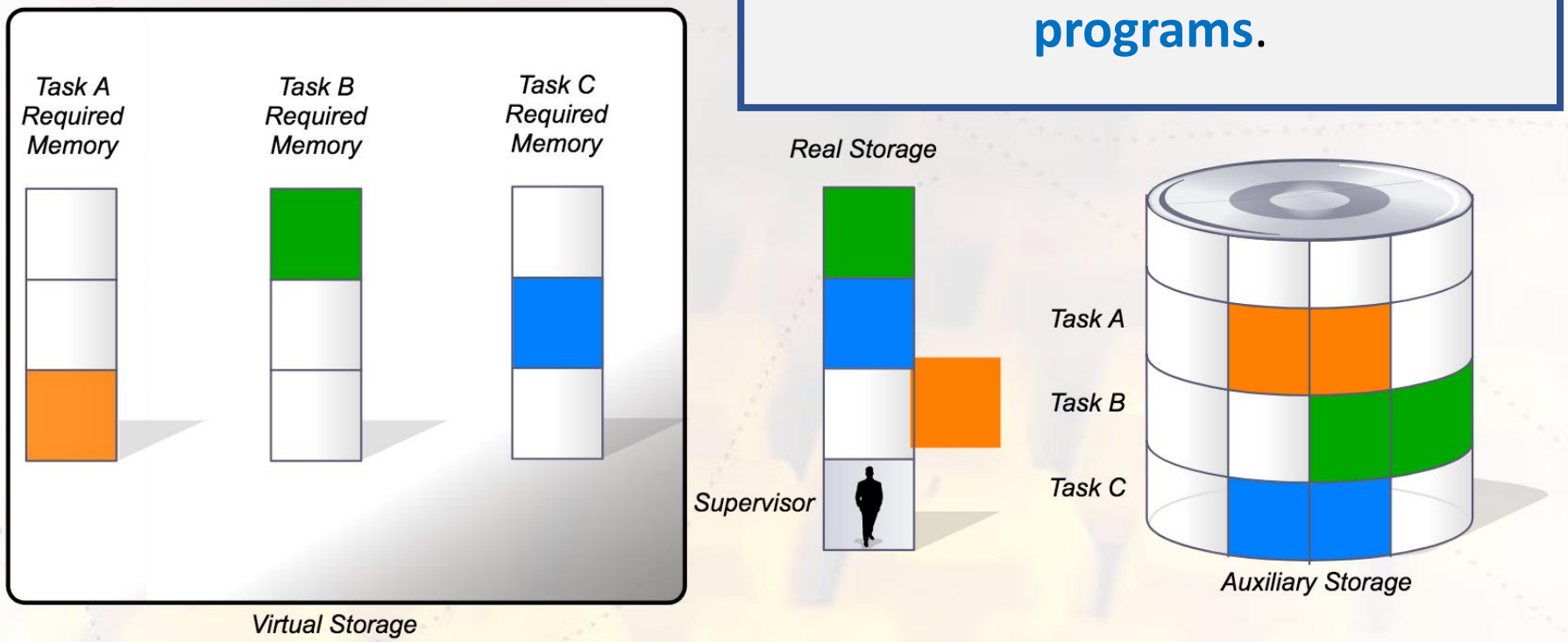
Storage

Only the virtual storage that is being **used** needs to be in real storage.



Storage

The rest can **stay** in **auxiliary storage**, enabling the **remaining** real storage to be **used for other programs**.



Storage

- **Virtual** storage is divided into **4KB** pieces called **pages**.

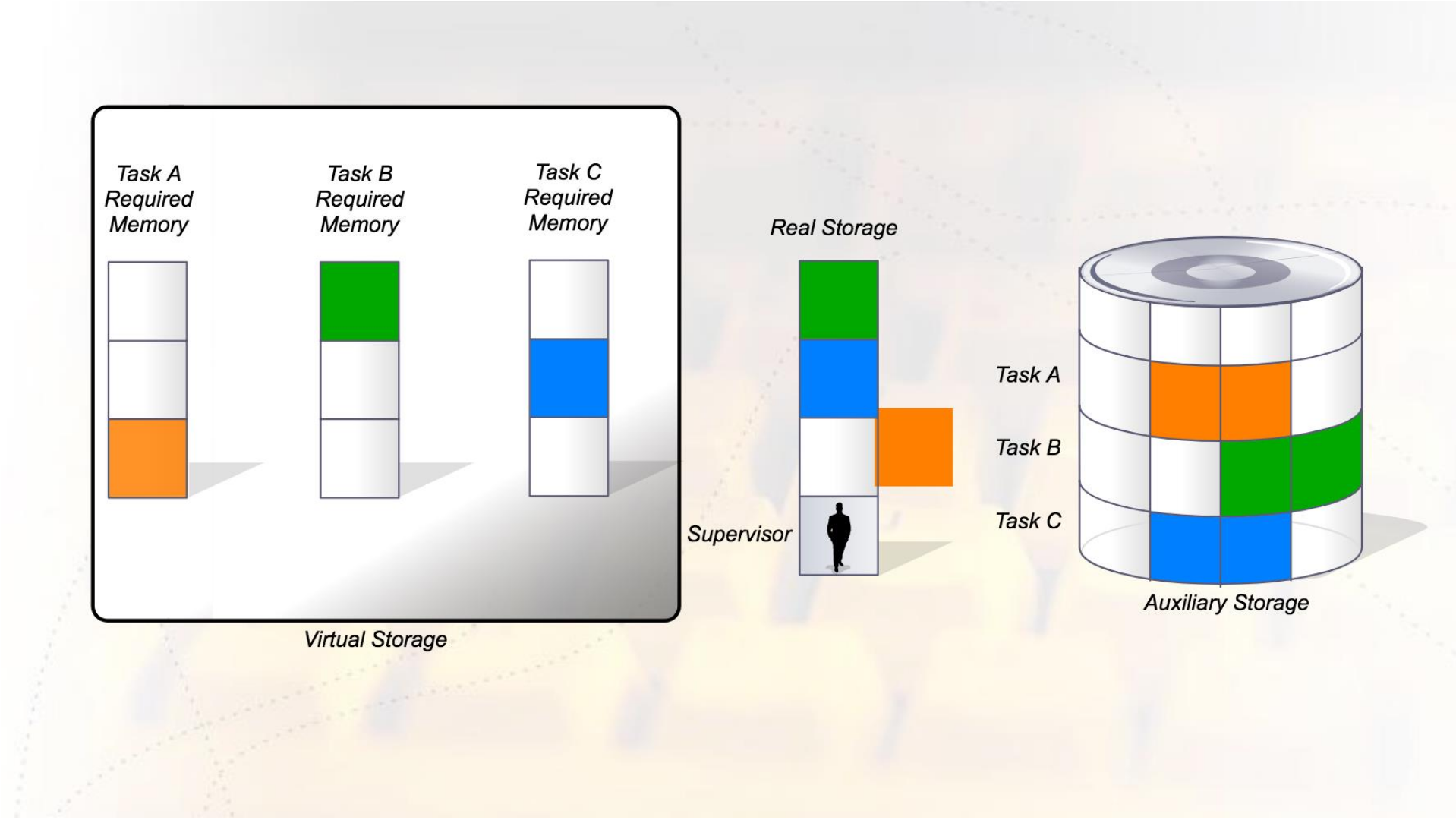
Storage

- **Virtual** storage is divided into **4KB** pieces called **pages**.
- **Real** storage is divided into **4KB** pieces called **frames**.

Storage

- **Virtual** storage is divided into **4KB** pieces called **pages**.
- **Real** storage is divided into **4KB** pieces called **frames**.
- **Auxiliary** storage is divided into **4KB** pieces called **slots**.

Storage


Task A Required Memory
Task B Required Memory
Task C Required Memory
Virtual Storage
Real Storage
Supervisor
Task A
Task B
Task C
Auxiliary Storage

Dynamic Address Translation (DAT)

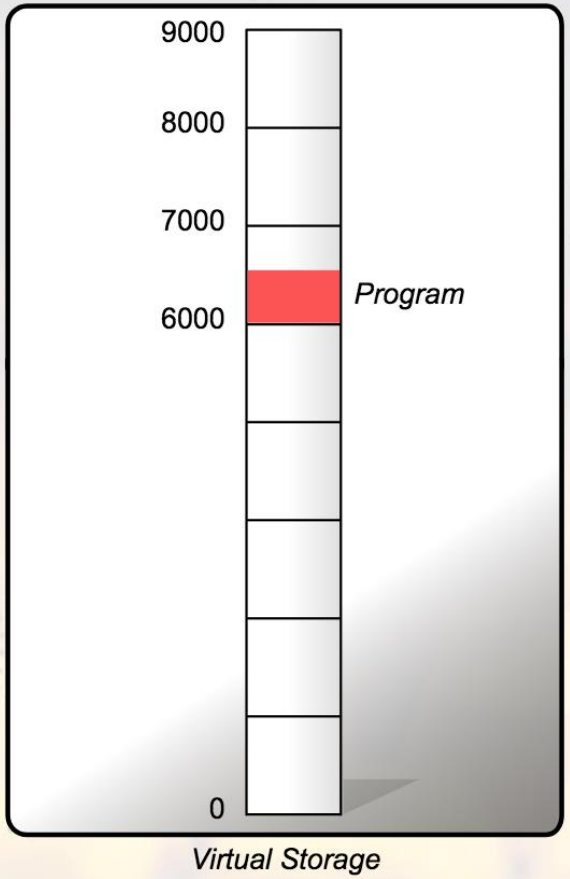
Address Translation

- When a program is **running**, the processor must be able to **locate** its virtual **pages** in real storage.

Address Translation

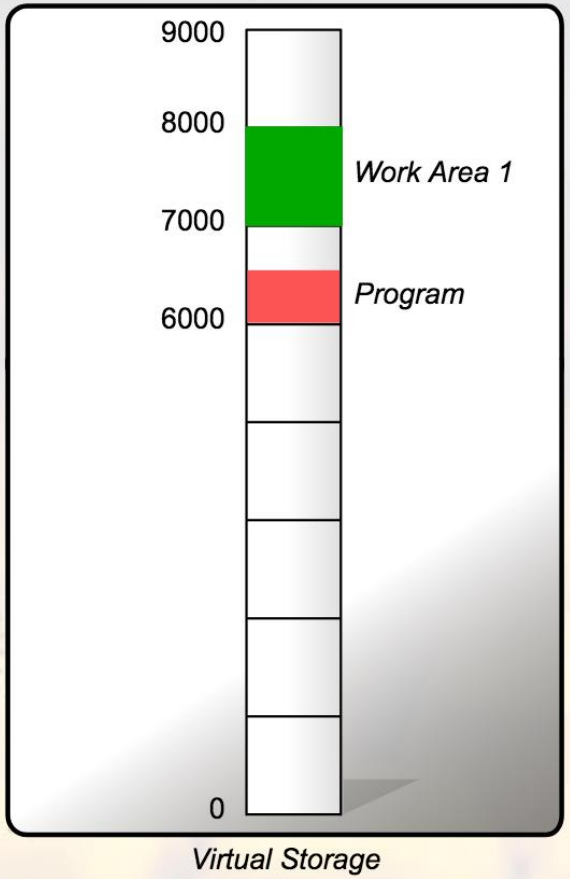
- When a program is **running**, the processor must be able to **locate** its virtual **pages** in real storage.
- **Address translation** is the process that enables a processor to **convert virtual addresses into real addresses**.

Address Translation



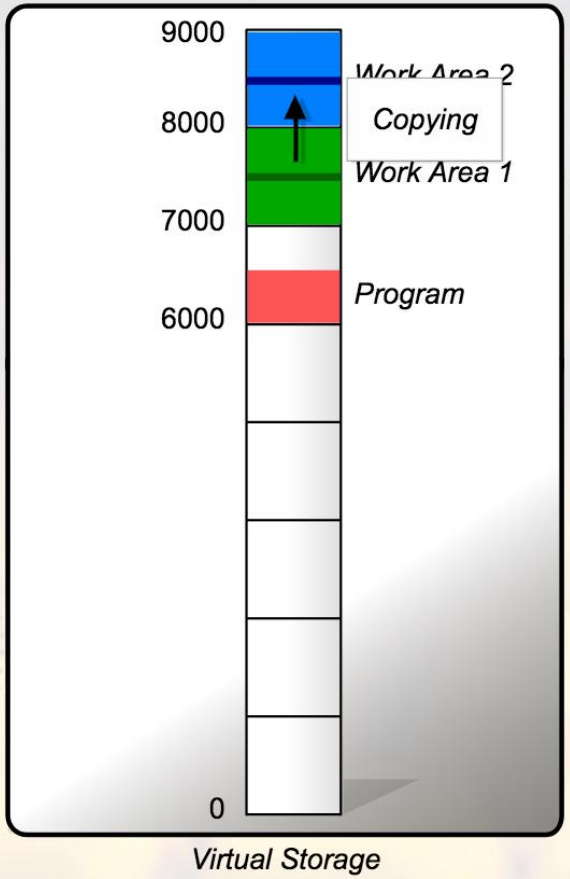
A program is **loaded** at address 6000.

Address Translation



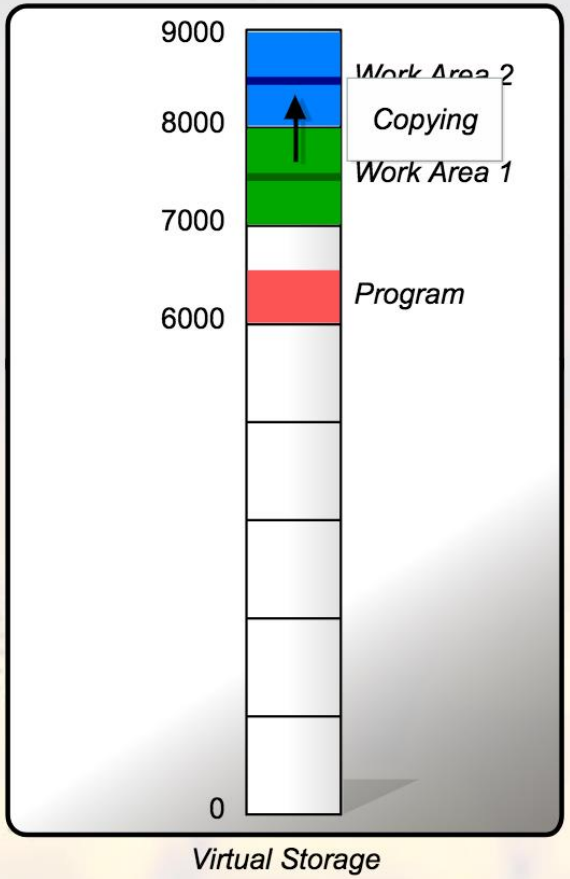
A 4KB work area is **obtained** at address 7000.

Address Translation



256 bytes of data are **copied** from address 7800 to 8800.

Address Translation



The hardware feature that makes all this possible is called the **dynamic address translation** (DAT).

GUIDE SHARE EUROPE UK REGION

Intro to z/OS: Part 2
Wednesday, 10:15am in Suzuka (Here)

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/af

- This session is AF

1. What is your conference registration number?

This is the three digit number on the bottom of your delegate badge

2. Was the length of this presention correct?

1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 2 3 4 5 6 7 8 9

3. Did this presention meet your requirements?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

IBM Systems Worldwide Client Experience Centers

IBM Systems Worldwide Client Experience Centers *maximize IBM Systems competitive advantage in the Cloud and Cognitive era by providing access to world class technical experts and infrastructure services to assist Clients with the transformation of their IT implementations..*

9 Worldwide Locations (* also Infrastructure Hubs):

Austin TX , *Poughkeepsie NY, Rochester MN, Tucson AZ, *Beijing CHINA, Boeblingen GERMANY, Guadalajara MEXICO,*Montpellier FRANCE, Tokyo JAPAN

Client Experience	***Architecture & Design***	***Infrastructure Solutions***	***Content***
Tailored, in-depth technology Innovation Exchange Events Relationship building Demonstrations Meetups Solution workshops Remote options	Advise clients, "Art of the Possible" Discovery & Design Workshops, Consulting, Showcases, Reference Architectures, Co-Creation of assets	Benchmarks, MVP & Proof of Technology "Test Drives" Demonstrations Infrastructure Services Certify ISV solutions Hosting Cloud Environment	Content Development IBM Redbooks Training Courses Video courses "Test Drives" Demonstrations
(Inbound & Outbound)	(Inbound & Outbound)	(Inbound to Centers)	

NEW: *Co-Creation Lab; CEC Cloud; RedHat Center of Competency*

For further information, please contact the Centers via email at:
ccenter@us.ibm.com

© 2019 IBM Corporation


z/OS


IBM