

z/VM



# Glossary

*Version 6 Release 3*

**Note:**

Before using this information and the product it supports, read the information in "Notices" on page 139.

This edition applies to version 6, release 3, modification 0 of IBM z/VM (product number 5741-A07) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GC24-6195-02.

© **Copyright IBM Corporation 1991, 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About this document</b> . . . . .	<b>v</b>
Intended audience . . . . .	v
Where to find more information . . . . .	v
Links to other documents and websites . . . . .	v
<b>How to send your comments to IBM</b> . . . . .	<b>vii</b>
<b>Summary of changes</b> . . . . .	<b>ix</b>
GC24-6195-04, z/VM Version 6 Release 3 (updated edition) . . . . .	ix
GC24-6195-03, z/VM Version 6 Release 3 . . . . .	ix
GC24-6195-02, z/VM Version 6 Release 2 . . . . .	ix
<b>Glossary</b> . . . . .	<b>1</b>
<b>Notices</b> . . . . .	<b>139</b>
Privacy policy considerations . . . . .	141
Trademarks . . . . .	141
<b>Bibliography</b> . . . . .	<b>143</b>
Where to get z/VM information . . . . .	143
z/VM base library . . . . .	143
z/VM facilities and features . . . . .	144
Prerequisite products . . . . .	145



---

## About this document

This document defines terms used in the IBM® z/VM® product documentation. It includes definitions for both general computing terms and VM-specific terms.

---

## Intended audience

This information is intended for all z/VM users and other persons who wish to become familiar with z/VM terms.

---

## Where to find more information

You can obtain more information about z/VM from the books listed in the “Bibliography” on page 143.

## Links to other documents and websites

The PDF version of this document contains links to other documents and websites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a website works only if you have access to the Internet. A document link is to a specific edition. If a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.



---

## How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to [mhvrcfs@us.ibm.com](mailto:mhvrcfs@us.ibm.com).
2. Go to IBM z/VM Reader's Comments ([www.ibm.com/systems/z/os/zvm/zvmforms/webqs.html](http://www.ibm.com/systems/z/os/zvm/zvmforms/webqs.html)).
3. Mail the comments to the following address:  
IBM Corporation  
Attention: MHVRCFS Reader Comments  
Department H6MA, Building 707  
2455 South Road  
Poughkeepsie, NY 12601-5400  
U.S.A.
4. Fax the comments to us as follows:  
From the United States and Canada: 1+845+432-9405  
From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:  
**z/VM V6.3 Glossary**  
**GC24-6195-04**
- The topic name or page number related to your comment
- The text of your comment

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

IBM or any other organizations will use the personal information that you supply only to contact you about the issues that you submit to IBM.

### If you have a technical problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- See IBM: z/VM Service Resources ([www.ibm.com/vm/service/](http://www.ibm.com/vm/service/)).
- Go to IBM Support Portal ([www.ibm.com/support/entry/portal/Overview/](http://www.ibm.com/support/entry/portal/Overview/)).



---

## Summary of changes

This document contains terminology, maintenance, and editorial changes. Technical changes are indicated by a vertical line to the left of the change. Some product changes might be provided through service and might be available for some prior releases.

---

### GC24-6195-04, z/VM Version 6 Release 3 (updated edition)

New terms include:

- CPU pool
- environment information interface
- IBM zEnterprise Data Compression (zEDC)
- OpenStack
- Remote Direct Memory Access over Converged Internet (RoCE)
- Soft Fence

---

### GC24-6195-03, z/VM Version 6 Release 3

New terms include:

- dispatch vector
- entitlement
- Extreme Cloud Administration Toolkit (xCAT)
- global aging list
- horizontal polarization mode
- HiperSockets™ bridge port
- logical processor
- mirrored DASD
- multiple subchannel set (MSS)
- parked
- Transport Layer Security (TLS)
- unparked
- upgrade installation
- vertical-high (Vh) processor
- vertical-low (Vl) processor
- vertical-medium (Vm) processor
- vertical polarization mode
- Virtual Edge Port Aggregator (VEPA)
- z/VM® HiperDispatch

---

### GC24-6195-02, z/VM Version 6 Release 2

New terms include:

- channel path ID virtualization
- ESA/390 CMS
- global minidisk
- identity entry
- live guest relocation
- local minidisk
- multiconfiguration virtual machine definition
- non-SSI source directory
- persistent data record (PDR)
- relocation domain

- single-configuration virtual machine definition
- SSI-enabled source directory
- SSI-ready source directory
- subconfiguration entry
- virtual architecture level
- z/Architecture<sup>®</sup> CMS (z/CMS)
- z/VM single system image (SSI) cluster

---

# Glossary

This glossary defines technical terms used in the z/VM product documentation. It is not a comprehensive resource of all IBM or z/VM computing terms.

The terms in this glossary are alphabetized using the letter-by-letter method. Blanks and hyphens in multiple-word terms are ignored. Slashes (/) and other special characters used in terms are alphabetized ahead of the alphabetic characters. Numerals included in terms are alphabetized after the alphabetic characters.

The following cross-references are used in this glossary:

**See** *term*

This refers the reader to a preferred term, or to a term that is the expanded form of an abbreviation or acronym.

**See also** *term*

This refers the reader to a term with a similar meaning, such as a broader term, a narrower term, or a contrasting term.

## Numerics

**3215** Refers to an IBM family of line mode displays.

**3270** Refers to an IBM family of display devices and printers.

**3270 emulation**

The use of software that enables a client to emulate an IBM 3270 display station or printer, and to use the functions of an IBM host system.

## A

**abbreviation**

A valid short form of a command name, operand, or option that is not a truncation of the word. For example, MSG is an abbreviation of the MESSAGE command. See also minimum truncation.

**abend** See abnormal end of task.

**abend dump**

The contents of main storage, or part of main storage, and other items such as registers, written to an external medium for debugging an error condition that resulted in the termination of a task before its regular completion. See also hard abend dump, soft abend dump, and snap dump.

**abnormal end of task (abend)**

The termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing. See also abnormal termination.

**abnormal termination**

- (1) The cessation of processing prior to planned termination.
- (2) A system failure or operator action that causes a job to end unsuccessfully.

**absolute address**

An address used for a storage access without any transformations formed on it.

**absolute path name**

A BFS path name that begins with the root directory. The absolute path name must begin with a slash (/), which indicates that the path begins at the root. See also [fully qualified path name](#) and [relative path name](#).

**absolute storage**

Storage consisting of byte locations sequenced according to their absolute addresses. Absolute storage is synonymous with real storage except for the effects of prefixing. Prefixing provides the ability to assign the range of real addresses 0 - 8191 to a different block in absolute storage for each processor, thus permitting more than one processor sharing main storage to operate concurrently with a minimum of interference, especially in the processing of interruptions.

**access control interface (ACI)**

A group of CP modules that mediate between CP and an external security manager (ESM) to handle authorization checking. z/VM supplies stub modules that are replaced when an ESM is installed.

**access list**

See [host access list](#).

**access method services (AMS)**

The facility used to define and reproduce VSAM key-sequenced data sets (KSDS). In CMS, AMS is invoked through the AMSERV command.

**access mode**

See [file access mode](#).

**access register (AR)**

A 32-bit register that contains an access-list-entry token (ALET) identifying an address space to be used for data references when in access-register mode. See also [access-list-entry token](#) and [access-register mode](#).

**access security**

Pertaining to information that a target LU and target transaction program use to verify whether a source program is authorized to make a connection. This information consists of a user ID and possibly a password.

**access-list entry (ALE)**

See [host access-list entry](#).

**access-list-entry token (ALET)**

A 32-bit value that is an indirect specification of an address space. When a program is operating in access-register mode, an ALET contained in an access register designates the address space containing an instruction operand.

**access-register mode**

An address translation mode in which the contents of the access registers are used to determine the address space containing a storage operand. The SET ADDRESS SPACE CONTROL and SET ADDRESS SPACE CONTROL FAST instructions can be used to change the PSW to enter and exit access-register mode. See also [primary-space mode](#).

**ACI** See [access control interface](#).

**active gateway**

A gateway that is treated like a network interface in that it is expected to exchange routing information, and if it does not do so for a period of time, the route associated with the gateway is deleted.

**active open**

The state of a connection that is actively seeking a service. See also [passive open](#).

**active record**

A record added or appended to a file currently in use.

**active work unit**

A work unit that has uncommitted work associated with it. A request was made on the work unit (other than an atomic request) and no commit or rollback has occurred.

**ADCON**

See [address constant](#).

**address constant (ADCON)**

A field containing an address, a length, or an offset, which is used in calculating storage addresses.

**address mask**

See [subnet mask](#).

**address space**

(1) A collection of bytes that are allocated, and in many ways managed, as a single entity by CP. Each byte within an address space is identified by a unique address. An address space represents an extent of storage available to a program. See also [host-primary address space](#).

(2) The memory locations that can be referred to by a process.

**addressing-capability exception**

A program exception that arises as a result of an attempt to access an address space for which access permission has been revoked by the owner of the address space.

**address-space-identification token (ASIT)**

An 8-byte token that uniquely identifies a particular address space allocated by CP. The ASIT is a unique system-wide identifier. After an ASIT value has been assigned to an address space, that ASIT value is not used again within the scope of a z/VM system IPL.

**Advanced Function Printing (AFP)**

The ability of programs to print all-points-addressable text and images.

**Advanced Function Printing Data Stream (AFPDS)**

A method of representing image data in a file that is destined for printing on a Print Services Facility™ (PSF)-controlled printer.

**advanced program-to-program communication (APPC)**

The general facility characterizing the LU 6.2 architecture and its various implementations in products.

**Advanced Program-to-Program Communication/VM (APPC/VM)**

An API for communicating between two virtual machines that is mappable to the LU 6.2 APPC interface and based on IUCV functions. Along with the TASK virtual machine, AVS virtual machine, and VTAM®, APPC/VM provides this communication within a single system, throughout a collection of systems, and throughout an SNA network. See also [APPC/VM VTAM Support](#).

**AFP** See [Advanced Function Printing](#).

**AFPDS**

See [Advanced Function Printing Data Stream](#).

**agent** (1) Software that acts on behalf of a user as it performs tasks within an application program. An agent may run on both the client and the server.

(2) In Coordinated Resource Recovery (CRR) synchronization point processing, the role of the CRR sync point manager (SPM) when sync point requests are received from an initiator that is a partner in a protected conversation.

(3) A task for a file pool server. The SFS file pool server and the CRR recovery server provide support for multitasking several agents.

(4) As defined in the SNMP architecture, an agent, or an SNMP server is responsible for performing the network management functions requested by the network management stations.

**AL** ANSI labeled tape without user label processing.

**ALE** See host access-list entry.

**ALEN-translation exception**

A program interruption that occurs when a program attempts to use a correctly formed but currently unassigned ALET. The ALET might have been associated with a valid host access-list entry that was subsequently deallocated.

**alert** A message or other indication that signals an event or an impending event.

**ALET** See access-list-entry token.

**alias** (1) A pointer to a file pool base file. An alias refers to the data in the base file. Creating an alias does not move or duplicate data. An alias does not need to be in the same directory as the base file, but does need to be in the same file pool as the base file. See also base file, erased alias, revoked alias, and unresolved alias.

(2) An alternative name for a shell command, which can be defined with options different than those for the command itself.

(3) In Parallel Access Volumes architecture, a subchannel that maps to a base subchannel for a DASD, allowing multiple concurrent I/O operations to that DASD.

**allocate**

To specify how a space (cylinder or page) on a DASD volume is to be used.

**allocation tree**

A conceptual structure showing the nodes that are allocating protected conversations within a CRR coordinated transaction.

**alternate path support**

The selection of a path to a device from any of the available paths, even though the primary path is busy. The selection is made in response to an I/O request for a device.

**American Standard Code for Information Interchange (ASCII)**

A standard code used for information exchange among data processing systems, data communication systems, and associated equipment. ASCII uses a coded character set consisting of 7-bit coded characters.

**AMS** See access method services.

**anchor block**

An area in user storage that contains space for 16 anchor slots. It also includes space for the register save area used by the CMS anchor facility. The anchor block storage persists after an abend occurs.

**anchor identifier**

A 3-character identifier that uniquely identifies the application program when using the ANCHOR macro to set, query, or clear an anchor word. The anchor identifier must be assigned by IBM.

**anchor slot**

The area in the anchor block in which an anchor identifier and anchor word are saved.

**anchor word**

An area of storage in which an application program with critical performance needs can save the address of its data. This avoids the overhead of obtaining dynamic storage each time the program is invoked. The anchor word persists between calls to the program and persists after an abend occurs.

**APAR** See authorized program analysis report.

**APAR number**

An identifier that IBM assigns to an APAR.

**APPC** See advanced program-to-program communication.

**APPC link**

A logical TSAF link that is physically controlled by VTAM and is established using the TSAF VTAM line driver.

**APPC/VM**

See Advanced Program-to-Program Communication/VM.

**APPC/VM VTAM Support (AVS)**

A component of z/VM that allows application programs using APPC/VM or CPI Communications to communicate with programs anywhere in an SNA network. AVS transforms APPC/VM into APPC/VTAM protocol.

**apply** In the installation or service of a product, to generate an auxiliary control file structure from a PTF.

**apply disk**

In VMSES/E, a minidisk or file pool directory containing the files that describe the maintenance levels: apply status table, auxiliary control files, version vector tables, select data file, and build status table.

**apply ID**

A 1- to 8-character alphanumeric identifier that is used to name the software inventory files created during apply processing. The user can change this value to define different maintenance levels.

**apply list**

A file listing PTFs applied to a product or component.

**apply message log**

The file in which the VMFAPPLY command writes status and error messages during apply processing.

**apply processing**

See apply.

**apply status table**

See system-level apply status table and service-level apply status table.

**apply string**

In VMSES/E, the set of apply disks.

**AR** See access register.

**AR mode**

See access-register mode.

**arbitrary character**

In XEDIT, a character, usually "\$", used between character strings in a target definition to ignore all intervening characters.

**architected segment**

A 1 MB portion of real storage defined by ESA/390, ESA/XC, and z/Architecture.

**ASCII** See American Standard Code for Information Interchange.

**ASIT** See address-space-identification token.

**asynchronous**

Pertaining to events that are not synchronized in time or do not occur in regular or predictable time intervals. See also synchronous.

**asynchronous communication**

A method of communication in a conversation that lets a program issue functions on other conversations while waiting for a particular function to complete. See also [synchronous communication](#).

**atomic request**

A file pool command or program function (CSL routine) that completes execution immediately and does not affect work in other resources (such as other file pools) on the work unit. Any file pool updates caused by an atomic request are committed (or rolled back) when the file pool server finishes processing the request. Atomic requests cannot be issued to a file pool that has uncommitted work for the specified (or default) work unit.

**attention interruption**

An I/O interruption caused by a terminal user pressing the attention key (or equivalent).

**attention key**

In z/VM, a terminal key that, when pressed, causes an interruption that results in a VM or CP read being presented to the virtual console.

**attention signal**

See [attention interruption](#).

**attention symbol**

On printing terminals under CP, the character that indicates each time the attention key is pressed. This is always the exclamation point (!) character.

**ATTN key**

See [attention key](#).

**AUL** ANSI labeled tape with user label processing.

**authority**

In a file pool server, the permission to access a file or directory. A user can have read authority or write authority (which includes read authority). A user can also have file pool administration authority, which is the highest level of authority in a file pool.

**authorized application**

A GCS application, started with the OSRUN command, that operates in supervisor state and can use privileged functions. See also [ordinary application](#).

**authorized program**

See [privileged program](#).

**authorized program analysis report (APAR)**

An official request to IBM to look into a suspected problem with IBM code or documentation. The APAR describes the problem, giving conditions of failure, error messages, abend codes, or other identifiers. It also contains a problem summary and resolution when applicable. See also [program temporary fix](#).

**authorized user ID**

In GCS, a user ID that provides access to the GCS supervisor, supervisor state, and (at times) certain restricted CP commands. This access is provided by including the user ID on a list of authorized user IDs compiled with the GCS GROUP command. The virtual machine associated with an authorized user ID is an *authorized* virtual machine, and programs running in that machine are *authorized* applications.

**authorized virtual machine**

A GCS virtual machine associated with an authorized user ID.

**automatic logon**

A process by which a user's virtual machine is initiated by other than the user of that virtual machine. For example, the primary z/VM system operator's virtual machine is activated

automatically during z/VM initialization. A privileged user can enter the AUTOLOG or XAUTOLOG command to start some other virtual machine.

**automatic software re-IPL**

The process by which CP attempts to restart the system after an abnormal termination. This process does not involve the hardware IPL process.

**autonomous system**

A group of routers all using a common routing protocol.

**AUX file**

See auxiliary control file.

**auxiliary control file (AUX file)**

A file that contains a list of the file types of source update files applied to a particular source file or to control the service level used during build. See also control file and preferred auxiliary file.

**auxiliary directory**

An extension of the CMS minidisk file directory, which contains the names and locations of certain CMS modules not included in the CMS minidisk file directory.

**auxiliary file**

See auxiliary control file.

**auxiliary storage**

In z/VM, data storage other than main storage.

**AVS** See APPC/VM VTAM Support.

**AVS virtual machine**

The virtual machine that manages a gateway that allows APPC communication between z/VM systems and an SNA network.

**B**

**backout**

The action taken by Coordinated Resource Recovery (CRR) for an application program to reverse the updates made to protected resources during a transaction (CRR logical unit of work). See also rollback.

**base disk**

(1) In VMSES/E, a minidisk or file pool directory containing the original product files as shipped on the product installation medium.

(2) In Parallel Access Volumes architecture, a term used to refer to a DASD accessed through a primary or base subchannel.

**base file**

The first occurrence of a file in a file pool. It remains the base for the life of the file, even if the file has been renamed. Aliases point to base files.

**base file type**

In VMSES/E, the file type used for a serviceable part when there is no service. The PTF number in the file type is set to "00000". For example, EXC00000 is the base file type for an exec.

**base string**

In VMSES/E, the set of base disks.

**basename**

The part of a path name that remains after all directory names are removed. For example, for the path name dir1/dir2/file.c, the basename is file.c.

**basic sequential access method (BSAM)**

An access method for storing or getting data blocks in a continuous sequence (using either a sequential access device or a direct access device). See also queued sequential access method.

**BFS** See [byte file system](#).

**BFS directory**

A directory in the byte file system. BFS directories can be arranged to form a hierarchy in which one directory can contain one or more subdirectories as well as files. See also [root directory](#) and [working directory](#).

**BFS file**

A file that resides in a byte file system and consists of a continuous stream of individual bytes of data. Such a file has no record format and no record attributes. See also [character special file](#), [external link](#), [FIFO special file](#), and [symbolic link](#).

**big endian**

A format for storage or transmission of binary data in which the most significant bit (or byte) comes first. The reverse convention is little endian.

**binary synchronous communication (BSC)**

A form of telecommunication line control that uses a standard set of transmission control characters and control character sequences for binary synchronous transmission of binary-coded data between stations. Synchronization of characters is controlled by timing signals generated at the sending and receiving stations.

**binder**

A program that converts the output of language translators and compilers into an executable program unit that can either be read directly into virtual storage for execution or stored in a program library. See also [linkage editor](#).

**binding file**

See [language binding file](#).

**bit** The smallest unit of computer information, which has two possible states that are represented by the binary digits 0 or 1.

**bits per inch (bpi)**

A measurement of data density by bit, for example on a tape.

**blade** A hardware unit that provides application-specific services and components. The consistent size and shape (or form factor) of each blade allows it to fit in a BladeCenter<sup>®</sup> chassis.

**BladeCenter chassis**

A modular chassis that can contain multiple blades, allowing the individual blades to share resources such as the management, switch, power, and blower modules.

**blank** (1) In EBCDIC, the character corresponding to X'40'.

(2) One of the characters that belong to the *blank* character class as defined by the LC\_CTYPE category in the current locale. In the POSIX locale, a <blank> is either a <tab> or a <space>.

**block** (1) A unit of DASD space on FBA devices.

(2) A string of data elements recorded or transmitted as a unit. The elements may be characters, words, or physical records. (T)

(3) The unit of storage on a CMS minidisk. The block size is determined when the minidisk is formatted.

(4) In the shared file system, a unit of storage consisting of 4096 bytes.

(5) In CMS multitasking, to stop the execution of a thread until a function has been completed or a condition is satisfied.

**block extent**

A continuous space on a direct access storage volume that does not have to be stored contiguously with a block (4096) of pages.

**block size**

The length of a block in bytes. In CMS, block sizes are 512, 1024, 2048, or 4096 bytes.

**blocked file**

A file in which each data block may contain one or more records. If a file is unblocked, each data block contains only one record.

**blocking**

A way of requesting a service over an interface so that if the request cannot be completed immediately, the requesting process is suspended until the request is completed. See also nonblocking.

**blocking mode**

If the execution of the program cannot continue until some event occurs, the operating system suspends the program until that event occurs.

**border**

In CMS, the boundary around a window. The corners of the border are indicated by the plus (+) sign. The user can enter 1-letter border commands in the border corners to control the window's movement and characteristics.

**boxed** In CP, to shut off all communication with an I/O device because CP has identified a serious problem with the device, for example, exceeding the hot I/O rate.

**bpi** See bits per inch.

**Bpi** See bytes per inch.

**break tree processing**

In Coordinated Resource Recovery (CRR), the process of breaking off other parts of a sync point tree to ensure that a unique LUWID is used by each sync point tree after a protected conversation abnormally ends.

**breakpoint**

A marked point in a process or programmatic flow that causes that flow to pause when the point is reached, typically to allow debugging or monitoring.

**bridge port**

See HiperSockets bridge port.

**broadcast message**

Information that can be sent by the system operator to all terminal users that are enabled to receive messages. The three major classes of messages are: (1) log (LOGMSG) messages automatically displayed at the user's terminal when the user logs on, (2) optional lower priority log messages, and (3) informational warning messages that alert users to some imminent event or action.

**BSAM**

See basic sequential access method.

**BSC** See binary synchronous communication.

**build** In the installation and service of a product, to do the necessary steps to produce executable code or systems.

**build disk**

In VMSES/E, a minidisk or file pool directory where executable versions of serviced objects are built.

**build ID**

A 1- to 8-character alphanumeric identifier that is used to name the software inventory files created during build processing. The user can change this value to define different maintenance levels.

**build list**

An exec that names the serviceable parts included in an object being built, and parameters and options used to build the object.

**build message log**

The file in which the VMFBLD command writes status and error messages during build processing.

**build process**

See [build](#).

**build requisite**

An object that is needed to build another object. For example, when one object is built using another object, the latter is a build requisite of the former. Also, if an object's build requisite is serviced, the object must be rebuilt after its build requisite is built.

**build status table**

See [system-level build status table](#) and [service-level build status table](#).

**build string**

In VMSES/E, the set of build disks.

**build-time requisite**

A product that must be installed before a certain product can run correctly.

**built-in function**

A specialized function, invoked by a keyword, that has been built into the system program code because it is commonly required by many users. Built-in functions are included in CMS EXEC, EXEC2, and REXX.

**built-in stage**

A stage supplied by CMS Pipelines. See also [user-written stage](#).

**byte** A unit of storage, consisting of eight adjacent bits that are operated on as a unit and constitute the smallest addressable unit in the system.

**byte file system (BFS)**

A file system in which a file consists of an ordered sequence of bytes rather than records. BFS files can be organized into hierarchical directories. Byte file systems are enrolled as file spaces in CMS file pools.

**byte ordering**

The method of sorting bytes under specific machine architectures. Of the two common methods, little endian byte ordering places the least significant byte first. The second method, big endian byte ordering, places the most significant byte first.

**bytes per inch (Bpi)**

A measurement of data density by byte, for example on a tape.

**C****cache fast write**

A DASD function where data is written directly to cache without using nonvolatile storage and the data is available for later destaging. This function should be used for data of a temporary nature, or data that is readily recreated.

**callable services library (CSL)**

A package of CMS routines that can be stored as an entity and made available to a high-level language, REXX, or an assembler program.

**canonical mode**

A **tty** (typewriter terminal) input processing mode where input is collected and processed one line

at a time. There are three CLISTs associated with canonical mode drivers: store output to a terminal, store raw input, store cooked data. I/O processing is asymmetric. See also [noncanonical mode](#).

**carriage control character**

A control character that is used to specify a write, space, or skip operation.

**carriage return and line feed (CRLF)**

A sequence of two control characters that moves the cursor to the left side of the screen and down one line or moves the printer carriage to the left and advances the paper by one line.

**cascaded agent**

In Coordinated Resource Recovery (CRR), an agent whose initiator is an agent of another initiator. In this role, an agent is responsible for responding to sync point requests from its cascaded initiator.

**cascaded initiator**

In Coordinated Resource Recovery (CRR), an agent that has protected conversations with cascaded agents. The cascaded initiator takes the role of initiator in relation to its agents and is responsible for propagating sync point requests to its cascaded agents.

**catalog storage group**

The storage group in a file pool that contains information about the objects (such as files and directories) and authorizations that exist in the file pool. Also called *storage group 1*. See also [file pool catalog](#) and [storage group](#).

**CAW** See [channel address word](#).

**CC** See [condition code](#).

**CCS** See [console communication services](#).

**CCW** See [channel command word](#).

**CECP** See [country-extended code page](#).

**central processing unit (CPU)**

A deprecated term for *central processor*.

**central processor (CP)**

The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program loading, and other machine operations.

**central processor complex (CPC)**

A physical collection of hardware that consists of central storage, one or more processors, timers, and channels. In the IBM zEnterprise® environment, the CPC consists of a System z® zEnterprise mainframe and any attached IBM zEnterprise BladeCenter Extension (zBX).

**central storage**

Storage that is an integral part of the central processor complex and includes both main storage and the hardware system area.

**CETI** See [continuously executing transfer interface](#).

**CF** See [coupling facility](#).

**CFVM**

See [coupling facility service machine](#).

**chain link record**

In CMS, a record that contains pointers to the physical blocks currently assigned to a CMS file.

**channel**

The part of a channel subsystem that manages a single I/O interface between a channel subsystem and a set of control units.

**channel address word (CAW)**

An area in storage that specifies the location in main storage at which a channel program begins.

**channel command word (CCW)**

In zSeries systems, an 8-byte command issued to the channel subsystem by a central processor and operating asynchronously with the issuing processor.

**channel path**

The interconnection between a channel and its associated control units.

**channel path ID virtualization**

A virtual machine option used in an SSI cluster that controls the virtualization of channel path IDs and DASD path group IDs.

**channel path identifier (CHPID)**

In a channel subsystem, a value assigned to each installed channel path of the system that uniquely identifies that path to the system. See also [I/O address](#).

**channel program**

One or more channel command words (CCWs) that direct the operation of a data channel.

**channel status word (CSW)**

An area in storage that provides information about the termination of I/O operations.

**channel subsystem (CSS)**

A collection of subchannels that directs the flow of information between I/O devices and main storage, relieves the processor of communication tasks, and performs path management functions.

**channel-attached**

(1) Pertaining to the attachment of devices directly by input/output channels to a host processor.

(2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. See also [link-attached](#).

**channel-to-channel adapter (CTCA)**

A device for connecting two channels on the same processor or on different processors.

**character class**

A named set of characters sharing an attribute associated with the name of the class. The classes and the characters that they contain are dependent on the value of the LC\_CTYPE category in the current locale.

**character delete symbol**

See [logical character delete symbol](#).

**character mode**

See [noncanonical mode](#).

**character special file**

A special BFS file that provides access to an input or output device. The only character special file supported by OpenExtensions is the terminal device file.

**check-out lock**

See [explicit lock](#).

**checkpoint**

An internal file pool server operation during which the changes recorded on the log minidisks are permanently made to the file pool.

**checkpoint start (CKPT start)**

A z/VM system restart that attempts to recover information about closed spool files previously stored on the checkpoint cylinders. The spool file chains are reconstructed, but the original sequence of spool files is lost. Unlike warm start, CP accounting and system message information are also lost. See also [clean start](#), [cold start](#), [force start](#), and [warm start](#).

**child process**

A process created as a result of a spawn. The child process receives a copy of the parent's environment and inherits open files. See also [parent process](#) and [process](#).

**choice** Text or graphics that a user can select. Examples of choices are push buttons, radio buttons, and menu items.

**CHPID**

See [channel path identifier](#).

**circumventive service**

Information that IBM supplies to circumvent a problem by disabling a failing function until a PTF is available as a corrective service fix. See also [patch](#) and [zap](#).

**CKD** See [count-key-data](#).

**CKPT start**

See [checkpoint start](#).

**class** See [privilege class](#).

**class A**

The IBM-defined CP privilege class that corresponds to a system operator. See also [privilege class](#) and [system operator](#).

**Class A network**

In IPv4, an internet network in which the high-order bit of the address is 0. The host number occupies the three, low-order octets.

**class Any**

The IBM-defined CP privilege class that identifies commands available to any user, regardless of the user's privilege class. These commands are primarily those used to gain access to, or relinquish access from, the z/VM system. See also [privilege class](#).

**class authority**

See [privilege class](#).

**class B**

The IBM-defined CP privilege class that corresponds to a system resource operator. See also [privilege class](#) and [system resource operator](#).

**Class B network**

In IPv4, an internet network in which the high-order bit of the address is 1, and the next high-order bit is 0. The host number occupies the two low-order octets.

**class C**

The IBM-defined CP privilege class that corresponds to a system programmer. See also [privilege class](#) and [system programmer](#).

**Class C network**

In IPv4, an internet network in which the two high-order bits of the address are 1 and the next high-order bit is 0. The host number occupies the low-order octet.

**class D**

The IBM-defined CP privilege class that corresponds to a spooling operator. See also [privilege class](#) and [spooling operator](#).

**class E**

The IBM-defined CP privilege class that corresponds to a system analyst. See also [privilege class](#) and [system analyst](#).

**class F**

The IBM-defined CP privilege class that corresponds to an IBM service representative. See also [privilege class](#) and [service representative](#).

- class G**  
The IBM-defined CP privilege class that corresponds to a general user. See also [general user](#) and [privilege class](#).
- class override file**  
In z/VM releases prior to V6.3, a file containing control statements that define changes in the CP privilege class structure.
- clean start**  
A z/VM system restart in which CP purges all spool files, system data files, accounting records, error recording records, symptom records, and the system log message. See also [checkpoint start](#), [cold start](#), [force start](#), and [warm start](#).
- clear connection**  
A standard connection to which no encryption algorithms have been applied; a non-secure connection.
- client** A computer system or process that requests a service of another computer system or process that is typically referred to as a server. Multiple clients can share access to a common server. See also [server](#).
- CLIST** See [command list](#).
- clock comparator**  
A hardware feature (required by z/VM) that causes an interruption when the time-of-day (TOD) clock has equaled or exceeded the value specified by a program or virtual machine.
- CLP** See [current line pointer](#).
- CMMA**  
See [Collaborative Memory Management Assist](#).
- CMR** See [conversation management routine](#).
- CMS** See [Conversational Monitor System](#).
- CMS anchor facility**  
A facility that keeps a list of 16 anchor words and their associated anchor identifiers.
- CMS batch facility**  
A facility that allows a user to run time-consuming or noninteractive CMS jobs in another CMS virtual machine dedicated to that purpose, which frees the user's own terminal and virtual machine for other work.
- CMS editor (EDIT) migration mode**  
A mode in XEDIT, initiated by the CMS EDIT command, that allows the user to enter CMS editor subcommands.
- CMS EXEC language**  
A general-purpose high-level programming language that can be used to write execs.
- CMS EXEC procedure**  
A CMS file with a file type of EXEC that contains a procedure consisting of one or more executable statements to be processed by the CMS EXEC processor. The statements can be commands or execution control statements. If the SET IMPEX setting is ON, the exec is treated as a command, and the procedure is invoked by entering the file name of the file; otherwise, the EXEC command must be used.
- CMS EXEC processor**  
The CMS facility that interprets and executes procedures written in the CMS EXEC language.
- CMS file**  
A file used by the CMS file system. The CMS file management routines write CMS files in fixed

physical blocks regardless of whether they have fixed-length or variable-length records. Each CMS file has an identifier consisting of a file name, a file type, and a directory name or file mode.

**CMS file system**

The architecture, organization, and processing of files in CMS. CMS files are unique to the CMS system and cannot be read or written using other operating systems. CMS files are stored in file pool directories or on minidisks. VSAM and OS data sets and DOS files are not compatible with the CMS file format and cannot be manipulated using CMS file system commands.

**CMS loader work file**

The CMS file loaded into storage by the LOAD command.

**CMS minidisk**

A minidisk formatted specifically for use with the CMS file system. It does not contain a volume table of contents (VTOC) and is incompatible with OS-formatted and DOS-formatted disks. OS and DOS access methods cannot be directly used with CMS-formatted disks.

**CMS minidisk file directory**

A directory on each CMS minidisk that contains data such as the name, format, size, and location of each CMS file on that disk. When a minidisk is accessed, its directory is read into virtual storage and identified with a file mode letter (A through Z). See also [auxiliary directory](#) and [file status table](#).

**CMS multitasking**

CMS services that allow a program to subdivide itself into multiple independently executable parts and to coordinate these execution streams so they together accomplish the objective of the program. These services also allow multitasking programs running in different virtual machines to communicate with each other and to coordinate their processing.

**CMS nucleus**

The portion of CMS that is resident in a user's virtual storage whenever CMS is executing. Each CMS user shares a copy of the CMS nucleus when the user initially loads (IPLs) the CMS named saved system.

**CMS Pipelines**

The CMS facility that provides functions for creating and processing pipelines. See also [pipeline](#).

**CMS search order**

The order of file modes that CMS searches when looking for a file. The standard order of search is alphabetic, A through Z. See also [file mode extension](#).

**CMS system disk**

The minidisk that contains the CMS nucleus and the disk-resident CMS commands. It is located at virtual address 190 and is defined as file mode S. The CMS system disk can also have an extension, usually defined as file mode Y.

**CMS system file**

A file residing on the CMS system disk instead of on a user's disk or in a user's file space.

**CMS user disk**

A minidisk containing files that can be accessed by a user. If the user has read/write access to a disk, the user can create programs and data files on the disk. Files are retained until the user deletes them. The user can also link to and access other users' disks, usually on a read-only basis.

**CMS/DOS**

A CMS environment that simulates many Disk Operating System/Virtual Storage Extended (DOS/VSE) functions, allowing users to use the interactive facilities of CMS to develop and test VSE programs. The CMS/DOS environment is activated by entering the CMS command SET DOS ON.

**CMS/DOS phase library**

See [DOSLIB library](#).

**CMSDOS**

The standard name of the CMS/DOS saved segment, which contains DOS simulation code used by VSE application programs.

**CMS-formatted disk**

See [CMS minidisk](#).

**CNTRL file**

See [control file](#).

**code page**

A particular assignment of code points to graphic characters. Within a given code page, a code point can have only one specific meaning. A code page also identifies how undefined code points are handled. See also [code point](#) and [extended character](#).

**code point**

A unique bit pattern that represents a character in a code page.

**cold start**

A z/VM system restart that ignores previous data areas and accounting information in main storage, and the contents of paging and spool files on CP-owned disks. See also [checkpoint start](#), [clean start](#), [force start](#), and [warm start](#).

**Collaborative Memory Management Assist (CMMA)**

A machine feature that allows z/Architecture guests with the appropriate support to exchange memory usage and status information with z/VM.

**collection**

A group of interconnected z/VM systems that use either the Inter-System Facility for Communications (ISFC) or the Transparent Services Access Facility (TSAF) to communicate with each other. See also [Communication Services collection](#) and [TSAF collection](#).

**command**

A statement used to initiate an action or start a service. A command consists of the command name and its parameters, arguments, flags, or other operands, if applicable. See also [subcommand](#), [user-written program](#), and [utility](#).

**command interpreter**

A program that reads the commands that you type and then processes them.

**command language interpreter**

See [command interpreter](#).

**command line**

The area of the screen where commands are displayed as they are typed.

**command list (CLIST)**

A list of commands and statements designed to perform a specific function for the user.

**command mode**

A state of a system or device in which the user can enter commands. In an editing session, the mode wherein the editor is waiting for the user to enter a command.

**command name**

The first term in a command string that requests a specific action, and is usually followed by operands.

**command privilege class**

See [privilege class](#).

**command processor**

A program that performs an operation specified by a command.

**command scan**

In CMS, a routine that scans the command line entered and converts it to a standard CMS parameter list. See also [parameter list](#).

**command substitution**

In OpenExtensions, the ability to capture the output of any command as an operand to another command by placing that command line within grave accents (` `). The shell first runs the command or commands enclosed within the grave accents and then replaces the whole expression, including grave accents, with their output. This feature is often used in assignment statements.

**commit**

- (1) In a CMS file pool, to permanently change a resource, such as a file.
- (2) In Coordinated Resource Recovery (CRR), to make permanent changes to protected resources during a transaction (CRR logical unit of work). CRR commits changes made by an application program or transaction program.

**commit level**

In CMS Pipelines, an indication of the progress of a stage. The commit level determines when a stage is dispatched relative to other stages and is used to report to other stages when an error has been detected.

**common dump receiver**

In a GCS virtual machine group, one user ID that is appointed to receive other group members' storage dumps. Unless the user specifies otherwise, all dumped information automatically goes to this user ID (identified with the GCS GROUP command). It should be a user ID authorized to receive fetch-protected data and storage with a key other than 14. See also [virtual machine group](#).

**common lock**

In GCS, a word-length field in common storage that contains the machine ID (2 bytes) and the task ID (2 bytes) that are currently holding the lock. If the common lock is free, it contains binary zeros.

**Common Programming Interface (CPI)**

An application protocol that provides languages, commands, and calls that allow the development of applications that are more easily integrated and moved across environments supported by Systems Application Architecture<sup>®</sup> (SAA). z/VM supports the communications and resource recovery elements of the CPI. See also [Common Programming Interface for Communications](#) and [SAA Resource Recovery Interface](#).

**Common Programming Interface for Communications (CPI-C)**

An application programming interface (API) that provides access to interprogram services such as (a) sending and receiving data, (b) synchronizing processing between programs, and (c) notifying a partner of errors in the communication.

**Common Programming Interface for Resource Recovery**

See [SAA Resource Recovery Interface](#).

**common storage**

In GCS, a shared segment of reentrant code that contains free storage space, the GCS supervisor, control blocks, and data that all members of a virtual machine group share. See also [high common storage](#) and [low common storage](#).

**communication link**

See [data link](#).

**Communication Services (CS) collection**

A group of interconnected domains made up of z/VM systems that use the Inter-System Communication Facility (ISFC) to communicate with other z/VM systems.

**communications directory**

A CMS facility that allows APPC/VM applications to connect to a resource using symbolic destination names and special NAMES files.

**communications partner**

The virtual machine at either end of a local APPC/VM path, not necessarily the target of the communications.

**communications server**

A virtual machine that routes communications on behalf of other virtual machines to other servers. The TSAF and AVS virtual machines are communications servers.

**community name**

A password used by hosts running Simple Network Management Protocol (SNMP) agents to access remote network management stations.

**compare states**

In Coordinated Resource Recovery (CRR), the function of conveying the state of the logical unit of work to a participating resource manager, and conveying the state of the protected resource's logical unit of work, including heuristic actions, back to the recovery server.

**completion code**

An indicator that reflects the status of a task set at the time of its completion.

**component**

- (1) A collection of objects that together form a separate functional unit of a product.
- (2) Under VMSES/E, a product-defined subdivision in the product parameter file.

**component override**

See component parameter override.

**component override area**

An area of the product parameter file or product parameter override file that contains one or more component parameter overrides.

**component parameter override**

A component parameter, defined in a component override area, that updates or replaces a component parameter defined in a component area of the product parameter file.

**concurrent server**

A server that can handle many connections at the same time. It can accept new connection requests while still processing the transactions started by previous requests.

**condition code (CC)**

A code that reflects the result of a previous I/O, arithmetic, or logical operation.

**condition variable**

In an application using CMS multitasking services, a variable whose value indicates a state of a mutex-protected shared resource. This state, or condition, is defined by the application. It can be waited on and signaled as being true.

**configuration mode**

A hardware state that allows you to dynamically change your I/O configuration without requiring a system outage (POR or IPL).

**configuration token**

See processor configuration token.

**connect**

To establish a path to communicate with another virtual machine or with the user's own virtual machine.

**connected**

In CMS Pipelines, pertaining to a stream that is attached to a stream from another stage, so that data can flow from one stage to another. If streams are not connected, then data cannot flow from one stage to another.

**connectivity program request block (CPRB)**

An interface control block that requesters and servers use to communicate information.

**connector**

In CMS Pipelines, a symbol that connects the input or output stream of a stage to the beginning or end of a pipeline specified on a CALLPIPE or ADDPIPE pipeline subcommand called by the stage. The minimal connector is an asterisk and a colon (\*:).

**console**

A logical device used for communication between the user and the system. See also Hardware Management Console, integrated ASCII console, integrated 3270 console, Open Systems Adapter-Express Integrated Console Controller, primary system operator console, system console, and virtual console.

**console communication services (CCS)**

A group of CP modules that interface with the VTAM service machine, providing full z/VM console capabilities for SNA terminal users.

**console function**

A subset of CP commands that simulates for a virtual machine user almost all of the functions available to an operator at a system operator console.

**console spooling**

See virtual console spooling.

**console stack**

A storage area that contains information waiting to be processed by CMS, which includes the program stack and the terminal input buffer.

**contention**

A situation in which two programs or users are attempting to access the same system resources. For example, contention can occur when two LUs try to allocate a conversation over the same session at the same time, when two users attempt to edit the same SFS file, but only one is allowed, or when two programs attempt to lock a resource at the same time.

**continuously executing transfer interface (CETI)**

An interface that uses continuously executing channel programs to transfer messages between two systems, or between an application and a control unit.

**control block**

A storage area used by a computer program to hold control information.

**control character**

A character whose occurrence in a particular context initiates, modifies, or stops a control function.

**control data**

In a file pool, the data that controls the DASD space and objects within the file pool. Control data consists of the POOLDEF file, the control minidisk, and all minidisks allocated to storage group 1.

**control file**

(1) A CMS file that is interpreted and directs the flow of a certain process through specific steps. For example, the control file could contain installation steps, default addresses, or PTF prerequisite lists.

(2) In service, a file with file type CNTRL that contains records that identify the updates to be applied and the macro libraries, if any, needed to assemble that source program. See also [auxiliary control file](#).

**control function**

A command to a processing component that is represented by a single bit combination (a control character) or a sequence starting with a control character followed by the parameter values for the command. Examples are the escape character and the end of file character.

**control minidisk**

In a file pool, the minidisk that tracks the physical DASD blocks allocated to the file pool.

**Control Program (CP)**

A component of z/VM that manages the resources of a single computer so that multiple computing systems appear to exist. Each apparent system, or virtual machine, is the functional equivalent of the real computer, and CP simulates the real machine architecture in the virtual machine. See also [virtual machine](#).

**control register**

A register used for operating system control of relocation, priority interruption, program event recording, error recovery, and masking operations.

**control section (CSECT)**

The part of a program specified by the programmer to be a relocatable unit, all elements of which are loaded into adjoining main storage locations.

**control statement**

A statement that controls or affects program execution in a data processing system.

**control terminal**

Any terminal currently controlling system resources and used by the system operator (privilege class A), the resource operator (privilege class B) or the spooling operator (privilege class D).

**control unit (CU)**

A device that coordinates and controls the operation of one or more input/output devices, such as displays, and synchronizes the operation of such devices with the operation of the system as a whole.

**control unit terminal (CUT)**

An IBM protocol used for communication between a control unit or other appropriate interface unit and a workstation. In this protocol, a program in the workstation emulates a terminal for the user in a single logical terminal session. See also [distributed function terminal mode](#).

**controller**

See [control unit](#).

**conversation**

A logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

**conversation characteristics**

The attributes of a conversation that determine the functions and capabilities of the programs within the conversation.

**conversation correlator**

A value that identifies an APPC conversation and is unique at the LU that generates it. The conversation correlator is established when the APPC conversation is established.

**conversation management routine (CMR)**

A routine that resides in the GCS load library and controls the routing of inbound connections through AVS to available service pool virtual machines.

**conversation partner**

One of the two programs involved in a conversation.

**conversation state**

See program state.

**Conversational Monitor System (CMS)**

A component of z/VM that runs in a virtual machine and provides both the interactive z/VM end-user interface and the general z/VM application programming interface. CMS runs only under the control of the z/VM Control Program (CP). See also ESA/390 CMS and z/Architecture CMS.

**conversion table**

See translation table.

**cooked mode**

See canonical mode.

**Coordinated Resource Recovery (CRR)**

A CMS facility that implements the LU 6.2 sync point architecture, which ensures that transactions can update multiple protected resources with integrity. This means that all updates within the transaction are either completed (committed) or not completed (rolled back or backed out). CRR consists of the coordination function (also known as *synchronization point processing*), the resynchronization function, and the logging function. The coordination function resides in the application program's virtual machine. The resynchronization and logging functions reside in the CRR recovery server. See also synchronization point processing, resynchronization, and two-phase commit protocol.

**coordinated transaction**

See logical unit of work and logical unit of work identifier.

**coordinated update**

See logical unit of work and logical unit of work identifier.

**copy file**

A file with a file type of COPY that contains nonexecutable real storage definitions that are referred to by macros and assemble files.

**copy function**

The function initiated by a PF key to copy the contents of a display screen onto an associated hardcopy printer. A remote display terminal copies the entire contents of the screen onto a printer attached to the same control unit. A local display terminal copies all information from the screen, except the screen status information, onto any printer attached to any local display control unit.

**COR** See corrective service.

**corequisite**

A change that must be applied to the system at the same time as another change. In VMSES/E at the system level, a product that must be installed with another product. In VMSES/E at the service level, a PTF that must be applied at the same time as another PTF. A corequisite PTF corrects a problem that requires changes to one or more elements of a product or component. No specific order is required for applying corequisite PTFs. See also hard requisite and if-requisite.

**corrective service (COR)**

Service that IBM supplies to correct a specific problem. See also preventive service.

**count-key-data (CKD)**

A DASD data storage architecture in which the data is stored in variable-length records. Each record contains a count field, usually followed by a key field, followed by the actual data of the record. The count field contains the cylinder number, head number, record number, and the length of the data. The key field contains the record's key (search argument). See also extended count-key-data.

**country-extended code page (CECP)**

An 8-bit code page that has a 93-character set on its nationally standardized code points but is extended to the multilingual character set for the national languages of some European countries.

**coupling facility (CF)**

A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex. See also IBM Internal Coupling Facility.

**coupling facility service machine (CFVM)**

A special disconnected virtual machine that is set up to IPL the Coupling Facility Control Code (CFCC).

**CP** See central processor or Control Program.

**CP directory**

See user directory.

**CP dump**

See hard abend dump, soft abend dump, and snap dump.

**CP privilege class**

See privilege class.

**CP read**

The state that exists when CP is waiting for a response or request for work from the user. On a display terminal, the screen status area indicates CP READ.

**CP READ**

On a display terminal used as a virtual console under z/VM, a screen status that indicates CP is waiting for a response or request for work from the user. See also screen status area.

**CP system disks**

Any disk space that CP uses, other than the CP paging, spooling, and temporary disk space; for example, minidisks.

**CP system residence volume**

The volume on which the CP module resides.

**CP trace table**

A table that z/VM uses for debugging. Its size is a multiple of 4096 bytes and depends on the size of real storage or a user-specified value. This table contains the chronological occurrences of events that take place in the real machine, recorded in a wraparound fashion.

**CP-accessed disk**

Any CMS-formatted minidisk that is available read-only to CP.

**CPC** See central processor complex.

**CPI** See Common Programming Interface.

**CPI Communications**

See Common Programming Interface for Communications.

**CPI Resource Recovery**

See SAA Resource Recovery Interface.

**CPI-C** See Common Programming Interface for Communications.

**CPI-RR**

See SAA Resource Recovery Interface.

**CP-owned disk**

See CP-owned volume.

**CP-owned volume**

A DASD volume formatted by ICKDSF and used for the CP system volume, real system paging, spooling, directory and dump space, and temporary disk space for virtual machines.

**CPRB** See connectivity program request block.

**CPU** See central processing unit.

**CPU pool**

A maximum amount of CPU resources (in terms of real IFL or CP processors) that a group of z/VM guests are allowed to consume in aggregate. A CPU pool is defined with a name and an associated capacity. One or more virtual machines can be assigned to a CPU pool (a virtual machine can be assigned to one CPU pool at a time) and have their aggregate CPU consumption limited to the pool's capacity. The CPU pool can be limited to a specific percentage of the shared logical IFL or CP processors currently varied online, or the CPU pool can be limited to an amount of processor power equivalent to a specific number of real IFL or CP processors.

**CPU timer**

A hardware feature that measures elapsed processor time and causes an interruption when a previously specified amount of time has elapsed. The CPU timer is decremented when the processor is executing instructions, is in a WAIT state, and is executing program loading instructions, but not when the processor is in a stopped state.

**CPXLOAD directives**

Instructions used before CPXLOAD commands or configuration file statements that dictate how CP will load customer-written CP routines.

**critical section**

In CMS multitasking, a block of code that manipulates a shared resource, such as a data structure or device.

**CRLF** See carriage return and line feed.

**cross system extensions (CSE)**

In z/VM releases prior to V6.3, a CP facility that allows users in up to four interconnected z/VM systems to participate in a multisystem environment, as though all participating systems were one complex.

**cross-system link**

A facility that extends CP link protocols to control access across multiple non-SSI z/VM systems for minidisks on shared DASD.

**cross-system spool**

In z/VM releases prior to V6.3, a CSE facility that extends CP spooling capabilities within the CSE complex.

**CRR** See Coordinated Resource Recovery.

**CRR recovery server**

A virtual machine that provides the CRR logging and resynchronization functions. A processor can have only one CRR recovery server.

**CS collection**

See Communication Services collection.

**CSE** See cross system extensions.

**CSECT**

See control section.

**CSL** See callable services library.

**CSL routine**

An assembler program that resides in a CSL and can be invoked from a high-level language, REXX, or an assembler program to do a specific function.

**CSS** See channel subsystem.

**CSW** See channel status word.

**CTCA** See channel-to-channel adapter.

**CU** See control unit.

**current directory**

See working directory.

**current line pointer (CLP)**

A pointer that indicates the line of a CMS file on which the CMS editor is currently working.

**current record**

The record pointed to by the current line pointer. The record that is currently available to the program.

**customer-written CP routine**

A routine written by a customer to modify a CP function. A file containing customer-written CP routines can be loaded into CP's virtual storage using a CPXLOAD command or configuration file statement.

**CUT** See control unit terminal mode (CUT).

**cylinder**

The tracks of a disk storage device that can be accessed without repositioning the access mechanism.

**D**

**DAC** (1) See data access component.

(2) See discretionary access control.

**daemon**

A program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their task, while others operate periodically.

**DASD**

See direct access storage device.

**DASD Dump Restore (DDR) program**

A service program that copies all or part of a minidisk onto tape, loads the contents of a tape onto a minidisk, or sends data from a DASD or from tape to the virtual printer.

**DAT** See dynamic address translation.

**data access component (DAC)**

The part of a file pool server machine that handles the manipulation of the data in the file pool. See also storage access component.

**data area**

A location in virtual storage used for communication between components and program modules. A data area contains only data; it does not contain executable code.

**data control block (DCB)**

A storage area that contains information that defines the characteristics of the data in a file and describes the I/O device requirements for handling the data.

**data definition (DD)**

A program statement that describes the features of, specifies relationships of, or establishes the context of data. A data definition reserves storage and can provide an initial value.

**data definition (DD) statement**

A job control statement that is used to define a data set for use by a batch job step, started task or job, or an online user.

**data definition name (ddname)**

The name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

**Data Facility Storage Management Subsystem for VM (DFSMS/VM)**

A tools suite that provides management of data and storage resources on z/VM. DFSMS/VM manages data in SFS storage groups by assigning management classes to SFS files and directories, which can then be automatically managed based on the criteria in each management class. DFSMS/VM manages CMS minidisks by facilitating the movement of minidisks from one location (DASD) to another. DFSMS/VM Removable Media Services (RMS) provides native z/VM support for the IBM 3494 and 3495 Tape Library Dataservers.

**data link**

The physical connection (communications lines, modems, controller, work stations, other communications equipment), and the rules (protocols) for sending and receiving data between two or more locations in a data network. See also [link](#).

**data link layer**

In the Open Systems Interconnection (OSI) reference model, the layer (Layer 2) that provides services to transfer data between entities in the network layer over a communication link. The data link layer detects and possibly corrects errors that may occur in the physical layer.

**data set**

The major unit of data storage and retrieval in z/OS<sup>®</sup>, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. Synonymous with *file* in z/VM.

**data space**

Informal term for an additional address space created by CP at the request of a program running in an XC virtual machine. This term is usually used to refer to these additional address spaces because they can contain data (operands of instructions) but instructions cannot be executed from within them.

**data stream**

(1) All information (data and control commands) sent over a data link, usually in a single read or write operation.

(2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

**datagram**

See [IP datagram](#).

**DBCS** See [double-byte character set](#).

**DCB** See [data control block](#).

**DCSS** See [discontiguous saved segment](#).

**DD** See [data definition](#).

**DD statement**

See [data definition \(DD\) statement](#).

**ddname**

See [data definition name](#).

**DDR** See DASD Dump Restore program.

**dedicated channel**

A channel attached to a virtual machine for its sole use, so that CP can bypass translating the addresses of virtual devices.

**dedicated device**

An I/O device or line that is not being shared among users, but has been permanently assigned to a particular virtual machine by a virtual machine definition, or has been temporarily attached by the resource operator to the user's virtual machine.

**dedicated maintenance mode**

In a file pool server machine, a processing mode in which the server has exclusive use of the file pool, and the file pool is unavailable to other users. See also multiple user mode.

**default file attributes**

Values assumed by the CMS editor based on certain reserved CMS file types. These values include record type (fixed-length or variable-length), record length, tab setting, uppercase or lowercase setting, record sequence numbering, verification mode, truncation column, and other related record characteristics.

**delimited string**

In CMS Pipelines, a character string beginning and ending with a delimiter. See also null string.

**delimiter**

A flag that is formed by a character or a sequence of characters to group or to separate items of data by marking the beginning and end of a unit of data. The delimiter is not a part of the flagged unit of data.

**delta disk**

In VMSES/E, a minidisk or file pool directory containing control files used for servicing. The control files can include apply lists, exclude lists, PTF parts lists, auxiliary control files, update files, and text files.

**delta string**

In VMSES/E, the set of delta disks.

**dependent PTF**

A PTF that has another PTF as a prerequisite or corequisite.

**dependent requisite**

A product that must be installed before another product can be installed correctly. Unlike prerequisites, dependent requisites are no longer satisfied when the requisite product is superseded. This occurs when a product requires a specific level of another product and newer levels of the product will not meet the requirements.

**derived file**

A CMS file created by the build process from selected versions of objects. Some derived files can be provided by IBM or the customer installation can create them.

**description table**

The software inventory table that contains the descriptions of the products that have been received on the system (system-level description table) or contains the abstract information for an APAR that has been received on the system (service-level description table). See also system-level description table and service-level description table.

**descriptor**

An unsigned integer that a UNIX system uses to identify an object supported by the kernel. Descriptors can represent files, pipes, sockets, and other I/O streams. They are created, acted on, and deallocated by system calls specific to the object.

**destage**

The asynchronous writing of new or updated data from cache or nonvolatile storage to DASD.

**device address**

See I/O address.

**device block**

A storage area used by a computer program to hold device information.

**device driver**

(1) A collection of subroutines that control the interface between I/O device adapters and the processor.

(2) In CMS Pipelines, a stage that reads data from or writes data to I/O and storage devices, host environments (including CP, CMS, and XEDIT), and REXX and EXEC 2 variables. See also filter.

**device identifier**

In a channel subsystem, an address, not apparent to the program, that is used to communicate with I/O devices. See also I/O address.

**device number**

In a channel subsystem, one to four hexadecimal digits that uniquely identify an I/O device. See also I/O address.

**device-level addressing**

In an ESCON<sup>®</sup> I/O interface, the level of addressing that identifies an I/O device to the channel or control unit after the control unit has been determined through link-level addressing. See also link-level addressing.

**DFSMS/VM**

See Data Facility Storage Management Subsystem for VM.

**DFT** See distributed function terminal mode.

**DIAGNOSE interface**

A programming mechanism that lets any virtual machine directly communicate with CP by way of the DIAGNOSE instruction. Specific interface codes let a virtual machine more efficiently request specific CP services.

**DIRCONTROL directory**

See directory control directory.

**direct access storage device (DASD)**

A mass storage medium in which the data access time is effectively independent of the data location.

**directory**

(1) A type of file containing the names and controlling information for other files or other directories. See also BFS directory, home directory, path name, root directory, SFS directory, and working directory.

(2) See also CMS minidisk file directory.

(3) See also user directory.

**directory control directory**

A type of SFS directory with the functional characteristics of a minidisk. A single access authority applies to the directory and all files in the directory. When the directory is accessed in read-only mode, changes are not visible until the directory is released and reaccessed. When the directory is accessed in read/write mode, changes become available as they are made. See also file control directory.

**directory definition**

See virtual machine definition.

**directory entry**

(1) In the source directory, a block of directory statements that begins with one of the following

statements: GLOBALDEFS, PROFILE, USER, IDENTITY, or SUBCONFIG. A directory entry ends when another entry begins or at the end of the user directory.

(2) See [virtual machine definition](#).

(3) An object that associates a file name with a file. Several directory entries can associate names with the same file. In OpenExtensions, see [link](#).

**directory identifier (dirid)**

The identification of an SFS directory in a command through either (a) the fully qualified directory name (in which the file pool ID and user ID can be allowed to default), (b) the file mode letter of the accessed directory, or (c) a plus (+) or minus (-) designation indicating a directory up or down the directory structure from the specified file mode.

**Directory Maintenance Facility (DirMaint™)**

An optional feature of z/VM that provides interactive facilities for maintaining the user directory for one or more z/VM systems.

**directory name**

A 1- to 16-character alphanumeric name for an SFS directory. Two or more SFS directories may have the same name if each has a different parent directory. See also [fully qualified directory name](#).

**directory statement**

A statement in the source form of the user directory. Directory statements define such things as the CP-formatted volume on which the object directory is created and the configuration and operational characteristics of the virtual machine known to the z/VM operating system.

**dirid** See [directory identifier](#).

**DirMaint**

See [Directory Maintenance Facility](#).

**dirname**

See [fully qualified directory name](#).

**disable**

(1) To prevent CP from processing requests for functions or resources, such as commands or devices.

(2) In SFS, to lock a file space or storage group.

**disabled wait state**

A situation in which the system or virtual machine is suspended and cannot accept incoming data.

**disconnected mode**

The mode of operation in which a virtual machine is executing without a physical line or terminal connected as an operator console. Under the single console image facility (SCIF), a user can be disconnected from a primary virtual console but still have console communications through the console of the secondary user.

**discontiguous saved segment (DCSS)**

A saved segment that begins and ends on a megabyte boundary and defines an area of virtual storage that is accessed by name and can be outside the address range of a virtual machine. A DCSS can contain code and data. See also [saved segment](#).

**discretionary access control (DAC)**

A method of restricting access to data objects based upon authorizations granted to users or groups of users. Discretionary access control protects all system objects from unauthorized access. Usually, authorization to access an object is granted by its owner. Occasionally, it can be granted by someone else, such as a privileged administrator. See also [mandatory access control](#).

**disk** See [minidisk](#).

**disk pack**

A set of flat, circular recording surfaces that a disk storage device uses.

**dispatch list**

The list of virtual machines currently contending for real processor time. Virtual machines in the dispatch list can be in two states: dispatchable or nondispatchable. A dispatchable virtual machine is one that is ready to use a real processor. A nondispatchable virtual machine is one that is not ready to use a real processor because it is waiting for a resource or the completion of an activity. Virtual machines in the dispatch list retain the transaction class they were assigned while waiting in the eligible list. When E0 virtual machines (virtual machines that do not wait in the eligible list for resources to become available) enter the dispatch list, they are included in the count of Q0 virtual machines. Thus, E0 virtual machines are called Q0 virtual machines while they are in the dispatch list. Similarly, Q1, Q2, and Q3 virtual machines are virtual machines that on entry to the dispatch list belonged to the E1, E2, and E3 transaction classes, respectively.

**dispatch request queue**

A queue of executable CP tasks, I/O tasks, and timer requests ready to be dispatched.

**dispatch vector**

The list of virtual CPUs assigned to and waiting for dispatch on a given logical CPU.

**dispatcher**

The program in CP that places virtual machines or CP tasks into execution. The dispatcher selects the next virtual machine to run and prepares the virtual machine for problem state execution.

**dispatching**

- (1) The starting of virtual machine execution.
- (2) In CMS multitasking, to give each thread a turn at running on a processor.

**dispatching class**

In CMS multitasking, a set of threads with two properties:

- No thread in the class is ever preempted by another in the same class unless the first thread voluntarily ceases execution.
- No two threads in a class are ever processed in parallel.

However, any thread can be preempted by or processed in parallel with any thread in a different class.

**dispatch/scheduler favoring scheme**

A set of criteria that the dispatcher and scheduler use to create a bias in favor of queue 1 (Q1) users. Q1 users are usually highly interactive users.

**display mode**

See [full-screen mode](#).

**distributed function terminal (DFT) mode**

- (1) A protocol used for communication between a terminal and an IBM control unit that allows multiple concurrent logical terminal sessions.
- (2) A mode of operation that uses this protocol.
- (3) See also [control unit terminal mode \(CUT\)](#).

**distributed processing**

- (1) Processing that takes place among two or more linked systems.
- (2) A transaction that involves more than one node to complete its processing. A node might be a z/VM virtual machine in the same processor or a different processor, or it might be an LU partner in a non-VM system (for example, z/OS). Using a file pool server is an example of distributed processing (a user machine and a server machine cooperating), and the file pool is an example of a distributed resource.

**Distributed Programming Interface (DPI)**

A programming interface that permits end users to add, delete, or replace management variables dynamically in the local Management Information Base (MIB) without requiring them to recompile the SNMP agent.

**distributed resource**

A resource that is accessed by users on two or more systems. See also distributed processing.

**distribution code**

A 1- to 8-character identification word, specified in the user directory, that is printed or punched with the user ID in the separator page (or punched card) to further identify the location or department of the user.

**DLL** See dynamic link library.

**DMSPARMS file**

A CMS file with a file type of DMSPARMS that contains the start-up parameters used by file pool servers: BFS, CRR, FIFO, or SFS.

**DNS** See Domain Name System.

**domain**

- (1) That part of a computer network in which the data processing resources are under common control. (T)
- (2) The set of all possible communications partners. Each domain has a different set of possible partners, although they may overlap.
- (3) In an internet, a part of the naming hierarchy in which a domain name consists of a sequence of names (labels) separated by periods (dots).
- (4) See relocation domain.

**domain controller**

A communications server that manages communications within a domain and between domains. See also VM domain controller.

**domain name server**

In Internet communications, a server program that supplies name-to-address conversion by mapping domain names to IP addresses. See also name server.

**Domain Name System (DNS)**

The distributed database system that maps domain names to IP addresses.

**domain naming**

A hierarchical system for naming network resources.

**Domination Rule**

The rule that defines when one security label dominates another:

1. The security level of the first label is greater than or equal to the security level of the second label.
2. All the security categories of the second label are found among those of the first label.

**dormant list**

A list of virtual machines that are idle or waiting for completion of a long event, such as a tape read.

**dormant state**

A state in which the active pages of a virtual machine have been paged out.

**DOS simulation under CMS**

See CMS/DOS.

**DOSLIB library**

A CMS library that contains the executable phases produced by the DOS Linkage Editor under CMS. These phases can be fetched and executed only under CMS/DOS.

**dot** A BFS path name component consisting of a single dot character (.) that refers to the directory specified by the preceding path name component.

**dot-dot**

A BFS path name component consisting of two dot characters (..) that refers to the parent directory of the preceding path name component. As a special case, in the root directory, dot-dot refers to the root directory itself.

**dotted decimal notation**

The syntactic representation for a 32-bit integer that consists of four 8-bit numbers, written in base 10 and separated by dots. Many internet application programs accept dotted decimal notations in place of destination machine names.

**double-byte character set (DBCS)**

A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by the 256 code points provided by EBCDIC, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS.

**double-density DASD volume**

A DASD volume with twice the standard number of cylinders for its device type. See also [single-density DASD volume](#) and [triple-density DASD volume](#).

**doubleword**

A contiguous sequence of bits or characters that comprises two computer words and is capable of being addressed as a unit.

**DPA** See [dynamic paging area](#).

**DPI** See [Distributed Programming Interface](#).

**drain** (1) For spooling devices, to bring spooling operations on a real device to a halt after the current spool file is processed.

(2) For DASD, to halt I/O operations on a real device after the current I/O operations are completed.

**dump** (1) To record or copy, at a particular instant, data from one storage device onto another storage device to protect the data and debug the program.

(2) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

(3) Data that is copied in a readable format from main or auxiliary storage to an external medium such as tape or disk.

(4) A capture of storage information at the time of an error.

(5) See also [abend dump](#), [DASD Dump Restore program](#), [stand-alone dump](#), and [virtual machine dump](#).

**Dump Viewing Facility**

A component of z/VM that can interactively diagnose system problems. Using this facility, a user can display, format, and print data interactively from virtual machine dumps, as well as display and format recorded trace data. Analysis of CP stand-alone dumps, CP ABEND dumps, and virtual machine dumps of a CP system, functions formerly performed by the Dump Viewing Facility, are now performed by the VM Dump Tool. See also [VM Dump Tool](#).

**duplex pair**

A logical volume comprised of two physical devices with all data recorded twice, once on each device.

**duplex protocol**

A communication protocol in which data can be sent and received at the same time. See also [half-duplex protocol](#).

**dynamic address translation (DAT)**

The change of a virtual storage address to a real storage address during execution of an instruction.

**dynamic configuration**

The process of configuring or reconfiguring a Communication Services (CS) collection or TSAF collection when a z/VM system enters or leaves the collection after a link goes up or down within the collection.

**dynamic I/O commands**

A group of CP commands that can change the I/O configuration without requiring a power-on reset (POR) of the hardware or an initial program load (IPL).

**dynamic I/O configuration**

In z/VM, the ability to change the I/O configuration without requiring a power-on reset (POR) of the hardware or an initial program load (IPL).

**dynamic I/O reconfiguration**

See [dynamic I/O configuration](#).

**dynamic link library (DLL)**

A module containing dynamic link routines that is linked at load or run time.

**dynamic paging area (DPA)**

An area of real storage that CP uses for virtual machine pages.

**dynamic resource allocation**

An allocation technique in which the resources assigned for execution of computer programs are determined by criteria applied at the moment of need.

**dynamically modified channel program**

A channel program changed by the program or by data being read in from a channel during the interval between the execution of the START I/O (SIO) instruction and the channel end interruption.

**E****Eb (exabit)**

(1) For processor, real and virtual storage capacities, and channel volume, 2 to the power of 60, or 1,152,921,504,606,846,976, bits.

(2) For disk storage capacity and communications volume, 10 to the power of 18, or 1,000,000,000,000,000,000 bits.

**EB (exabyte)**

(1) For processor, real and virtual storage capacities, and channel volume, 2 to the power of 60, or 1,152,921,504,606,846,976, bytes.

(2) For disk storage capacity and communications volume, 10 to the power or 18, or 1,000,000,000,000,000,000, bytes.

**EBCDIC**

See [extended binary-coded decimal interchange code](#).

**EC** See [engineering change](#).

**ECB** See [event control block](#).

**ECC** See error checking and correction.

**ECKD™**  
See extended count-key-data.

**EDF** See enhanced disk format architecture.

**EDIT macro**

(1) A macro that executes in the CMS editor (EDIT) migration mode.

(2) A CMS file whose file name starts with a dollar sign (\$) character and whose file type is EXEC.

**edit mode**

The XEDIT environment in which a user can enter editor subcommands to insert, change, delete, or rearrange the contents of a CMS file. See also input mode.

**effective GID**

(1) The current POSIX group ID (GID), but not necessarily the group to which the user's login ID belongs (real GID). For example, if a logged-in user changes to another GID, the GID to which the user changes becomes the effective GID until the user changes back to the original GID. See also real GID and saved set-GID.

(2) An attribute of a process that is used in determining various permissions, including file access permissions. This value is subject to change during the process lifetime. (P1)

**effective root directory**

The point at which a byte file system starts when searching for a file. The path name of the effective root directory begins with a slash (/).

**effective UID**

(1) The current POSIX user ID (UID), but not necessarily equivalent to the user's login ID (real UID). For example, if a logged-in user changes to another UID, the UID to which the user changes becomes the effective UID until the user changes back to the original UID. See also real UID and saved set-UID.

(2) An attribute of a process that is used in determining various permissions, including file access permissions. This value is subject to change during the process lifetime. (P1)

**EGP** See Exterior Gateway Protocol.

**electronic envelope**

In VMSES/E, an electronically-delivered container for recommended service upgrade (RSU) or corrective service (COR) service files.

**eligible list**

The list of virtual machines waiting to move into the dispatch list. The virtual machines in the eligible list are assigned to one of four transactions classes, depending on whether they are to wait in the eligible list at all (E0 virtual machines do not wait in the eligible list for resources to become available) or on the expected length of their transactions (E1 for short transactions, E2 for medium-length transactions, and E3 for long-running transactions). FIFO queuing is used within the eligible list when eligibility priorities are equal. See also dispatch list.

**ELN** See exchange log names.

**empty directory**

A directory that contains, at most, directory entries for dot and dot-dot.

**empty string**

A character array whose first element is a null character.

**emulation**

The use of a data processing system to imitate another data processing system such that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated system.

**Emulation Program (EP)**

An IBM control program that allows a channel-attached IBM communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control Unit, or an IBM 2703 Transmission Control Unit.

**enabled wait state**

A situation in which the system or virtual machine is suspended, but can accept incoming data.

**encapsulation**

A process used by layered protocols in which a lower-level protocol accepts a message from a higher-level protocol and places it in the data portion of the low-level frame. As an example, in Internet terminology, a packet would contain a header from the physical layer, followed by a header from the network layer (IP), followed by a header from the transport layer (TCP), followed by the application protocol data.

**end of file (EOF)**

- (1) A coded character recorded on a data medium to indicate the end of the medium.
- (2) In CMS Pipelines, a condition in which no data can flow through a stream because the stream has been severed.
- (3) When editing a file with XEDIT, the line below the file contents area.

**end of volume (EOV)**

The logical end of a physical tape volume.

**end-of-tape marker (EOT)**

A marker on a magnetic tape used to indicate the end of the permissible recording area.

**end-of-transmission character (EOT)**

A transmission control character used to indicate the conclusion of a transmission that might have included one or more texts and associated message headings.

**engineering change (EC)**

A change to a hardware device, either to the physical device or to the microcode.

**enhanced disk format (EDF) architecture**

A system of allocating DASD space, in which portions of DASD space are allocated to users as minidisks.

**ensemble**

A collection of one or more zEnterprise nodes (including any attached zBX) that are managed as a single logical virtualized system by the Unified Resource Manager, through the use of a Hardware Management Console.

**ensemble member**

A zEnterprise node that has been added to an ensemble.

**Enterprise Systems Architecture/390 (ESA/390)**

An IBM mainframe computer and operating system architecture. It provides both 24-bit and 31-bit addressing modes. See also [z/Architecture](#).

**Enterprise Systems Architecture/Extended Configuration (ESA/XC)**

An IBM virtual machine architecture in which DAT-off programs, such as CMS and CMS applications, can create and access additional address spaces called data spaces. These additional address spaces can also be shared with programs running in other virtual machines. CP provides ESA/XC in an XC mode virtual machine.

**Enterprise Systems Connection (ESCON)**

An IBM fiber optic connection technology that interconnects mainframe computers, workstations, and network-attached storage devices through a channel and supports half duplex data transfers.

**entitlement**

The amount of real processor time a logical partition is guaranteed.

**entry point**

The address or label of the first instruction executed on entering a computer program, routine, or subroutine. A computer program, routine, or subroutine may have a number of different entry points, each perhaps corresponding to a different function or purpose.

**envelope**

(1) A container stored in a user's profile that contains the user's encrypted password or password phrase so that it can be retrieved and decrypted by authorized users as part of a password synchronization solution.

(2) See electronic envelope.

**environment**

(1) The settings for shell variables and paths set when the user logs in. These variables can be modified later by the user.

(2) A block of information passed ("exported") to a command when the command is called. This block contains a number of environment variables. The environment provides information that the program can use in its operation, in a form that relieves you of the need to specify it with every command.

(3) The set of all factors that can affect how a program behaves.

(4) A named collection of logical and physical resources used to support the operation of a function.

(5) See also environment variable.

**environment information interface**

A z/VM interface that can provide CPU resource information and enable virtual machines to understand the characteristics of the environment in which they are running. Information is provided at many levels: machine, logical partition, CPU pool, and virtual machine. This interface can be exploited by application tools that need to understand the configuration of the application used by z/VM guest operating systems.

**environment variable**

(1) A variable that describes the operating environment of the process and typically includes information about the home directory, command search path, the terminal in use, and the current time zone.

(2) The name of a file that contains shell commands to customize a shell environment.

(3) A variable included in the current software environment that is available to any called program that requests it.

**EOF** See end of file.

**EOT** (1) See end-of-tape marker.

(2) See end-of-transmission character.

**EOV** See end of volume.

**EP** See emulation program.

**erased alias**

An alias that no longer points to a base file because the base file was erased.

**ERE** See extended regular expression.

**EREP file**

A collection of error records produced by the malfunctioning of hardware components on a computer and stored for processing by the EREP program.

**error checking and correction (ECC)**

In a processor, the detection and correction of all single-bit errors, plus the detection of double-bit and some multiple-bit errors.

**error code**

A value that identifies an error condition.

**error recording area**

The DASD space defined during system generation that CP uses to record formatted outboard error recordings, machine check records, and channel check records.

**ESA virtual machine**

A virtual machine that simulates ESA/390 functions. An ESA virtual machine can also simulate z/Architecture, as determined by the program that is loaded into the virtual machine.

**ESA/390**

See [Enterprise Systems Architecture/390](#).

**ESA/390 CMS (CMS)**

The traditional version of CMS, generally referred to simply as CMS, which runs in the ESA/390 architecture in an ESA or XA virtual machine, and runs in the ESA/XC architecture in an XC virtual machine. See also [z/Architecture CMS](#).

**ESA/XC**

See [Enterprise Systems Architecture/Extended Configuration](#).

**escape character**

(1) In XEDIT, a character used as a control character identifier; that is, when this character appears in the text, the next character is interpreted as a control character. The escape character is used in XEDIT input mode to allow the user to enter a subcommand without leaving input mode.

(2) In shell programming and `tty` programming, the `\` (backslash) character, which indicates that the next character is not intended to have the special meaning usually assigned to it.

**escape sequence**

(1) A sequence of characters that begins with a `\` (backslash) and is interpreted to have a special meaning to the shell.

(2) A sequence sent to a terminal to perform actions such as moving the cursor, changing from usual to reverse video, and clearing the screen.

**escape symbol**

See [logical escape symbol](#).

**ESCD** See [ESCON Director](#).

**ESCM** See [ESCON Manager](#).

**ESCON**

See [Enterprise Systems Connection](#).

**ESCON channel**

A channel that supports ESCON protocols.

**ESCON Director (ESCD)**

A device that provides connectivity capability and control for attaching any two links to each other in an ESCON environment.

**ESCON Manager (ESCM)**

An IBM licensed program that provides host control and intersystem communication capability for ESCON Director operations.

**ESM** See external security manager.

**Ethernet**

A packet-based networking technology for local area networks (LANs) that supports multiple access and handles contention by using Carrier Sense Multiple Access with Collision Detection (CSMA/CD) as the access method. Ethernet is standardized in the IEEE 802.3 specification.

**event** (1) An occurrence of significance to a task; for example, the completion of an asynchronous operation, such as an input/output operation.

(2) See also system event.

**event control block (ECB)**

A control block used to represent the status of an event. Event control blocks are standard mechanisms used to synchronize multiple events. The process of turning on the event complete bit is referred to as posting the event control block. Asynchronous routines such as a timer or an external interrupt handler post event control blocks to signal completion.

**event name**

In applications using CMS multitasking services, a character string of arbitrary length and composition that is the primary identifier of an event.

**event signal**

In applications using CMS multitasking services, a signal that originates with CMS or an application to indicate that an event has occurred. The signal consists of the event name and additional data to be matched against keys specified by processes that want to monitor the event.

**exabit** See Eb.

**exabyte**

See EB.

**exchange log names**

In Coordinated Resource Recovery (CRR), an exchange of log data (log names, LU names, and TPN) for validating levels of log data between the CRR recovery server and participating resource managers, or between the CRR recovery server on one processor and another CRR recovery server on another processor.

**exclude list**

A file listing PTFs to be omitted from a product or component.

**exclusion list**

In z/VM releases prior to V6.3, a list of user IDs excluded from the cross system extensions environment.

**exclusive access**

A minidisk access mode that means the user holding the exclusive access has stable access with the added restriction that no other user has or can get access to the specified minidisk until the access is released. See also stable access.

**exclusive lock**

See explicit exclusive lock and implicit exclusive lock.

**exclusive segment**

A range of pages within a saved system, member saved segment, or discontinuous saved segment that can be accessed by only one virtual machine at a time.

**exec** A file containing a procedure that consists of one or more executable statements. The statements can be commands or execution control statements. See also CMS EXEC procedure, EXEC 2 program, REXX exec, executable file, executable program, and GCS command file.

**EXEC 2 exec**

See [EXEC 2 program](#).

**EXEC 2 language**

A general-purpose high-level programming language that can be used to write execs.

**EXEC 2 processor**

The CMS facility that interprets and executes procedures written in the EXEC 2 language.

**EXEC 2 program**

A CMS file with a file type of EXEC that contains a procedure consisting of one or more executable statements to be processed by the CMS EXEC 2 processor. The statements can be commands or execution control statements. If the SET IMPEX setting is ON, the exec is treated as a command, and the procedure is invoked by entering the file name of the file; otherwise, the EXEC command must be used.

**exec function**

One of a collection of functions that causes the running process to be completely replaced by the program passed as an argument to the function.

**EXEC procedure**

See [CMS EXEC procedure](#).

**executable file**

A file that contains programs or commands that perform operations or actions to be taken. See also [exec](#), [executable program](#), and [processable file](#).

**executable phase library**

See [DOSLIB library](#).

**executable program**

- (1) A program that has been link-edited and can therefore be run in a processor.
- (2) The set of machine language instructions that constitute the output from the compilation of a source program.
- (3) A program that can be executed as a self-contained procedure. It consists of a main program and, optionally, one or more subprograms.
- (4) A shell script.

**executable statement**

- (1) In an exec procedure or XEDIT macro, any statement processed by the REXX/VM Interpreter, the EXEC 2 processor, or the CMS EXEC processor. This statement can be an execution control statement or assignment statement, a command, or a null line.
- (2) A statement that causes an action to be taken by the program. For example, to calculate, to test conditions, or to alter usual sequential processing.

**exit** A customer-written or customized procedure that, under certain conditions, receives control from the system. An exit consists of an exit point and an exit routine. See also [installation-wide exit](#) and [user exit](#).

**exit point**

A specific point in a system function or program where control can be passed to one or more specified exit routines.

**exit routine**

A program that receives control from another program in order to perform specific functions.

**expanded storage**

Electronic storage not directly addressable by a processor. Expanded storage can be accessed by a processor by means of instructions (PAGE IN and PAGE OUT) that transfer 4-KB blocks of data

from expanded storage to main storage or from main storage to expanded storage. In z/VM, expanded storage can be shared by CP and one or more virtual machines.

**explicit exclusive lock**

In SFS and BFS, an explicit lock on a file or directory that allows only one person to access (read or write) a file or directory at a given time.

**explicit lasting lock**

In SFS and BFS, an explicit lock on a file or directory that lasts until the lock is deleted with the DELETE LOCK command or DMSDELOC routine.

**explicit lock**

In SFS and BFS, a lock that the user explicitly sets on a file or directory by entering a CREATE LOCK command before running a program or by issuing a DMSCRLOC routine within a program. See also explicit exclusive lock, explicit lasting lock, explicit session lock, explicit share lock, explicit update lock, and implicit lock.

**explicit session lock**

In SFS and BFS, an explicit lock on a file or directory that lasts until the end of a CMS session.

**explicit share lock**

In SFS, an explicit lock on a file or directory that allows multiple readers, but no writers.

**explicit update lock**

(1) In SFS, an explicit lock on a file or directory that allows other users to read while one user reads or updates that file or directory.

(2) In BFS, an explicit lock that has the same effect as an explicit exclusive lock.

**extended binary-coded decimal interchange code (EBCDIC)**

A set of 256 characters, with each character represented by 8 bits (1 byte).

**extended character**

(1) A character other than a 7-bit ASCII character. An extended character can be a 1-byte code point with the eighth bit set (ordinal 128 through 255).

(2) A double-byte character that is stored in a DBCS font file, not in the hardware of a DBCS-capable work station. When displaying or printing extended characters, the work station receives them from the DBCS font table under control of the extended character processing function of the operating system.

**extended communications**

A set of program-to-program communication routines that are an extension to CPI-Communications routines. These routines are not part of IBM's Systems Application Architecture.

**extended count-key-data (ECKD)**

An extension of the count-key-data (CKD) DASD data storage architecture that is specifically designed to support nonsynchronous DASD control units.

**extended PLIST (untokenized parameter list)**

Four addresses that indicate the extended form of a command as it was entered at a terminal. See also parameter list.

**extended regular expression (ERE)**

A pattern (sequence of characters or symbols) constructed according to the rules defined in POSIX.2 2.8.4.

**extended remote copy (XRC)**

A remote copy service that provides asynchronous DASD copy operations.

**Exterior Gateway Protocol (EGP)**

A reachability routing protocol used by gateways in a two-level internet.

**eXternal Data Representation (XDR)**

A standard developed by Sun Microsystems, Incorporated for representing data in machine-independent format.

**external link**

A type of BFS file that points to an object outside of the byte file system (data residing on a CMS minidisk or in an directory), creates an implicit mount point, or contains data in an application-defined format.

**external object**

Within a file pool, an object that the file pool server can refer to, but not control. An external object can be used to store a remote name of an entity residing outside the file pool. The external entity can be a file, but does not have to be.

**external security manager (ESM)**

An application program not included in the z/VM base product that can be used to maintain z/VM security and integrity by identifying and verifying users and authorizing users to access protected resources.

**external storage**

Data storage located on external devices, such as DASD or tape.

**Extreme Cloud Administration Toolkit (xCAT)**

An open source scalable distributed computing management and provisioning tool that provides a unified interface for hardware control, discovery, and OS diskful/diskfree deployment. The tool can be used for the deployment and administration of Linux clusters.

**F****favored dispatching**

A virtual machine performance option that lets an installation assign a given virtual machine an eligible list class of E0 and immediately add the virtual machine to the dispatch list. This option is available through the QUICKDSP option in the user directory and through the SET QUICKDSP command.

**favored execution**

A virtual machine performance option that lets an installation force the system to allocate more of its resource to a given virtual machine than would usually be the case.

**FBA** See fixed-block architecture.

**FC** See Fibre Channel.

**FCB** (1) See file control block.  
(2) See forms control buffer.

**FCP** See Fibre Channel Protocol.

**FDDI** See Fiber Distributed Data Interface.

**feature**

A part of an IBM hardware or software product that can be ordered separately by the customer.

**fence** A logical boundary that separates one or more paths or elements from the remainder of the logical DASD subsystem. This separation allows isolation of failing components so that they do not affect normal operations.

**fence state**

See Soft Fence.

**fetch protection**

A storage protection feature that determines right-of-access to main storage by matching the protection key associated with a main storage fetch reference with the storage keys associated with those frames of main storage.

**Fiber Connection (FICON®)**

An IBM fiber optic connection technology that coexists with and exploits ESCON equipment and infrastructure, but supports full duplex data transfers and enables greater throughput rates over longer distances. FICON is based on the Fibre Channel Protocol.

**Fiber Distributed Data Interface (FDDI)**

An ANSI standard for high-speed transmission over fiber optic cables.

**Fibre Channel (FC)**

A full duplex serial data transfer architecture, developed by a consortium of computer and mass storage device manufacturers and standardized by ANSI. This architecture allows for an active intelligent interconnection scheme, called a fabric, to connect devices.

**fibres channel connection**

See [Fibre Connection](#).

**Fibre Channel Protocol (FCP)**

A transport protocol for communicating SCSI commands over Fibre Channel networks.

**FICON**

See [Fibre Connection](#).

**FICON channel**

A channel that support FICON protocols.

**field** (1) An area in a window or panel into which a user can type text.

(2) The smallest identifiable part of a record.

**FIFO** See [first-in-first-out](#).

**FIFO special file**

A type of BFS file that is used to send data from one process to another and that persists beyond the life of the process. The receiving process reads the data first-in-first-out (FIFO). See also [unnamed pipe](#).

**file** (1) A named set of records stored or processed as a unit. (T)

(2) In OpenExtensions, an object that can be written to, or read from, or both. A file has certain attributes, including access permissions and type. See also [character special file](#), [directory](#), [FIFO special file](#), and [regular file](#),

(3) The act of storing a container of data, such as a file.

**file access mode**

A mode that indicates whether the file can be used read-only or read/write.

**file control block**

A control block used by OS Simulation to contain information about an I/O device or file, such as name, device type, format, and I/O control data.

**file control directory**

A type of SFS directory for which separate access authorities are granted to the directory and to the individual files in the directory. When the directory is accessed, changes to the directory become available as they are made. See also [directory control directory](#).

**file definition**

(1) Information that describes the contents and characteristics of a file.

(2) Equating a CMS file identifier (file name, file type, file mode) with an OS data set name using the FILEDEF command, or equating a DOS file ID with a CMS file identifier using the DLBL command.

**file descriptor**

In OpenExtensions, a per-process unique, nonnegative integer used to identify an open file for the purpose of file access.

**file identifier (file ID)**

The identifier of a CMS file that consists of a file name, a file type, and a file mode or directory identifier. The file ID is associated with a particular file when the file is created, defined, or renamed under CMS.

**file mode**

(1) The third field in a CMS file ID, which consists of 2 characters containing the file mode letter (A through Z) followed by the file mode number (0 through 6). The file mode letter indicates the minidisk or file pool directory on which the file resides. The file mode number indicates the access mode of the file. See also [CMS search order](#).

(2) A letter (A through Z) that CP assigns to all files on a disk that CP has accessed.

(3) In OpenExtensions, an object containing the file permission bits and other characteristics of a file. (P1)

**file mode extension**

A file mode that is accessed as a logical extension of another file mode; for example, minidisk C can be accessed as an extension of minidisk A. In the CMS search order, an extension of a file mode is searched before the next file mode in the alphabetical order; for example, if minidisk C is accessed as an extension of minidisk A, minidisk C is searched following minidisk A but ahead of minidisk B. When a minidisk is accessed as a file mode extension, the extension is automatically read-only. Therefore, a minidisk can be accessed as an extension of itself to protect the files on it from being accidentally overwritten.

**file mode number**

The second character (a number 0 through 6) of a CMS file mode, which can be used to control the usage of the file. The file mode number is assigned when the file is created and can be changed by the user. File mode number 1, indicating a file used for reading and writing, is assigned by default.

**file name**

(1) A name assigned to or declared for a file.

(2) The first field in a CMS file ID. A CMS file name is a 1- to 8-character field in which the valid characters are A through Z, a through z, 0 through 9, \$ (dollar sign), # (pound sign), @ (at sign), + (plus sign), - (hyphen), : (colon), and \_ (underscore). Although lowercase alphabetic characters are valid for use within the CMS file system, some CMS commands do not support file IDs that contain lowercase letters.

(3) In OpenExtensions, a name consisting of 1 - {NAME\_MAX} bytes used to identify a file. The characters composing the name may be selected from the set of all character values excluding the slash character and the null character. The file names dot and dot-dot have special meanings. See also [dot](#), [dot-dot](#), and [path name](#).

**file owner**

The user who has the highest level of access authority to a file, as defined by the file.

**file pool**

A collection of minidisks managed by a file pool server. It contains user files and directories and associated control information.

**file pool administration authority**

Authority that gives the user the ability to enter the file pool administration commands and that gives the user quasi-ownership of every object in the file pool. The user can do anything to directories, files, and other objects that their owners can do. The user cannot do something the owner cannot do. For example, the user cannot create an SFS alias in a owner's directory if the owner himself did not have any authorization to the base file.

**file pool administration machine**

A virtual machine with file pool administration authority.

**file pool catalog**

The part of a file pool that contains information about the objects stored in the file pool and the authorizations granted on those objects. See also [catalog storage group](#).

**file pool ID**

The name of a file pool. The file pool ID is part of a fully qualified directory name, identifying where the directory and all files in it are located and containing up to eight characters followed by a colon (:).

**file pool server machine**

A virtual machine that is configured to manage a file pool.

**file space**

A user's allocation of space within a file pool.

**file status table (FST)**

A table that describes the attributes of a file on a CMS minidisk, including file name, file type, file mode, date last written, and other status information.

**file system**

- (1) A collection of files and directories.
- (2) The collection of files and file management structures on a physical or logical mass storage device, such as a disk or disk partition. A single device can contain several file systems.
- (3) A mountable subtree of the directory hierarchy.
- (4) A collection of files and certain of their attributes. A file system provides a name space for file serial numbers referring to those files.
- (5) See also [mountable file system](#).

**file system command**

A CMS command that creates or manipulates CMS files.

**file system control block (FSCB)**

A collection of information about a CMS file, which CMS OS simulation commands and user programs use. A file system control block is established for a file by the FILEDEF command or the FSCB macro instruction.

**File Transfer Protocol (FTP)**

A TCP/IP protocol used for transferring files to and from foreign hosts. FTP also provides the capability to access directories. Password protection is provided as part of the protocol.

**file type**

- (1) The second field of a CMS file ID. A CMS file type is a 1- to 8-character field in which the valid characters are A through Z, a through z, 0 through 9, \$ (dollar sign), # (pound sign), @ (at sign), + (plus sign), - (hyphen), : (colon), and \_ (underscore). Although lowercase alphabetic characters are valid for use within the CMS file system, some CMS commands do not support file IDs that contain lowercase letters.
- (2) In OpenExtensions, one of the four possible types of files: regular file, directory, character special file, and FIFO special file.
- (3) See also [reserved file type](#).

**file type abbreviation (ftabbrev)**

The 3-character PTF abbreviation or the real CMS file type for a part that is not serviced by replacement.

**file type abbreviation table**

A system-level software inventory table that identifies the mapping between PTF-numbered file types and the real CMS file type. The service-level software inventory does not contain this table.

**FILECONTROL directory**

See [file control directory](#).

**filter** In CMS Pipelines, a stage that transforms data but does not move data into or out of the pipeline.

See also [device driver](#).

**filter package**

In CMS Pipelines, a file with a file type of MODULE that contains stages.

**firmware**

Licensed Internal Code (LIC) that is shipped with hardware. Firmware is considered an integral part of the system and is loaded and run at power on. Firmware is not open for customer configuration and is expected to run without any customer setup.

**first-in-first-out (FIFO)**

A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time. See also [last-in-first-out](#).

**first-level**

Pertaining to an operating system running in a logical partition. See also [second-level](#).

**first-level storage**

Real main storage. See also [second-level storage](#) and [third-level storage](#).

**fixed-block architecture (FBA)**

A DASD data storage architecture in which the data is stored in fixed-length blocks. Each block is addressed by a block number relative to the beginning of the particular file.

**fixed-length record**

A record having the same length as all other records with which it is logically or physically associated. See also [variable-length record](#).

**flag** (1) A modifier that appears on a command line with the command name that defines the action of the command.

(2) An indicator or parameter that shows the setting of a switch.

(3) A variable indicating that a certain condition holds.

(4) A character that signals the occurrence of some condition, such as the end of a word.

(5) An internal indicator that describes a condition to the processor.

**flat file**

A file that contains no control characters or tags.

**floating-point**

Describing a number in which the decimal point can occur in any position. In contrast, fixed-point numbers are defined as having a specific number of decimal places (as in currency) or none at all (integers).

**floating-point register**

A register that can manipulate data in a floating-point representation system.

**flush list**

A set of pages available to replenish the free list.

**force start**

A z/VM system restart that attempts to recover information about closed spool files previously stored on the checkpoint cylinders. All unreadable or incorrect spool file information is ignored. See also [checkpoint start](#), [clean start](#), [cold start](#), and [warm start](#).

**foreign host**

Any host on a network other than the host on which a user is working. See also local host.

**foreign node**

See foreign host.

**form feed**

A printer control that causes the printer to eject the current page.

**format**

- (1) The arrangement or layout of data in a data medium.
- (2) To prepare a DASD volume or minidisk for use by creating a data scheme for storage.
- (3) To use ICKDSF to prepare a DASD volume for use by CP.
- (4) In CMS, to use the FORMAT command to initialize a minidisk for use with CMS files, count or reset the number of cylinders on a minidisk, or write a label on a minidisk.

**formatted file**

A file displayed and arranged with particular characteristics, such as line spacing, headings, and number of characters and lines per page. See also unformatted file.

**forms control buffer (FCB)**

A buffer for controlling the vertical format of printed output.

**forward recovery**

The capability of a resource to recover all the data in the event of a media failure, including updates to the data that have been made since the last backup of the data.

**frame** (1) A block of information transmitted between two or more stations in the data link layer of a network. It includes delimiters, control characters, information, and checking characters.

(2) See page frame.

**frame table**

A table, contained in real memory, that contains the real memory locations of all currently defined pages.

**free list**

A list maintained by CP that points to a set of pages that can be allocated to satisfy both virtual machine and system page requests.

**free storage**

Storage that has not been allocated. The blocks of storage available for temporary use by programs or by the system.

**FSCB** See file system control block.

**FST** See file status table.

**ftabbrev**

See file type abbreviation.

**FTP** See File Transfer Protocol.

**full path name**

See absolute path name.

**full-duplex**

See duplex protocol.

**full-pack minidisk**

A minidisk that contains all of the addressable cylinders of a real DASD volume.

**full-screen CMS**

A CMS mode in which CMS is in a window and can take advantage of 3270-type architecture

and windowing support, and various classes of output are routed to a set of default windows. Also, users can type commands anywhere on the physical screen and scroll through commands and responses previously displayed. See also [windowing](#).

**full-screen editor**

An editor used at a display terminal where an entire screen of data is displayed at once and where the user can access the data through commands or by using a cursor.

**full-screen mode**

(1) A form of screen presentation in which the contents of an entire terminal screen can be displayed at once and in which the user can access data through commands or by using a cursor. Full-screen mode is often used for fill-in-the-blanks prompting. See also [line mode](#).

(2) An environment in which an entire 3270 display screen is under the control of a program running in a virtual machine.

**fullword**

A computer word, which is 4 bytes (32 bits).

**fully qualified directory name (dirname)**

The complete name of an SFS directory, which identifies the file pool ID, the user ID, and the parent directories.

**fully qualified LU name**

A name that identifies each LU in an SNA network. It consists of a network ID followed by a network LU name. See also [locally known LU name](#).

**fully qualified path name**

(1) A BFS path name that includes the identification of the byte file system. The path name starts with *../VMBFS:filepoolid:filepaceid/*, in which *../VMBFS:* is a keyword string (not case sensitive).

(2) A path name that identifies a file system or directory exported by a remote NFS server. The path name starts with *../NFS:foreign\_host/directory\_name*, in which *../NFS:* is a keyword string (not case sensitive).

**function key**

A key that causes a specified operation or sequence of operations to be performed when it is pressed. Used to refer to keys labeled *<Fn>*; for example, *<F1>*. See also [program function \(PF\) key](#).

**function level**

A number indicating the distribution release of a feature or facility.

**G****gadget**

A windowless graphical object that looks like its equivalent like-named widget but does not support the translations, actions, or popup widget children supplied by that widget.

**gateway**

(1) A functional unit that interconnects a local data network with another network having different protocols.

(2) A host that connects a TCP/IP network to a non-TCP/IP network at the application layer.

(3) The LU name of a VM system, CS collection, or TSAF collection that is a source for communications to an SNA-defined network or the target of communications from an SNA-defined network. See also [global gateway](#), [private gateway](#), and [system gateway](#).

(4) See also [router](#).

**gateway manager**

A virtual machine in which one or more gateways are active. AVS is a gateway manager.

**Gb (gigabit)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 30, or 1,073,741,824, bits.

(2) For disk storage capacity and communications volume, 10 to the power of 9, or 1,000,000,000, bits.

**GB (gigabyte)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 30, or 1,073,741,824, bytes.

(2) For disk storage capacity and communications volume, 10 to the power of 9, or 1,000,000,000, bytes.

**GC** See [graphics context](#).

**GContext**

See [graphics context](#).

**GCS** See [Group Control System](#).

**GCS command file**

A file with a file type of GCS that contains a series of commands. The file can also include other commands that are acceptable to GCS, such as CP commands, and REXX statements and functions. The file is invoked by issuing a command, which is the file name of the file.

**GCS exec**

See [GCS command file](#).

**GDDM<sup>®</sup> interface for X Window System (GDDMXD)**

A graphical interface that formats and displays characters, graphics, and images on workstation display devices that support the X Window System.

**GDDMXD**

See [GDDM interface for X Window System](#).

**GDF file**

See [graphics data format file](#).

**GDPS<sup>®</sup>**

See [Geographically Dispersed Parallel Sysplex<sup>™</sup>](#).

**general purpose register (GPR)**

See [general register](#)

**general register**

A register that can be used as base-address registers and index registers in address arithmetic and as accumulators in general arithmetic and logical operations.

**general user**

The user who controls functions associated with a particular virtual machine. This user type corresponds to IBM-defined CP privilege class G. See also [privilege class](#).

**Geographically Dispersed Parallel Sysplex (GDPS)**

An application that integrates Parallel Sysplex<sup>®</sup> technology and remote copy technology to enhance application availability and improve disaster recovery.

**GID** See [group ID](#).

**gigabit**

See [Gb](#).

**gigabyte**

See [GB](#).

**global aging list**

A list of frames used by the z/VM system's frame replenishment algorithm to maintain a reclaimable pool of frames to replenish the system's available frame lists.

**global character**

See wildcard character.

**global definitions entry**

A source directory entry that begins with the GLOBALDEFS statement. This entry contains global settings to be used by all virtual machine definitions.

**global gateway**

A gateway that programs outside a CS or TSAF collection can use to access global resources inside the collection, or that global resource manager programs can use to access resources outside a CS or TSAF collection. See also private gateway and system gateway.

**global minidisk**

In a z/VM SSI cluster, a minidisk that can be shared with users on any member of the cluster. See also local minidisk.

**global resource**

A resource accessible from anywhere within a CS or TSAF collection and whose identity is known throughout the collection. A shared file system file pool is an example of a global resource. See also local resource, private resource, and system resource.

**global resource manager**

An application that runs in a server virtual machine and identifies itself to the CS or TSAF collection as a global resource owner using the Identify system service (\*IDENT). See also local resource manager, private resource manager, and system resource manager.

**GPR** See general register.

**graphics context (GC)**

The storage area for graphics output. Used only with graphics that have the same root and depth as the graphics content.

**graphics data format file (GDF file)**

A picture definition in a coded order format used internally by the GDDM function and, optionally, providing the user with a lower-level programming interface than the GDDM application programming interface.

**group** (1) A collection of users who can share access authorities for protected resources.

(2) A list of names that are known together by a single name.

(3) A set of related records that have the same value for a particular field in all records.

(4) A series of records logically joined together.

(5) See single user group and virtual machine group.

**group configuration file**

A file, created by the GROUP command, that contains the blueprint for building the user's virtual machine group. The name of the file is *systemname* GROUP, where *systemname* is the name of the user's GCS saved system.

**Group Control System (GCS)**

A component of z/VM, consisting of a named saved system that the user can IPL and run in a virtual machine. One function of GCS is to provide a native VM/SNA network—a network that functions as part of the z/VM system without help form a second operating system. GCS is also a virtual machine supervisor and provides multitasking services. Applications that can run in a GCS virtual machine group include VTAM, AVS, RSCS, and NetView®.

**group ID (GID)**

A nonnegative integer assigned to a group of related POSIX users. The GID can often be substituted in commands that take a group name as an argument. A system user is assigned a GID in the user directory; if no GID is assigned, a default value is used. When the identity of a group is associated with a process, the GID value is referred to as a real GID, an effective GID, or an (optional) saved set-GID. See also [effective GID](#), [real GID](#), and [saved set-GID](#).

**group name**

A name that uniquely identifies a group of POSIX users to the system.

**group number**

See [group ID](#).

**guest** See [guest operating system](#).

**guest coupling simulation**

Software that simulates the hardware and software required to run a sysplex environment as second-level guests under z/VM.

**guest LAN**

An emulated local area network (LAN) segment that can be shared by guest virtual machines on the same z/VM system. See also [network interface card](#).

**guest operating system**

An operating system, such as Linux on IBM System z or z/OS, running in a virtual machine managed by the z/VM Control Program (CP). See also [host](#).

**guest real storage**

The storage that appears real to the operating system running in a virtual machine. See also [guest virtual storage](#), [host real storage](#), and [host virtual storage](#).

**guest virtual machine**

A virtual machine in which an operating system is running.

**guest virtual storage**

The storage that appears virtual to the operating system running in a virtual machine. See also [guest real storage](#), [host real storage](#), and [host virtual storage](#).

**H****half-duplex protocol**

A communication protocol in which data can be sent in only one direction at a time. See also [duplex protocol](#).

**halfword**

A contiguous sequence of bits or characters that constitutes half of a fullword and can be addressed as a unit.

**handle**

A temporary data representation that identifies a file.

**handshaking**

See [VM/VS handshaking](#).

**hard abend dump**

An automatic dump of the storage directly owned by CP. The dump is generated by CP during an abend that results in system termination and possibly a restart.

**hard link**

- (1) A mechanism that allows the **In** command to assign more than one name to a file. Both the new name and the file being linked must be in the same file system.
- (2) The relationship between two directory entries that represent the same file; the result of an execution of the **In** utility or the POSIX.1 **link()** function.

**hard requisite**

In VMSES/E, a prerequisite to a PTF that supplies a change required by the PTF. See also [corequisite](#) and [if-requisite](#).

**Hardware Configuration Definition (HCD)**

A program that creates and manages the hardware and software I/O configuration.

**Hardware Configuration Manager (HCM)**

A Windows-based graphical user interface to the Hardware Configuration Definition program.

**Hardware Management Console (HMC)**

A user interface through which data center personnel configure, control, monitor, and manage System z hardware and software resources. The HMC communicates with each central processor complex (CPC) through the Support Element (SE). On an IBM zEnterprise server, using the Unified Resource Manager on the HMCs/SEs, personnel can also create and manage an ensemble.

**hardware system area (HSA)**

A logical area of central storage, not addressable by application programs, used to store Licensed Internal Code and control information.

**HCD** See [Hardware Configuration Definition](#).

**HCM** See [Hardware Configuration Manager](#).

**HDLC** See [high-level data link control](#).

**header**

The portion of a message that contains control information.

**header file**

A file that contains constant declarations, type declarations, and variable declarations and assignments.

**HELP Facility**

A CMS facility that provides help information for the z/VM product. Applications and products that run on z/VM, as well as individual users, can also supply help information to be accessed and displayed through the HELP facility.

**heuristic damage**

The corruption of data integrity in protected resources. In a distributed system, this occurs when all of the following conditions are true:

- Parts of a transaction are disconnected from each other before synchronization point processing is complete
- Commit or rollback is forced to occur at one or more locations
- At least one location made a decision different from the others.

Heuristic damage can occur only through intervention from the outside, generally to force completion of a process that is holding locks on needed data. See also [heuristic decision](#).

**heuristic decision**

To unilaterally force a logical unit of work to commit or roll back. When distributed protected resources become disconnected during a coordinated update, and resynchronization is unable to effect a timely resolution because one or more of the resources is not active, it might be desirable to force a commit or rollback in order to free locked resources for further processing. A heuristic decision is typically made by the system operator, but it can also be a programmed decision.

**high common storage**

In GCS, common storage that resides above 16 MB. High common storage can be used only as free storage.

**high private storage**

In GCS, private storage that resides above 16 MB.

**high-level data link control (HDLC)**

In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC.

**high-level language (HLL)**

A programming language that does not reflect the structure of any particular computer or operating system.

**high-order**

The most significant; leftmost. For example, bit 0 in a register is the high-order bit.

**high-water mark**

The highest contiguous address, starting from location 0, where the virtual system's real addresses equal the virtual system's virtual addresses.

**HiperDispatch**

See z/VM HiperDispatch.

**HiperSockets**

(1) A hardware feature that provides high-speed TCP/IP communication channels between logical partitions (LPARs) on the same central processor complex (CPC). HiperSockets uses an adaptation of the queued direct I/O (QDIO) architecture.

(2) In z/VM, the virtualization of the HiperSockets channel, which provides high-speed communication between guest virtual machines. See also network interface card.

**HiperSockets bridge port**

A special port that is used to connect a z/VM virtual switch to a HiperSockets channel, essentially bridging the virtual switch's simulated network to a HiperSockets network. If the virtual switch also has an active OSA Uplink port, the HiperSockets channel is also bridged to the external physical network, providing external network connectivity for the HiperSockets Bridge Capable ports.

**history file**

(1) A CMS file that describes changes (with a date and time stamp) made to the z/VM system or its installed software products.

(2) In OpenExtensions, a file that lists the shell commands that have been processed. The default history file is `.sh_history`.

**HLL** See high-level language.

**HMC** See Hardware Management Console.

**HOLDING**

On a display terminal used as a virtual console under z/VM, a screen status that indicates the current contents of the screen will remain on the screen until the user requests that the screen be erased. See also screen status area.

**home directory**

The user's current directory after login or after issuing the `cd` command with no operand.

**hop count**

The number of gateways or routers through which a packet passes on its way to its destination.

**horizontal polarization mode**

An operational mode of a partition in which PR/SM™ distributes entitlements equally among all logical processors of the same type in the partition. See also vertical polarization mode.

**host** (1) A computer connected to a network, which provides an access point to that network. A host provides end-user services and can be a client, a server, or a client and server simultaneously. See also foreign host and local host.

(2) The z/VM Control Program (CP) in its capacity as manager of a virtual machine in which another operating system is running. See also [guest operating system](#).

(3) In TCP/IP, any system that has at least one Internet Protocol address associated with it.

**host access list**

A CP-maintained list which defines the address spaces that are available to an XC virtual machine when it is in access-register mode. Host access lists are not directly addressable by virtual machines. CP adds and removes entries from the host access list for the virtual machine when requested through z/VM services.

**host access-list entry**

The information in a host access list that identifies a particular address space and represents a virtual machine's capability to access it. An access-list-entry token (ALET) selects the host access-list entry to be used to access a particular storage operand.

**host logical storage**

The address space in which CP executes. This address space consists of three parts: the system execution area, the system execution space page management table, and the system frame table.

**host real storage**

The storage that appears real to the z/VM Control Program (CP). If z/VM is running in an LPAR, this storage is real storage; if z/VM is running in a virtual machine, this storage is virtual storage. See also [guest real storage](#), [guest virtual storage](#), and [host virtual storage](#).

**host virtual storage**

The storage that appears virtual to the z/VM Control Program (CP). See also [guest real storage](#), [guest virtual storage](#), and [host real storage](#).

**host-primary address space**

The initial address space for a virtual machine, created by CP at virtual machine logon. This is the address space from which instructions are fetched, and from which all operands are fetched and stored when in primary-space mode. For XA and ESA virtual machines, the host-primary address space represents virtual machine real storage. For XC virtual machines, the host-primary address space is one of a set of address spaces that the virtual machine may own.

**hot I/O**

A serious error condition caused by an I/O interruption that disrupts system operation.

**hot I/O rate**

In z/VM, the maximum rate of consecutive unsolicited interruptions allowed by CP from an I/O device.

**HSA** See [hardware system area](#).

**HyperPAV**

See [IBM HyperParallel Access Volume](#).

**HyperSwap<sup>®</sup>**

An IBM software technology that can substitute Peer-to-Peer Remote Copy (PPRC) secondary devices for PPRC primary devices and that is managed exclusively through Geographically Dispersed Parallel Sysplex (GDPS). HyperSwap is designed to swap a large number of devices quickly so that there is minimal impact to application availability.

**hypervisor**

A program that allows multiple instances of operating systems or virtual servers to run simultaneously on the same hardware device. A hypervisor can run directly on the hardware, can run within an operating system, or can be imbedded in platform firmware. Examples of hypervisors include PR/SM, z/VM, and PowerVM<sup>®</sup> Enterprise Edition.

**I**

**I/O adapter (IOA)**

A hardware and microcode element which, when combined with an integrated I/O processor, enables I/O devices to be attached to a processor without requiring a physical channel.

**I/O address**

An address provided by the channel subsystem (CSS) that consists of four parts: channel path identifier, subchannel number, device number, and device identifier. See also [channel path identifier](#), [subchannel number](#), [device number](#), and [device identifier](#).

**I/O configuration**

The collection of channel paths, control units, and I/O devices that attaches to the processor.

**I/O configuration data set (IOCDs)**

The data set that contains an I/O configuration definition built by the I/O configuration program (IOCP).

**I/O configuration program (IOCP)**

A program that defines to a system all available I/O devices and channel paths.

**I/O processor (IOP)**

A processor dedicated to controlling channels or communication links.

**I/O throttling**

The process of limiting or controlling the number of I/O operations that a guest operating system can initiate to a specific real device, thus preventing the guest from interfering with or dominating the I/O resources of that real device.

**I/O throttling list**

The list of real I/O devices whose I/O operations CP is currently limiting or controlling.

**I/O throttling rate**

The number of I/O operations per second that a specific real device can process from a guest operating system.

**IBM HyperParallel Access Volume (HyperPAV)**

A feature of IBM DASD subsystems that allows multiple I/O operations to a DASD through a pool of base and alias subchannels that are shared within a logical subsystem (LSS).

**IBM Integrated Facility for Linux (IFL)**

A System z specialty processor that provides additional processing capacity for Linux or OpenSolaris workloads. IFLs can be allocated only in a LINUX only or z/VM-mode LPAR.

**IBM Internal Coupling Facility (ICF)**

A System z specialty processor that allows multiple LPARs running z/OS to manage data and distribute workload in a Parallel Sysplex clustered system. See also [coupling facility](#).

**IBM System z Application Assist Processor (zAAP)**

A System z specialty processor that provides a Java™ execution environment, which enables Java-based web applications to be integrated with core z/OS business applications and backend database systems.

**IBM System z Integrated Information Processor (zIIP)**

A System z specialty processor that provides computing capacity for selected data and transaction processing workloads, and for selected network encryption workloads.

**IBM zEnterprise BladeCenter Extension (zBX)**

A heterogeneous hardware infrastructure that consists of a BladeCenter chassis attached to a zEnterprise server. A BladeCenter chassis can contain IBM blades or optimizers.

**| IBM zEnterprise Data Compression (zEDC)**

| A data compression acceleration solution designed for high performance, low latency  
| compression without significant CPU overhead. It is optimized for use with large sequential files

and it uses an industry standard compression library. zEDC can help to reduce disk usage, provide optimized cross-platform exchange of data, and provide higher write rates for IBM System Management Facilities (SMF) data.

**IBM zEnterprise System (zEnterprise)**

A heterogeneous hardware infrastructure that can consist of an IBM zEnterprise server and an attached IBM zEnterprise BladeCenter Extension (zBX), managed as a single logical virtualized system by the Unified Resource Manager.

**IBM zEnterprise Unified Resource Manager**

Licensed Internal Code (LIC), also known as firmware, that is part of the Hardware Management Console. The Unified Resource Manager provides energy monitoring and management, goal-oriented policy management, increased security, virtual networking, and data management for the physical and logical resources of a given ensemble.

**ICF** See IBM Internal Coupling Facility.

**ICMP** See Internet Control Message Protocol.

**ID card**

Under z/VM, the identification card that indicates the destination user ID of a deck of cards. These cards are read into the system card reader or into the card reader of an RSCS remote station.

**identity entry**

A source directory entry that begins with the IDENTITY statement. This entry begins a multiconfiguration virtual machine definition and includes statements common to all instances of that virtual machine.

**IDRC** See improved data recording capability.

**IEDN** See intraensemble data network.

**IEEE** See Institute of Electrical and Electronic Engineers.

**IEEE 802**

A set of networking standards developed by the IEEE, which includes standards for local area networks (LANs), Ethernet, and token-ring networks. In general, the IEEE 802 standards define physical network interfaces such as network interface cards, bridges, routers, connectors, cables, and all the signaling and access methods associated with physical network connections.

**IETF** See Internet Engineering Task Force.

**IFL** See IBM Integrated Facility for Linux.

**if-requisite**

- (1) In VMSES/E at the system-level, a requisite of an optional product.
- (2) In VMSES/E at the service-level, an out-of-component hard requisite that must be applied to an optional product. See also hard requisite and out-of-component requisite.

**IGMP** See Internet Group Management Protocol.

**IGP** See Interior Gateway Protocol.

**IIS** See initial installation system.

**image library**

A set of modules that defines the spacing, characters, and copy modification data that a printer needs to format and to print information. See also system data file.

**IMAP** See Internet Message Access Protocol.

**IML** (1) See initial machine load.

(2) See initial microcode load.

(3) See initial microprogram load.

**immediate command**

A type of CMS command that, when entered after an attention interruption, causes program execution, tracing, or terminal display to stop. Another immediate command can be entered to resume tracing or terminal display. The commands are called immediate commands because they are executed when they are entered; they are not stacked in the console stack. Within an exec, immediate commands can be established or canceled by the CMS IMMCMD command.

**implicit exclusive lock**

In SFS and BFS, an implicit lock that permits only one writer of an object and allows other users implicitly to lock the file as share, but not exclusive.

**implicit lock**

In SFS and BFS, an automatic lock that remains in effect until the file is closed and any changes have been committed or rolled back. See also explicit lock.

**implicit share lock**

In SFS and BFS, an implicit lock that permits multiple readers of an object and allows other users implicitly to lock the object as share or exclusive.

**implied CP command**

In CMS, a CP command invoked without preceding the command line with "CP". This capability is set by the CMS command SET IMPCP.

**implied EXEC**

In CMS, an exec procedure invoked without identifying it as such; that is, the word EXEC is not used for the invocation. Only the file name is used, as if entering a CMS command. This capability is set by the CMS command SET IMPEX.

**improved data recording capability (IDRC)**

A feature on IBM tape devices that allows them to read and write compacted recording formats.

**inactive page**

A page in real storage that has not been referenced during a predetermined period.

**inactive work unit**

A work unit that has no uncommitted work associated with it. Either no requests have yet been made, an atomic request was made, or requests were made and have been committed or rolled back.

**include file**

A file that contains preprocessor text, which is called by a program using a standard programming call.

**index** (1) To add or edit the attribute values that identify a specific item or object so that it can be retrieved later.

(2) A table that contains key values or references for locating information in an indexed file.

**indicator**

A 1-byte area of storage that contains either the character "1" to denote a true condition or the character "0" to denote a false condition.

**in-doubt resource**

In CRR, a protected resource that has successfully completed the first phase of the two-phase commit and is waiting for a decision from the initiator to either commit or roll back the changes and start the second phase of the two-phase commit.

**inherit**

To copy resources or attributes from a parent to a child; for instance, a parent process to a child process.

**initial access**

The main path CP uses when communicating between a partition and a channel path.

**initial installation system (IIS)**

In VMSES/E, a functional subset of the z/VM system that is shipped on the z/VM system DVD and system DDR tapes and is used during installation of z/VM.

**initial machine load (IML)**

A procedure that prepares a device for use.

**initial microcode load (IML)**

The loading of microcode from a storage medium into memory.

**initial microprogram load (IML)**

The loading of a microprogram, which is a set of elementary instructions maintained in special storage that corresponds to a specific computer operation.

**initial program load (IPL)**

The initialization procedure that causes an operating system to commence operation. In z/VM, after the IPL of the z/VM operating system, an operating system must be IPLed in each virtual machine to control that virtual machine's work. The virtual machine operating system can be CMS or a full-function operating system such as Linux on IBM System z, z/OS, or z/VM itself. Each virtual machine can have a different operating system. See also [IPL parameter](#).

**initialization**

- (1) The operations required for setting a device to a starting state, before the use of a data medium, or before implementation of a process.
- (2) Preparation of a system, device, or program for operation.
- (3) To set counters, switches, addresses, latches, or storage contents to zero or to other starting values at the beginning of, or at the prescribed points in, a computer program or process.

**initiator**

In Coordinated Resource Recovery (CRR), the role of the sync point manager when the transaction program issues a sync point request that begins the coordinated updating of distributed resources.

**INMN**

See [intranode management network](#).

**i-node** See [inode](#).

**inode** The internal structure that describes an individual file in OpenExtensions or Linux. There is one inode for each file. An inode contains the node, type, owner, access times, number of links, and location of a file. A table of inodes is stored near the beginning of a file system.

**inode number**

A number specifying a particular inode in the file system.

**input area**

An area on the screen that contains input fields where the user can enter information such as a user ID and password, or command text.

**input focus**

The position, indicated on the screen, where a user's interaction with the keyboard will appear.

**input mode**

The XEDIT environment in which a user can enter new lines of data. See also [edit mode](#).

**input redirection**

The specification of an input source other than standard input/output.

**input stream**

(1) A sequence of control statements and data submitted to an operating system by an input device.

(2) In CMS Pipelines, a stream that flows into a stage. See also [output stream](#), [primary input stream](#), [secondary input stream](#), and [tertiary input stream](#).

(3) In REXX, which defines only simple, character-based forms of input and output, the stream of characters from the user.

(4) The sequence of job control statements and data submitted to an operating system through an input unit especially started for this purpose by the operator.

**in-queue virtual machine**

A virtual machine in the dispatch list waiting to be dispatched.

**installation exit**

See [installation-wide exit](#).

**installation verification procedure (IVP)**

A procedure distributed with a program that exercises the newly installed program and verifies that the basic facilities of the program are functioning correctly.

**installation-wide exit**

A programming interface that can be used to modify or extend the functions of the z/VM system. Such modifications or extensions consist of exit routines written to replace one or more existing z/VM modules or to add one or more modules or subroutines to z/VM. Generally, an installation-wide exit is activated for all users on the system and is run as part of a system program. See also [user exit](#).

**Institute of Electrical and Electronic Engineers (IEEE)**

A professional society accredited by the American National Standards Institute (ANSI) to issue standards for the electronics industry.

**instruction counter**

See [program status word address](#).

**instruction simulation**

A process by which CP performs the software equivalent of a hardware instruction.

**instruction simulation wait**

A virtual machine wait state, in which the virtual machine waits for the completion of instruction simulation.

**integrated 3270 console**

A 3270 console device, provided through a Hardware Management Console, that can be used as a z/VM primary system operator console.

**integrated adapter**

(1) An integral part of a central processor complex that provides for direct connection of a device and uses neither a control unit nor the standard I/O interface.

(2) A combination of an integrated I/O processor and an I/O adapter.

**integrated ASCII console**

An ASCII console device, provided through a Hardware Management Console, that can be attached to a z/VM virtual machine.

**Integrated Services Digital Network (ISDN)**

A digital, end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

**intensive recording mode**

A mode of operation in which I/O errors are recorded for a specified device.

**interactive**

A classification given to a virtual machine that uses less than its allocation time slice because of terminal I/O. See also noninteractive.

**Interior Gateway Protocol (IGP)**

The protocol used to exchange routing information between collaborating routers in the Internet. The Routing Information Protocol (RIP) is an example of an IGP.

**intermediate server**

A program that handles communications requests to a resource manager program on behalf of a user program. The TSAF and AVS virtual machines are special types of intermediate servers called *communications servers*.

**internal trace table**

See CP trace table.

**International Organization for Standardization (ISO)**

An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

**internet**

A collection of interconnected networks that use the Internet suite of protocols. See also Internet and intranet.

**Internet**

The worldwide collection of interconnected networks that use the Internet suite of protocols and permit public access.

**Internet Control Message Protocol (ICMP)**

The protocol used to handle errors and control messages in the Internet Protocol (IP) layer of TCP/IP. Reports of problems and incorrect datagram destinations are returned to the original datagram source.

**Internet Engineering Task Force (IETF)**

The task force of the Internet Architecture Board (IAB) that is responsible for solving the short-term engineering needs of the Internet. The IETF consists of numerous working groups, each focused on a particular problem. Internet standards are typically developed or reviewed by individual working groups before they can become standards.

**Internet Group Management Protocol (IGMP)**

The protocol used by Internet Protocol (IP) hosts to report their host group memberships to multicast routers.

**Internet Message Access Protocol (IMAP)**

A protocol that allows a client to retrieve and manipulate mail messages that are stored on a server.

**Internet Protocol (IP)**

In the Internet suite of protocols, a connectionless protocol that routes data through a network or interconnected networks and acts as an intermediary between the higher protocol layers and the physical network. See also Transmission Control Protocol.

**Internet Protocol (IP) address**

A unique address for a device or logical unit on a network that uses the Internet Protocol standard. See also Internet Protocol version 4 and Internet Protocol version 6.

**Internet Protocol version 4 (IPv4)**

A version of the Internet Protocol that uses 32-bit addresses.

**Internet Protocol version 6 (IPv6)**

A version of the Internet Protocol that uses 128-bit addresses.

**internet router**

A device that forwards and routes IP datagrams from one network to another. Routers allow hosts on different networks to communicate with each other.

**Internet suite of protocols**

A set of protocols developed for use on the Internet and published as Requests for Comments (RFCs) through the Internet Engineering Task Force (IETF).

**internetwork**

See [internet](#).

**interprocess communication (IPC)**

In CMS multitasking, the exchange of information between processes or threads through messages placed on queues maintained by the kernel.

**interrupt**

- (1) A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed.
- (2) To stop a process in such a way that it can be resumed. See also [signal](#).

**Inter-System Facility for Communications (ISFC)**

A function of CP that provides communication services between transaction programs on interconnected z/VM systems. A group of interconnected domains consisting of z/VM systems that use ISFC to communicate with each other is known as a Communication Services (CS) collection. CP acts as the domain controller for all of the users who are defined in the directory of that system and authorized to use APPC/VM communications.

**inter-user communication vehicle (IUCV)**

A CP interface for passing data between virtual machines or between CP and a virtual machine.

**intraensemble data network (IEDN)**

A private high-speed network for application data communications within an ensemble. Data communications for workloads can flow over the IEDN within and between nodes of an ensemble. All of the physical and logical resources of the IEDN are configured, provisioned, and managed by the Unified Resource Manager.

**intranet**

A private network that integrates Internet standards and applications (such as Web browsers) with an organization's existing computer networking infrastructure.

**intranode management network (INMN)**

A private service network that the Unified Resource Manager uses to manage the resources within a single zEnterprise node. The INMN connects the Support Element (SE) to an IBM zEnterprise server and to any attached IBM zEnterprise BladeCenter Extension (zBX).

**intrinsic**

A collection of C language routines for building and using widgets. Intrinsic provide the ability to organize a collection of widgets into an application.

**IOA** See [I/O adapter](#).

**IOCDS**

See [I/O configuration data set](#).

**IOCP** See [I/O configuration program](#).

**IOP** See [I/O processor](#).

**IP** See [Internet Protocol](#).

**IP address**

See [Internet Protocol \(IP\) address](#).

**IP datagram**

In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

**IPC** See interprocess communication.

**IPL** See initial program load.

**IPL parameter**

An option for an initial program load passed to CP through the stand-alone loader.

**IPL processor**

The processor on which the control program is initialized.

**IPL volume**

The DASD volume that is IPLed and contains a loader.

**IPv4** See Internet Protocol version 4.

**IPv6** See Internet Protocol version 6.

**ISDN** See Integrated Services Digital Network.

**ISFC** See Inter-System Facility for Communication.

**ISO** See International Organization for Standardization.

**IUCV** See inter-user communication vehicle.

**IVP** See installation verification procedure.

**J****Japanese Industry Standard (JIS)**

A standard of coding character sets.

**JCL** See Job Control Language.

**JES** See Job Entry Subsystem.

**JES2** See Job Entry Subsystem.

**JES3** See Job Entry Subsystem.

**JIS** See Japanese Industry Standard.

**job control**

(1) Facilities for monitoring and accessing background processes.

(2) A facility that allows users to selectively stop (suspend) the processing of processes and continue (resume) their processing at a later point.

**Job Control Language (JCL)**

In z/OS and z/VSE<sup>®</sup>, a problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

**Job Entry Subsystem (JES)**

A z/OS subsystem that receives jobs into the system and processes all output data produced by the jobs. There are two types of Job Entry Subsystems: JES2 and JES3.

**K**

**Kanji** A graphic character set consisting of symbols used in Japanese ideographic alphabets. Each character is represented by 2 bytes.

**Katakana**

A character set of symbols used on one of the two common Japanese phonetic alphabets, which is used primarily to write foreign words phonetically.

**Kb (kilobit)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 10, or 1024, bits.

(2) For disk storage capacity and communications volume, 10 to the power of 3, or 1000, bits.

**KB (kilobyte)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 10, or 1024, bytes.

(2) For disk storage capacity and communications volume, 10 to the power of 3, or 1000, bytes.

**kernel** (1) Part of OpenExtensions containing programs for such tasks as I/O, management, and communication. See also [shell](#).

(2) Part of the Linux operating system that contains programs for such tasks as input/output, management and control of hardware, and the scheduling of user tasks.

(3) See also [reusable server kernel](#).

**keyword**

A part of a processing statement or command operand that consists of a specific character string.

**kilobit**

See [Kb](#).

**kilobyte**

See [KB](#).

**KornShell**

A command interpreter developed on UNIX, which, with the UNIX System V shell, forms the basis for the OpenExtensions shell.

**L**

**label** (1) In programming languages, one or more characters used to identify a statement or an item of data in a computer program.

(2) In CMS Pipelines, an alphanumeric string that defines how streams are connected between stages. See also [label definition](#) and [label reference](#).

**label definition**

In CMS Pipelines, the first occurrence of a label in a pipeline. A label definition makes it possible to connect other pipelines to the labeled stage. See also [label reference](#).

**label reference**

In CMS Pipelines, a second or subsequent occurrence of a label in a pipeline. A label reference defines an additional input and output stream for the stage on which the label is defined. See also [label definition](#).

**LAN** See [local area network](#).

**LAN broadcast**

The sending of a transmission frame intended to be accepted by all other data stations on the same LAN.

**LAN multicast**

The sending of a transmission frame intended to be accepted by a group of selected data stations on the same LAN.

**language binding file**

A programming language header file or assembler macro that aids in using an application

programming interface. The binding file defines the entry points, declares the data types of function parameters, maps long function names to external symbols, and defines constants for return codes, reason codes, and other values.

**Language Environment®**

A component of z/VM that provides the runtime environment for programs written in C/C++, COBOL, or PL/I.

**last agent optimization**

An optimized flow technique that is part of the LU 6.2 sync point architecture and that reduces the number of sync point flows in a sync point tree.

**last-in-first-out (LIFO)**

A queuing technique in which the next item to be retrieved is the item most recently placed in the queue. See also first-in-first-out.

**lasting lock**

See explicit lasting lock.

**LCU** See logical control unit.

**LDAP** See Lightweight Directory Access Protocol.

**library**

- (1) A collection of functions, calls, subroutines, or other data.
- (2) A data file that contains copies of a number of individual files and control information that allows them to be accessed individually.
- (3) A repository for demountable recorded media, such as magnetic disk packs and magnetic tapes.

**LIC** See Licensed Internal Code.

**Licensed Internal Code (LIC)**

Software provided for use on specific IBM machines and licensed to customers under the terms of IBM's Customer Agreement. Microcode can be Licensed Internal Code and licensed as such.

**LIFO** See last-in-first-out.

**Lightweight Directory Access Protocol (LDAP)**

A protocol that provides an easy way to maintain directory information in a central location for storage, update, retrieval, and exchange.

**limp mode**

A z/VM system participating in Coordinated Resource Recovery (CRR) without a CRR recovery server, which could result in some degradation in CMS performance.

**line delete symbol**

See logical line delete symbol.

**line editor**

An editor that displays data one line at a time and that allows data to be accessed and modified only by entering commands.

**line end symbol**

See logical line end symbol.

**line feed**

The movement of the print or display position to the corresponding position on the next line.

**line mode**

- (1) A form of screen presentation in which the information is presented one line at a time in the message area of the terminal screen, equivalent to using a typewriter-like terminal. See also full-screen mode.

(2) See [canonical mode](#).

**line printer client**

See [line printer remote](#).

**line printer daemon (LPD)**

The printer server that allows remote hosts to print on a local printer.

**line printer remote (LPR)**

A client that allows the local host to submit a file to be printed on a remote printer server.

**line printer requester**

See [line printer remote](#).

**lines per page**

The number of printed lines that a user wants to have on a page. For separator pages, lines per page defines the number of printable lines in the forms control buffer.

**link**

(1) In CMS, sharing a disk owned by another user. A user can share the disk on a temporary or permanent basis. The sharing is usually read-only and can require a password to access the data.

(2) A connection that provides the physical transfer of data from one node to another.

(3) In a file system, a connection between a directory and an object. The link is established when the object is created.

(4) In SNA, the combination of the link connection (the transmission medium) and two link stations (one at each end of the link connection).

(5) In TCP/IP, a term for a communications line. A TCP/IP link may share the use of a communications line with SNA.

(6) In RSCS, a connection, or ability to communicate, between two adjacent nodes in a network.

(7) In TSAF, the physical connection between two systems.

(8) In OpenExtensions, a new path name or directory entry for an existing file. The new directory entry can be in the same directory that holds the file or in a different directory. You can access the file under the old path name or the new one.

**linkage editor**

A computer program for creating load modules from one or more object modules or load modules by resolving cross-references among the modules and, if necessary, adjusting addresses. See also [binder](#).

**link-attached**

Pertaining to devices that are connected to a controlling unit by a data link. See also [channel-attached](#).

**link-edit**

To create a loadable computer program by means of a linkage editor.

**link-level addressing**

In an ESCON I/O interface, the level of addressing that identifies the channel path between a channel and a control unit. See also [device-level addressing](#).

**Linux on IBM System z**

An operating system supplied by a Linux distribution partner that executes on IBM System z hardware or in a z/VM virtual machine.

**listening mode**

A type of CRR participation in which the participating resource or product registers only for the postcoordination function, because the resource or product only needs to know when sync points occur and what their outcome is.

**little endian**

A format for storage or transmission of binary data in which the least significant bit (or byte) comes first. The reverse convention is big endian.

**live guest relocation**

A function of a z/VM SSI cluster that allows a running virtual server (guest virtual machine) to be moved dynamically from one member system to another.

**load** To move files from tape or DVD to disk, auxiliary storage to main storage, or minidisks to virtual storage within a virtual machine. See also initial machine load, initial microcode load, initial microprogram load, and initial program load.

**load map**

A map containing the storage addresses of control sections and entry points of a program loaded into storage.

**load module**

- (1) A computer program in a form suitable for loading into main storage for processing.
- (2) The output of a linkage editor.

**loadable unit**

In VMSES/E, a portion of a product that can be installed independently of the rest of the product, but is serviced as part of the product.

**loader** A routine, commonly a computer program, that reads data into main storage. See also stand-alone program loader.

**local** Pertaining to a device, file, or system that is accessed directly, without the use of a communication line. See also remote.

**local area network (LAN)**

A network located on a user's premises or within a limited geographical area, in which a set of devices are connected to one another for communication and that can be connected to a larger network. See also Ethernet and wide area network.

**local disk**

In VMSES/E, a minidisk or file pool directory containing local service, customized files, and any circumventive service.

**local host**

The computer on which a user is working. See also foreign host.

**local minidisk**

In a z/VM SSI cluster, a minidisk that can be shared with users only on one member of the cluster. See also global minidisk.

**local modification**

Any change that a user originates for a product or component. See also local service.

**local network**

The portion of a network that is physically connected to the host without intermediate gateways or routers.

**local program**

The conversation partner being discussed within a particular context. See also remote program.

**local resource**

A resource accessible only from within a single z/VM system and whose identity is known only within a single z/VM system in the CS or TSAF collection. See also global resource, private resource, and system resource.

**local resource manager**

An application that runs in a virtual machine and identifies itself to the local system in the CS or

TSAF collection as a local resource owner using the Identify system service (\*IDENT). See also global resource manager, private resource manager, and system resource manager.

**local service**

Change applied to a z/VM system that was not supplied by IBM as corrective service (COR) or as a recommended service upgrade (RSU). See also local modification.

**local string**

In VMSES/E, the set of local disks.

**local tracking number**

The unique identifier assigned to a local modification. The local tracking number is used in the file type of update files and in the update file identification records of auxiliary control files. Each installation has its own system of local tracking numbers.

**locale** In OpenExtensions, the definition of the subset of a user's environment that depends on language and cultural conventions. A locale defines such things as the definition of characters and the collation sequence of those characters.

**locally known LU name**

(1) The LU name that transaction programs use to identify a communications partner in a TSAF or CS collection.

(2) The LU name that transaction programs use to identify a remote LU in the SNA network. See also fully qualified LU name.

**lock** (1) The process by which integrity of data is ensured by preventing more than one user from accessing or changing the same data or object at the same time.

(2) A mechanism with which a resource is restricted for use by the holder of the lock.

(3) A means of serializing a sequence of events or serializing access to data.

(4) To exclude a page of virtual storage from future paging activity.

**locked page**

A page that is nonpageable and therefore remains in real storage until released.

**log data**

(1) Information that a communications program can send to its partner to help diagnose errors.

(2) In the shared file system, a record of changes made to the file pool so that if the system encounters a problem and stops in the middle of an operation, the file pool is not damaged.

(3) In CRR, records of the state of all resources participating in a CRR coordinated transaction.

**log minidisks**

(1) In file pools, two duplicate minidisks that contain information about changes made to the file pool. File pool servers use the file pool log minidisks to help protect the integrity of the file pool if a system failure occurs.

(2) In Coordinated Resource Recovery (CRR), two duplicate minidisks that contain information about the states of various logical units of work during synchronization point processing. CRR recovery servers use the CRR log minidisks to perform resynchronization processing to protect the integrity of the transaction if a failure occurs during synchronization point processing.

**logger** A functional unit that records events and physical conditions, usually with respect to time.

**logical character delete symbol**

A special editing symbol, usually the at (@) sign, that causes CP to delete it and the immediately preceding character from the input line. If many delete symbols are consecutively entered, that same number of preceding characters are deleted from the input line. The value can be redefined or unassigned by the installation or the user.

**logical control unit (LCU)**

A group of contiguous words in the hardware system area (HSA) that provides all of the information necessary to control I/O operations through a group of paths that are defined in the I/O configuration data set (IOCDs). Logical control units represent to the channel subsystem a set of control units that attach common I/O devices.

**logical device**

A representation of a real device.

**logical editing symbols**

Symbols that let the user correct entering errors, combine multiple lines of input on one physical line, and enter logical editing symbols as data. The logical editing symbols can be defined, reassigned, or unassigned by the user. See logical character delete symbol, logical escape symbol, logical line delete symbol, and logical line end symbol.

**logical escape symbol**

A special editing symbol, usually the double quotation ("), that causes CP to consider the immediately following character as a data character instead of a logical editing symbol.

**logical line**

A command or data line that can be separated from one or more additional command or data lines on the same physical input line by a logical line end symbol.

**logical line delete symbol**

A special editing symbol, usually the cent (¢) sign, that causes CP to delete the previous logical line in the physical input line back to and including the previous logical line end symbol.

**logical line end symbol**

A special editing symbol, usually the pound (#) sign, that lets the user enter the equivalent of several command or data lines in the same physical line; that is, each logical line except the last line is terminated with the logical line end symbol.

**logical operator**

A virtual machine (other than the virtual machine in which the programmable operator facility is running) designated to receive messages from the programmable operator and to perform actions on behalf of the programmable operator.

**logical partition (LPAR)**

A subset of a single system that contains hardware resources and operates as an independent system.

**logical processor**

In LPAR mode, a processor in a logical partition.

**logical record**

A formatted record that consists of a 2-byte logical record length and a data field of variable length.

**logical record interface (LRI)**

Builds logical records from record segments on input and breaks logical records into segments on output for the application. LRI is used with QSAM spanned records.

**logical saved segment**

A saved segment defined by CMS within a physical saved segment. A single physical saved segment can contain many logical saved segments. A logical saved segment can contain different types of program objects, such as MODULE files, TEXT files, execs, callable services libraries, language information files, user-defined objects, or a single minidisk directory. See also physical saved segment and saved segment.

**logical segment definition file**

A file that identifies the contents of a logical saved segment. The file name and file type must match a logical segment record in the physical segment definition file. The default file type is LSEG.

**logical unit (LU)**

An entity addressable within an SNA-defined network, similar to a node within a VM network. LUs are categorized by the types of communication they support. A TSAF collection in an SNA network is viewed as one or more LUs.

**logical unit name (LU name)**

A symbolic name given to a particular LU in an SNA-defined network.

**logical unit of work**

(1) In a file pool, a group of related operations that the file pool server performs for a user. The operations in a logical unit of work can be either committed or rolled back as a unit.

(2) In Coordinated Resource Recovery (CRR), a convenient abstraction for the application processing (including the underlying system support) performed to take a set of protected resources (such as file pools) from one consistent state to another (commit changes) in such a way that the unit of work appears atomic. If a failure occurs during the synchronization point processing of a logical unit of work, any changes made by (or for) the logical unit of work are rolled back, so that the protected resources are returned to their previous consistent state. A CRR logical unit of work is also known as a *transaction* or a *logical unit of work identifier (LUWID)* and consists of one or more LUWID instances. See also [logical unit of work identifier](#) and [LUWID instance](#).

**logical unit of work identifier (LUWID)**

The identifier of a CRR logical unit of work. The LUWID includes three parts: the fully qualified LU network name; the instance number, which is unique at the LU that creates it; and the sequence number, which is incremented by one following a sync point. Also, the conversation correlator is used to further qualify LUWIDs. See also [LUWID instance](#).

**logically connected terminal**

A terminal connected by leased or switched telecommunications or by local attachment to a multiple-access virtual machine by the CP DIAL command.

**logically sparse file**

A file that contains sparse records.

**logo configuration file**

A file that selects logo picture files for terminals and printers and text files for certain areas of the logon screen.

**logo picture file**

A file that contains the logon screen logo for a particular system.

**look-aside entry**

A nucleus resident routine that has been executed.

**low common storage**

In GCS, common storage that resides below 16 MB.

**low private storage**

In GCS, private storage that resides below 16 MB.

**low-order**

Least significant; rightmost. For example, in a 32-bit register (0–31), bit 31 is the low-order bit.

**LPAR** See [logical partition](#).

**LPAR mode**

In an activation profile for an LPAR, the type of LPAR that determines which operating systems you can run in the LPAR. z/VM supports ESA/390, LINUX-only, and z/VM modes.

**LPD** See line printer daemon.

**LPP** See lines per page.

**LPR** See line printer remote.

**LRI** See logical record interface.

**LU** See logical unit.

**LU name**

See logical unit name.

**LU type**

In SNA, the classification of an LU-LU session in terms of the specific subset of SNA protocols and options supported by the logical units (LUs) for that session.

**LU type 6.2**

A set of protocols and services defined by IBM's SNA for communication between application programs.

**LU-LU session**

In SNA, a session between two logical units (LUs). It provides communication between two end users, or between an end user and an LU services component.

**LUWID**

See logical unit of work identifier.

**LUWID instance**

A subset of a CRR logical unit of work (transaction) that is identified by the LUWID and represents the work done by a user ID for this LUWID. An LUWID instance consists of one or more resource logical units of work.

**M**

**MAC** (1) See mandatory access control.

(2) See Media Access Control.

**machine check**

An error condition that is caused by an equipment malfunction.

**machine ID**

A 2-byte field that uniquely defines a virtual machine within a GCS virtual machine group. The machine ID is sometimes combined with the task ID to identify uniquely a task within the virtual machine group.

**MACLIB library**

A library that contains macros, copy files, or source program statements for use under CMS.

**macro** See macrodefinition and macroinstruction.

**macro library**

A library of macrodefinitions.

**macrodefinition**

A set of statements that defines the name of, format of, and conditions for generating a sequence of assembler language statements from a single source statement.

**macroinstruction**

In assembler language programming, an assembler language statement that causes the assembler

to process a predefined set of statements called a macrodefinition. The statements usually produced from the macrodefinition replace the macroinstruction in the program.

**mail gateway**

A machine that connects two or more electronic mail systems (often different mail systems on different networks) and transfers messages between them.

**main storage**

Directly-addressable storage that provides high-speed processing of data by the processors and the channel subsystem. The terms "main storage" and "absolute storage" are used to describe storage that is addressable by means of an absolute address.

**Management Information Base (MIB)**

(1) A standard used to define SNMP objects, such as packet counts and routing tables, that are in a TCP/IP environment.

(2) In SNMP, a database of objects that can be queried or set by a network management system.

**mandatory access control (MAC)**

A security regimen that governs which subjects can access which objects, and in what way, based upon the relationship between their security labels. Mandatory access control restricts a subject's access to an object based upon the security label of the subject, the security label of the object, and the type of access the subject wants. See also discretionary access control.

**map**

In CMS, the file that contains a CMS output listing, such as (i) a list of macros in the MACLIB library, including macro size and location within the library; (ii) a listing of the directory entries for the DOS/VS system or private source, relocatable, or core image libraries; (iii) a linkage editor map for CMS/DOS programs; (iv) a module map containing entry point locations; and (v) a list of files dumped, loaded, scanned, or skipped by the TAPE or VMFPLC2 command.

**mapping**

The process of relating Internet Protocol addresses to physical addresses in the network.

**mask**

(1) A pattern of characters that controls the keeping, deleting, or testing of portions of another pattern of characters.

(2) To use a pattern of characters to control retention or elimination of another pattern of characters.

(3) See also subnet mask.

**master console**

In a system with multiple consoles, the console used for communication between the operator and the system.

**master file directory**

See CMS minidisk file directory.

**maximum transmission unit (MTU)**

The largest possible unit of data (in bytes) that can be sent on a given physical medium in a single frame.

**Mb (megabit)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 20, or 1,048,576, bits.

(2) For disk storage capacity and communications volume, 10 to the power of 6, or 1,000,000, bits.

**MB (megabyte)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 20, or 1,048,576, bytes.

(2) For disk storage capacity and communications volume, 10 to the power of 6, or 1,000,000, bytes.

**MCCU**

See multisystem channel communications unit.

**MDISK**

The user directory statement that defines a minidisk.

**Media Access Control (MAC)**

The method used by network adapters to determine which adapter has access to the physical network at a given time.

**megabit**

See Mb.

**megabyte**

See MB.

**member**

(1) A data object in a structure, such as a library.

(2) See SSI cluster member.

**member saved segment**

A special type of discontinuous saved segment that begins and ends on a page boundary and belongs to at least one and up to 64 segment spaces. A member saved segment is accessed by its own name or by the name of a segment space to which it belongs. A member saved segment may contain logical saved segments. See also discontinuous saved segment and segment space.

**memory**

Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent running or processing.

**memo-to-users**

A file provided on a product service medium that contains specific service information for a product. See also user memo.

**merge** When receiving files from a product service medium using VMFMRDSK, the process of moving current service files from each minidisk or file pool directory in the target string to the minidisk or directory that contains the previous service level. The result is that the primary target minidisk or directory is left empty and ready to receive the latest service.

**message**

(1) A communication from the system or program to a user that alerts the user to a problem, requests the user to do something, or provides the user with information. Most system messages have message identifiers. See also response.

(2) Data sent from a source application to a target application program.

**message header**

The leading part of a message that contains information such as the source or destination code of the message, the message priority, and the type of message.

**message identifier**

In z/VM, a unique alphanumeric string that identifies the origin of the message, the condition that caused the message, and its severity.

**message key**

In CMS multitasking, a part of the message text that describes or qualifies the message in some way.

**message level**

In CMS Pipelines, a number specifying which informational messages you can receive from CMS Pipelines. Depending on the message level, you can receive these messages along with other

messages. These messages help to determine what stage was running, what pipeline subcommand was running, and what pipeline was running that caused the previous message to be issued.

**message queue**

A list of messages awaiting processing or waiting to be sent to a terminal.

**message repository**

A source file that contains message texts for a z/VM component or user application. The source file is compiled into internal form by the GENMSG command. The message text in a repository file can be translated and used to support national languages.

**message text**

The part of a message without the header, control information, or identifier.

**metacharacter**

In OpenExtensions, a character that may have a special meaning in a regular expression. You can usually use a backslash in front of the character to remove the special meaning.

**MIB** See Management Information Base.

**microcode**

A set of hardware-level instructions that is implemented in a part of storage that is not program-addressable. See also firmware.

**MIF** See Multiple Image Facility.

**MIH** See missing interrupt handler.

**millicode**

In System z, microcode that implements functions that are either performance-critical or require direct control of the hardware. Among other tasks, millicode implements complex instructions and interrupt handling, and participates in system functions such as reset and recovery.

**minidisk**

A logical subdivision (or all) of a direct access storage device that has its own virtual device number, consecutive virtual cylinders (starting with virtual cylinder 0), and a VTOC or disk label identifier. See also full-pack minidisk, permanent minidisk, temporary minidisk, and virtual disk in storage.

**minidisk cache**

Memory or expanded storage used to improve access time to minidisk data.

**minidisk directory**

See CMS minidisk file directory.

**minidisk pool**

A logical DASD containing one or more DASD or minidisk extents that are formatted with 4 KB records. A minidisk pool is mapped to address spaces.

**minidisk-relative block number**

A 4 KB record contained on a minidisk or DASD extent with its origin relative to zero.

**minimum truncation**

The shortest form of a command name, operand, or option that can be entered and still be recognized by z/VM. For example, AC is the minimum truncation for the ACCESS command. See also abbreviation.

**mirrored DASD**

A long-distance remote copy option that updates a secondary copy of a DASD volume to match changes made to a primary volume. Although mirror processing is most often associated with disaster recovery or preparing for disaster recovery, it can also be used for everyday processing and data migration.

**missing interrupt handler (MIH)**

A z/VM facility that detects incomplete I/O conditions by monitoring I/O activity. It also tries to correct incomplete I/O conditions without operator intervention.

**mode** In OpenExtensions, a collection of attributes that specifies the type and access permissions for a file or directory.

**mode name**

Part of the CPI Communications side information that is used by LU 6.2 to designate the properties for the session that will be allocated for a conversation.

**module**

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

**MORE**

On a display terminal used as a virtual console under z/VM, a screen status that indicates when the user's display screen is full and there is more data to be displayed. See also screen status area.

**Motif** See OSF/Motif.

**mount** (1) To place a data medium, such as a tape, in a position to operate.

(2) In OpenExtensions, to make a byte file system (BFS) subdirectory tree, an entire BFS, or a Network File System (NFS) to be logically included as a component of a BFS at any place in the hierarchy, and therefore available to a virtual machine.

**mount point**

The path name of the directory on which the file system or subdirectory tree is mounted.

**mountable file system**

A file system stored in a BFS and therefore able to be logically mounted in another file system.

**MP** See multiprocessor.

**MPLF** See Multi-Path Lock Facility.

**MPRoute**

See Multiple Protocol Routing.

**MSS** See multiple subchannel set.

**MTU** See maximum transmission unit.

**multiconfiguration virtual machine definition**

A virtual machine definition that consists of an identity entry, any included profile entry, and all associated subconfiguration entries. In an SSI-enabled source directory, this virtual machine definition allows multiple virtual machine instances to be defined, which enables the user ID to be logged on concurrently and independently on multiple members of the SSI cluster. Each of these virtual machine instances can have a configuration different from the others.

**multiconnection server**

See concurrent server.

**Multi-Path Lock Facility (MPLF)**

A DASD facility that controls locking through a set of channel commands.

**Multiple Image Facility (MIF)**

A facility that allows channels to be shared among PR/SM logical partitions in an ESCON or FICON environment.

**Multiple Protocol Routing (MPRoute)**

A TCP/IP protocol that implements the OSPF protocol described in RFC 1583, 1058, and 1723.

**multiple subchannel set (MSS)**

A System z hardware facility that increases the total number of subchannels (device addresses) without expanding the 4-digit subchannel number. The MSS facility provides the ability to have sets of subchannels, with a current implementation of two sets. Each set provides 64,000 addresses. Subchannel set 0 can be used for any type of device. Subchannel set 1 can be used only for parallel access volume alias devices.

**multiple user mode**

In a file pool server machine, a processing mode in which the server processes user requests for file pool data. See also dedicated maintenance mode.

**multiplexer channel**

A channel designed to operate with a number of I/O devices simultaneously. Several I/O devices can transfer records at the same time by interleaving items of data.

**multiplexing**

In data transmission, a function that permits two or more data sources to share a common transmission medium so that each data source has its own channel.

**multiprocessor (MP)**

A computer using two or more processors under integrated control.

**multistream pipeline**

In CMS Pipelines, a pipeline that contains at least one stage that has more than one input stream, more than one output stream, or both.

**multisystem channel communications unit (MCCU)**

An I/O device that interconnects systems using block-multiplexer channels.

**multitasking**

- (1) A mode of operation that provides for the concurrent performance execution of two or more tasks.
- (2) In CMS, services that allow a program to subdivide itself into multiple independently executable parts, and coordinate these execution streams so they together accomplish the objective of the program.

**multivolume file**

A file that occupies more than one tape reel.

**multivolume tape support**

CMS support of tape switching. For this support, CMS does all necessary volume label processing. All volume switching is performed by either the default CMS volume switching routine (DMSTVS) or the DMSTVI tape label processing exit (if one has been provided by the user or installation).

**mutex** In CMS multitasking, a variable with an associated wait queue used to enforce mutual exclusion from a shared resource. CMS multitasking allows only one thread at a time to acquire the mutex, and only the thread that has acquired the mutex can execute the critical section of code for the resource. See also semaphore.

**N****name server**

In networking, the server that converts network names to addresses. See also domain name server.

**named pipe**

See FIFO special file.

**named saved system (NSS)**

The control program portion of an operating system that is assigned a name and saved in pageable format in a system data file. The purpose of a named saved system is to provide a

shared copy of an operating system that virtual machines can load by name. The shared copy requires less storage management than is used when each virtual machine loads its own copy of an operating system by device number. CMS can also be saved in and loaded from a named saved system.

**namedef**

A temporary name that represents either a file name and file type, or a directory name. Namedefs are used in program functions so that it is not necessary to directly code a file name and file type or directory name in the program.

**native mode**

Refers to running an operating system on the real machine instead of as a guest of z/VM.

**NAU** See network addressable unit.

**negative prerequisite**

In VMSES/E, a product that cannot exist on a system at the same time as another product.

**netdata**

The format of the data that is transmitted and received by the NETDATA command and that can be sent by the SENDFILE command.

**network**

(1) A physical configuration in which two or more computers, workstations, or printers are connected for the purpose of exchanging data.

(2) A logical organization overlaid on one or more physical networks. An internet is an example of a logical network.

**network adapter**

A physical device, and its associated software, that enables a processor or controller to be connected to a network.

**network addressable unit (NAU)**

In SNA networking, any device on the network that has a network address, including logical units, physical units, and system service control points.

**network administrator**

The person responsible for the installation, management, control, and configuration of a network.

**network elements**

As defined in the SNMP architecture, network elements are gateways, routers, and hosts that contain management agents responsible for performing the network management functions requested by the network management stations.

**network file system (NFS)**

A protocol developed by Sun Microsystems, Inc., that allows computers in a network to access each other's file systems. After accessed, the file system appears to reside on the local host.

**Network Information Center (NIC)**

In networking, an organization at SRI International that provides information about TCP/IP and the Internet. NIC assigns Internet Protocol addresses and acts as the repository for all Requests for Comments and TCP/IP documentation and policies.

**network interface card (NIC)**

(1) A printed circuit board that plugs into a server and controls the exchange of data over a network and provides the electronic functions for the data link protocol or access method, such as Ethernet.

(2) In z/VM, the simulation of a hardware HiperSockets adapter or OSA-Express QDIO-mode adapter. A guest operating system can use the NIC to connect to a guest LAN.

**network job entry (NJE)**

A set of formats and protocols that are used to transmit jobs, sysout data sets, operator

commands and operator messages, and job accounting information from one computing system to another. NJE is supported by JES2, JES3, RSCS, VSE/POWER, and BDT.

**network layer**

Layer 3 of the Open Systems Interconnection (OSI) model that defines protocols governing data routing.

**network management stations**

As defined in the SNMP architecture, clients that execute management applications to monitor and control network elements.

**newline character**

A control character that causes the print or display position to move to the first position on the next line.

**NFS** See network file system.

**NIC** (1) See Network Information Center.

(2) See network interface card.

**NJE** See network job entry.

**node** (1) In communications, an end point of a communication link or a junction common to two or more links in a network. Nodes can be processors, communication controllers, cluster controllers, terminals, or workstations. Nodes can vary in routing and other functional capabilities.

(2) A single IBM zEnterprise 196 (z196) and any optionally attached IBM zEnterprise BladeCenter Extension (zBX). A node can be a member of only one ensemble.

**node ID**

See node identifier.

**node identifier (node ID)**

The name by which a node is known to all other nodes in a network.

**nonblocking**

A way of requesting a service over an interface so that if the request cannot be completed immediately, the requesting process is able to continue and is not suspended. See also blocking.

**nonblocking mode**

If the execution of the program cannot continue until some event occurs, the operating system does not suspend the program until that event occurs. Instead, the operating system returns an error message to the program.

**noncanonical mode**

A **tty** input processing mode where input character erase and killing are eliminated, making input characters available to the user program as they are typed. See also canonical mode.

**noninteractive**

A classification given to a virtual machine that usually uses all of its allocated queue slice. See also interactive.

**noninteractive user**

A user whose virtual machine completed its previous queue slice without having an I/O interrupt from the terminal.

**nonprivileged program**

In GCS, a program called by a GCS application that operates in problem state. See also privileged program.

**nonresident page**

A page whose contents are on DASD but not in real storage. A page is considered nonresident when an attempt to load its real address returns a nonzero condition code.

**non-SSI source directory**

A source directory that does not contain the **SSI** option on the **DIRECTORY** statement and does not include any **IDENTITY**, **BUILD**, or **SUBCONFIG** statements.

**nonvolatile storage (NVS)**

Additional random access electronic storage available with certain DASD control units, used to retain data during a power failure.

**NOT ACCEPTED**

On a display terminal used as a virtual console under z/VM, a screen status that indicates that the user is: (1) trying to enter another command, but the terminal buffer still contains a previous command, or (2) using the copy function to copy the contents of the screen onto an associated hardcopy printer, but the printer is busy, nonexistent, or otherwise unavailable. See also screen status area.

**NSS** See named saved system.

**nucleus**

The part of CP, CMS, or GCS that is resident in main storage.

**NUCON**

The nucleus constant area of CMS.

**null character**

A character with all bits set to zero.

**null line**

A logical line with a length of zero designated by a return character as the first and only information.

**null string**

(1) A character string with a length of zero.

(2) In CMS Pipelines, a delimited string formed by two adjacent delimiters. See also delimited string.

**null value**

A parameter position for which no value is specified.

**NVS** See nonvolatile storage.

**0**

**object** (1) In computer security, a passive entity that contains or receives information. Access to an object implies access to the information it contains. Examples of objects in z/VM are: minidisks, spool files, named saved segments, and virtual storage. See also public object and subject.

(2) In VMSES/E, a usable form defined in build lists. A built part of a product. Examples of objects in z/VM are nuclei, modules, execs, help files, and macro libraries.

(3) In object-oriented programming, a single instance of a set of data and the operations that can act on that data.

**object code**

Compiler or assembler output that is executable machine code or is suitable for more processing to produce executable machine code. See also source code.

**object directory**

The form of the user directory that can be processed by CP. The object directory is created from the source directory by the **DIRECTXA** command. For CP to instantiate a virtual machine configuration, it must exist in the object directory.

**object module**

A module that is the output of an assembler or a compiler and is input to a linkage editor.

**object-oriented programming**

A type of programming in which code is divided into modules called objects that communicate with each other by passing messages.

**observer**

A virtual machine designated to receive the console activity of another without affecting the observed virtual machine.

**octet** A byte composed of eight binary elements.

**offline**

Pertaining to the operation of a device that is not under the control of a system. See also [online](#).

**one-phase commit protocol**

A protocol that is used by a sync point manager to commit a transaction when only resources that are managed as a single entity are changed during the transaction. For example, one-phase commit would be used when a single database on a single system is changed. If either the system or communications fail during a one-phase commit, the sync point manager might not be informed of whether the changes were committed or rolled back. This protocol is not implemented by the CRR sync point architecture. See also [two-phase commit protocol](#).

**online** (1) Pertaining to the operation of a functional unit or device that is under the control of the system or of a host. See also [offline](#).

(2) Pertaining to a user's access to a computer by way of a terminal.

(3) Pertaining to the state of being connected to the Internet, served by a service, or available through a system.

**online message**

The text that appears in the top area of a logon screen.

**open file**

(1) A file that has a temporary logical connection to a running program. The connection exists until the program closes the connection or the program terminates.

(2) In OpenExtensions, a file that is currently associated with a file descriptor.

**Open Shortest Path First (OSPF)**

An Interior Gateway Protocol that distributes routing information within a single autonomous system.

**open system**

(1) A system whose characteristics comply with standards made available throughout the industry and that therefore can be connected to other systems complying with the same standards.

(2) A system that implements specifications for interfaces, services, and supporting formats to enable properly engineered applications software to be ported with minimal changes across a wide range of systems.

(3) In computer security, a system in which resources that are not defined to the system are not protected.

**Open Systems Adapter Support Facility (OSA/SF)**

A host-based tool supplied with z/VM that allows you to customize an OSA's modes of operation.

**Open Systems Adapter-Express**

An integrated IBM System z hardware feature that combines the functions of an I/O channel with the functions of a network port to provide direct connectivity between System z applications and their clients on the attached network.

**Open Systems Adapter-Express Integrated Console Controller (OSA-ICC)**

An OSA-Express function that provides 3270 console sessions that can be used to IPL logical partitions or as system operator/master consoles for System z operating systems. The OSA-ICC also provides 3215 console sessions with the z/TPF master console.

**Open Systems Interconnection (OSI)**

The interconnection of open systems in accordance with specific ISO standards.

**OpenExtensions**

The implementation of certain Portable Operating System Interface for Computer Environments (POSIX) standards in z/VM to support an environment within which operating systems, servers, distributed systems, and workstations share common interfaces. The OpenExtensions POSIX interfaces are provided as C/C++ library routines in the C/C++ runtime library included in Language Environment. A language-neutral version of the POSIX functions is provided as a set of CMS callable services library (CSL) routines. OpenExtensions includes a POSIX-compliant file system known as the byte file system (BFS). CMS provides a set of OPENVM commands for managing BFS directories and files. The OpenExtensions shell provides a UNIX-like interactive user environment.

**OpenExtensions user ID**

See POSIX user ID.

**OpenStack**

An open source cloud computing platform for public and private clouds. z/VM enables the OpenStack API interface for other IBM deliverables.

**operand**

- (1) A string of characters entered with a command name or on a statement that is used to define the data on which a program operates or to control the execution of the program. See also parameter.
- (2) In the CMS environment, a string of characters entered with a command name to define the data on which the command processor operates. See also option.
- (3) An instruction field that represents data (or the location of data) to be manipulated or operated upon. Not all instructions require an operand field.

**operator console**

See primary system operator console. See also virtual console.

**optimized last agent**

See last agent optimization.

**optimizer**

A special-purpose hardware component or appliance that can perform a limited set of specific functions, with optimized performance when compared to a general-purpose processor. Because of its limited set of functions, an optimizer is an integrated part of a processing environment, rather than a standalone unit.

**option** (1) A string of characters entered with a command or on a statement that controls the execution of a program.

(2) In the CMS environment, a string of characters that control the execution of a command processor or program.

**ordinary application**

A GCS application, started with the OSRUN command, that operates in problem state. See also authorized application.

**OS/MVS simulation**

A set of programming interfaces defined by the MVS™ operating system and simulated by CMS.

**OSA** See Open Systems Adapter-Express.

**OSA/SF**

See [Open Systems Adapter Support Facility](#).

**OSA-ICC**

See [Open Systems Adapter-Express Integrated Console Controller](#).

**OSF/Motif**

An X Window System toolkit defined by Open Software Foundation, Inc. (OSF), which enables the application programmer to include standard graphic elements that have a 3-D appearance. Performance of the graphic elements is increased with gadgets and windowless widgets.

**OSI** See [Open Systems Interconnection](#).

**OSM** An OSA-Express channel path identifier (CHPID) type that provides connectivity to the intranode management network (INMN).

**OSPF** See [Open Shortest Path First](#).

**OSX** An OSA-Express channel path identifier (CHPID) type that provides connectivity to the intraensemble data network (IEDN).

**out-of-band data**

Data that is placed in a secondary channel for transmission. Primary and secondary communication channels are created physically by modulation on a different frequency, or logically by specifying a different logical channel. A primary channel can have a greater capacity than a secondary one.

**out-of-component requisite**

In VMSES/E, a PTF to one product or component that is required by a PTF to another product or component. See also [if-requisite](#).

**output display area**

For display devices, the upper portion of the display screen that contains an historical log of the most recent lines of console input to and output from the virtual machine and CP. On 3270 type displays, this area is protected; that is, the user is unable to enter information into the output display area. XEDIT and full-screen CMS do not use this display format.

**output file**

- (1) A file that a program opens so that it can write to that file.
- (2) A file that contains the results of processing.

**output redirection**

The specification of an output destination other than the standard one.

**output stream**

- (1) Messages and other output data that an operating system or a processing program displays on output devices.
- (2) In CMS Pipelines, a stream that flows out of a stage. See also [input stream](#), [primary output stream](#), [secondary output stream](#), and [tertiary output stream](#).
- (3) In REXX, which defines only simple, character-based forms of input and output, the stream of characters sent to the user.

**overhead**

The additional processor time charged to each virtual machine for the CP functions needed to simulate the virtual machine environment and for paging and scheduling time.

**overlay**

- (1) The technique of repeatedly using the same areas of internal storage during different stages of a program.
- (2) An accidental replacement of data in internal storage or on disk caused by a programming error.

**override**

- (1) A value that updates or replaces a previously defined value.
- (2) The process of implementing an override.

**override \$PPF**

See product parameter override file.

**override area**

See component override area.

**override file**

See class override file and product parameter override file.

**owner** The user who has the highest level of access authority to a data object or action, as defined by the object or action.

**P**

**pack** See disk pack.

**packet** A sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole.

**packet switching**

The act of sending and routing packets from source to destination based on information contained in their headers.

**packet-switched data network (PSDN)**

A network that uses packet switching as a means of transmitting data.

**page** A fixed-length block that has a virtual address and can be transferred between real storage and auxiliary storage.

**page frame**

A block of 4096 bytes of real storage that holds a page of virtual storage.

**page frame table**

See frame table.

**page locking**

Marking a page as nonpageable so that it remains in real storage until released.

**page number**

The part of a virtual storage address needed to refer to a page.

**page reclamation**

Making available and addressable the contents of a page in real storage that is waiting to be (or already has been) paged out. Page reclamation can occur after a page fault or after a request to fix or load a page, as long as no page-in operation has been scheduled for that particular page frame.

**page table**

A table that indicates whether a page is in real storage and that correlates virtual addresses with real storage addresses.

**page zero**

Storage locations 0 to 4095.

**page-aligned**

In fixed-block architecture, having a starting block number that is a multiple of eight and containing a number of blocks that is a multiple of eight.

**paging**

Transferring pages between real storage and auxiliary storage.

**paging activity indexes**

Values that affect the algorithm that CP uses in controlling the dispatching and scheduling of all active logged-on virtual machines.

**paging area**

An area of auxiliary storage (and an associated area of real storage) that CP uses for the temporary storage of pages when paging occurs.

**Parallel Access Volumes (PAV)**

A function that enables multiple concurrent accesses to a single DASD volume from a single host.

**parallel channel**

A channel that uses copper bus and tag cables as a transmission medium.

**parallel processing**

(1) The concurrent or simultaneous execution of two or more processes in a single unit.

(2) In CMS multitasking, the execution of threads of a particular application at the same time on different processors of the central processor complex.

**Parallel Sysplex**

A sysplex that uses one or more coupling facilities.

**parameter**

A value or reference passed to a function or program that serves as input or controls actions. The value is supplied by a user or by another program or process. See also operand.

**parameter driven installation (PDI)**

An installation method in which instructions for the installation are given at the beginning of the process rather than in response to prompts during the process.

**parameter list (PLIST)**

In CMS, a string of 8-byte arguments that call a CMS command or function. The first argument must be the name of the command or function to be called. General register 1 points to the beginning of the parameter list. See also extended PLIST.

**parent directory**

(1) The next higher-level directory in which the current directory is defined.

(2) In OpenExtensions, the directory that contains a directory entry for the given directory and is represented by the path name dot-dot in the given directory.

**parent process**

A process created to carry out a program. The parent process in turn creates child processes to process requests. See also child process and process.

**parent process ID**

An attribute of a new process (child process) after it is created by a currently active process (the parent process). The parent process ID of the child process is the process ID of the parent. When the parent process ends, the parent process ID is the process ID of an implementation-defined system process. In OpenExtensions, the parent process ID of the children of an ended process is set to the process ID of the INIT process, or 1.

**parked**

The state of an unused, but online, logical processor. A parked processor is not assigned work and remains in a wait state most of the time. See also unparked

**parm disk**

A disk that CP accesses during IPL. The parm disk contains all the system definition information required at IPL; for example, the system configuration and logo configuration files. See parm disk volume.

**parm disk volume**

The volume on which the parm disk resides.

**parse** To break down a string of information, such as a command or file, into its constituent parts.

**part** A CMS file provided on a z/VM installation or service medium and used as input to the build process. A part is the smallest serviceable unit of a component.

**part handler**

An exec provided by VMSES/E that builds a specific type of object or loads parts from service media.

**partition**

(1) A fixed-size division of storage.

(2) See logical partition.

**partitioned data set (PDS)**

A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**partner**

See conversation partner.

**parts catalog**

In VMSES/E, a set of software inventory files that catalog all parts of a product on a minidisk or file pool directory. All product parts are cataloged when they are loaded onto the system, when they are generated, and when they are moved.

**parts catalog table**

In VMSES/E, a table that identifies the product or component that owns all files residing on a disk and the VMSES/E command that last modified or created the file. The parts catalog table is modified each time VMSES/E modifies or creates a file on a disk identified in the :MDA section of the product parameter file.

**passive open**

The state of a connection that is prepared to provide a service on demand. See also active open.

**password envelope**

See envelope.

**password phrase**

A string consisting of mixed-case letters, numbers, and special characters, including blanks, that is used to control access to data and systems.

**password phrase envelope**

See envelope.

**patch** A circumventive service change applied directly to object code in a text deck.

**patch update file**

A file containing a single patch. The file can also specify requisites for applying the patch.

**path** (1) The route through a file system to a specific file.

(2) In a network environment, a route between any two nodes.

(3) In APPC/VM or IUCV, a named connection between two application programs that are on the same or different systems.

**path name**

(1) The identification of an object (such as a file or directory) in the OpenExtensions byte file system (BFS). The path name identifies the object within the BFS hierarchy by specifying the directories leading to the object. See also absolute path name, fully qualified path name, and relative path name.

(2) The identification of a file system or directory that can be exported by a remote Network File System (NFS) server and mounted on the local z/VM system. See also fully qualified path name.

**path name component**

The name of a BFS object, as specified in a path name.

**pattern**

(1) In REXX, the parts of a parsing template that allow a string to be split by a literal pattern or positional pattern. Parentheses may be supplied to create a variable pattern.

(2) A sequence of characters used, either with regular expression notation or for path name expansion, as a means of selecting various character strings or path names, respectively. The syntaxes of the two patterns are similar, but not identical.

(3) A sequence of characters used by commands that search for strings. Some characters have special meanings in patterns. For example, in the OpenExtensions grep command, \$ stands for the end of a line and abc\$ refers to the sequence *abc* appearing at the end of a line. Some patterns can be matched by many different strings.

**pattern matching**

The specification of a pattern of characters for search purposes.

**pattern-matching character**

See wildcard character.

PAV See Parallel Access Volumes.

**Pb (petabit)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 50, or 2,251,799,813,685,248, bits.

(2) For disk storage capacity and communications volume, 10 to the power of 15, or 1,000,000,000,000,000, bits.

**PB (petabyte)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 50, or 2,251,799,813,685,248, bytes.

(2) For disk storage capacity and communications volume, 10 to the power of 15, or 1,000,000,000,000,000, bytes.

PC See processor controller.

PCA See protected conversation adapter.

PDI See parameter driven installation.

PDR See persistent data record.

PDS See partitioned data set.

PDU See protocol data unit.

**peer system**

From the perspective of a particular system node in a network, an equivalent system; one with which jobs and data can be mutually exchanged.

**peer-to-peer communication**

Pertaining to data communications between two nodes that have equal status in the interchange. Either node can begin the conversation.

**Peer-to-Peer Remote Copy (PPRC)**

A remote-copy service that provides a synchronous or asynchronous copy of a volume or disk for disaster recovery, device migration, and workload migration.

PER See program-event recording.

**performance option**

One or more functions, such as favored execution (CP SET SHARE command) or favored

dispatching (CP SET QUICKDSP command), that can be assigned to a virtual machine to improve its performance, response time (if terminal-oriented), or throughput under z/VM.

### **Performance Toolkit for VM**

An optional feature of z/VM that gathers, analyzes, and displays z/VM performance data. It can also process Linux performance data obtained from the Linux performance gatherer, RMF™ PM.

### **permanent minidisk**

A shareable minidisk defined by an MDISK directory statement that exists even when not linked by a user.

### **permission**

(1) The ability to access a protected object, such as a file or directory. The number and meaning of permissions for an object are defined by the access control list.

(2) Authorization to perform activities, such as reading and writing local files, creating network connections, and loading native code.

### **persistent data record (PDR)**

A record that holds information about the state of each member of a z/VM SSI cluster.

### **petabit**

See Pb.

### **petabyte**

See PB.

### **PF key**

See program function (PF) key.

### **physical layer**

Layer 1 of the Open Systems Interconnection (OSI) model. The physical layer details protocols governing transmission media and signals.

### **physical saved segment**

One or more pages of storage that have been named and retained on a CP-owned volume (DASD). After created, it can be loaded within a virtual machine's address space or outside a virtual machine's address space. Multiple users can load the same copy. A physical saved segment can contain one or more logical saved segments. A system segment identification file (SYSTEM SEGID) associates a physical saved segment with its logical saved segments. See also logical saved segment and saved segment.

### **physical segment definition file**

A file that identifies the logical saved segments to be included within a physical saved segment. The file name is the name of the DCSS or member saved segment and the recommended file type is PSEG.

### **physical unit (PU)**

In SNA, the component that manages and monitors the resources, such as attached links and adjacent link stations, associated with a node.

### **physical unit block (PUB)**

In a VSE system, an entry in a table containing the channel and device address of a device. There is a physical unit block for each physical device available in the system.

**PID** See process ID.

### **pinned data**

Data that is held in a storage controller because of a permanent error condition. The data can be destaged to DASD or explicitly discarded. Pinned data exists only when using fast-write.

**pipe** (1) In OpenExtensions, an interprocess communication mechanism that connects an output file descriptor to an input file descriptor. Usually the standard output of one process is connected to the standard input of another, forming a pipeline.

(2) A connective symbol used between two commands on a command line, indicating that the output from the preceding command is to be used as the input for the following command.

(3) To direct the data so that the output from one process becomes the input to another process.

**pipeline**

In CMS Pipelines, a series of programs, called stages, each performing part of a task and passing the results to the next stage. Several parts of different tasks can be performed concurrently. See also [multistream pipeline](#), [pipeline subcommand](#), [subroutine pipeline](#).

**pipeline subcommand**

In CMS Pipelines, a subcommand that can be issued from a user-written stage or with the PIPCMD stage. Pipeline subcommands process data and interact with the calling pipeline.

**plaintext**

Text that is not encrypted.

**PLIST** See [parameter list](#).

**point-to-point channel path configuration**

In an I/O interface, a configuration that consists of a single link between a channel and one control unit. See also [switched point-to-point channel path configuration](#).

**polarization mode**

See and [horizontal polarization mode](#) and [vertical polarization mode](#).

**polling**

(1) Interrogation of devices to avoid contention, to determine operational status, or to determine readiness to send or receive data.

(2) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited one at a time to transmit.

**POOLDEF file**

A CMS file with a file type of POOLDEF that contains information that a file pool server and CRR recovery server use to locate all the minidisks in their file pools.

**pool-relative block number**

A 4 KB record contained within a minidisk pool with its origin relative to zero.

**POP** See [Post Office Protocol](#).

**POR** See [power-on reset](#).

**port** (1) The origin or destination of TCP/IP messages in a host. To be used, a port must have an associated socket.

(2) An endpoint for communication between devices, generally referring to a logical connection.

**port mapper**

A program that maps client programs to the port numbers of server programs. The port mapper is used with Remote Procedure Call (RPC) programs.

**port number**

In TCP/IP, a 16-bit number used to identify a logical connector between TCP or UDP and a higher-level protocol or application. Some protocols, such as the File Transfer Protocol (FTP) and the Simple Mail Transfer Protocol (SMTP), use the same port number in all TCP/IP implementations.

**portability**

The ability to use applications or files on different operating systems.

**portable character set**

The set of characters described in POSIX.2 that is supported on all conforming systems. See also [portable file name character set](#).

**portable file name character set**

The set of characters from which portable file names are constructed. For a file name to be portable across conforming implementations of POSIX.1, it shall consist only of the uppercase and lowercase characters of the alphabet (A through Z and a through z), the digits 0 through 9, the period (.), the underscore (\_), and the hyphen (-). The hyphen shall not be used as the first character of a portable file name. Uppercase and lowercase letters shall retain their unique identities between conforming implementations. In the case of a portable path name, the slash character may also be used.

**Portable Operating System Interface for Computer Environments (POSIX)**

An interface standard governed by the IEEE and based on UNIX. See also [OpenExtensions](#).

**portmap**

See [port mapper](#).

**positional parameter**

A parameter that must appear in a specified location relative to other positional parameters.

**POSIX**

See [Portable Operating System Interface for Computer Environments](#).

**POSIX user ID (UID)**

A fullword nonnegative integer that identifies a POSIX user. The UID is the sole basis for authority checking against such POSIX-defined resources as BFS files. A z/VM system user is assigned a UID in the user directory; if no UID is assigned, a default value is used. When the identity of the user is associated with a process, the UID value is referred to as a real UID, an effective UID, or an (optional) saved set-UID. See also [effective UID](#), [real UID](#), and [saved set-UID](#).

**POSIX user name**

The system login name that identifies a POSIX user. It is analogous to a user's z/VM user ID and is the lowercase version of the z/VM user ID.

**Post Office Protocol (POP)**

A protocol used for exchanging network mail.

**PostScript**

A standard that defines how text and graphics are presented on printers and display devices.

**power-on reset (POR)**

A key sequence that restarts the operating system (or other program) without turning off the electrical power of the system.

**PPF** See [product parameter file](#).

**PPRC** See [Peer-to-Peer Remote Copy](#).

**PR/SM**

See [Processor Resource/Systems Manager™](#).

**preferred auxiliary file**

An auxiliary file that applies to a particular version of a source module to be updated when multiple versions of the module exist.

**preferred paging area**

See [preferred paging device](#).

**preferred paging device**

A high-speed auxiliary storage device where frequently used pages are paged out.

**preferred paging volume**

See [preferred paging device](#).

**prefix area**

Positions on the XEDIT full-screen display in which prefix subcommands or prefix macros can be entered.

**prefix macros**

XEDIT macros entered in the prefix area of any line on a full-screen display.

**prefix storage area (PSA)**

Pages 0 and 1 in real storage assigned to a processor. Each processor's PSA is mapped to absolute storage.

**prefix subcommands**

XEDIT subcommands entered in the prefix area of any line on a full-screen display.

**prefixing**

The ability to assign the range of real addresses 0 - 8191 to a different block in absolute storage for each processor, thus permitting more than one processor sharing main storage to operate concurrently with a minimum of interference, especially in the processing of interruptions.

**prepared**

In an SFS file pool server, the state of a logical unit of work that is ready to be committed, but could not complete.

**prerequisite**

A change that must be applied to the system before another change can be applied. In VMSES/E, at the system level, a product that must be installed before another product can be installed. In VMSES/E, at the service level, a PTF that must be applied before another PTF can be applied.

**preventive service**

Program temporary fixes (PTFs), provided in a recommended service upgrade (RSU), for upgrading the current release of a z/VM system after it has been installed. See also selective preventive service.

**Preventive Service Planning (PSP) bucket**

Information about service updates available after the latest RSU.

**primary address space**

See host-primary address space.

**primary input stream**

In CMS Pipelines, the input stream defined by specifying a stage. The primary input stream is input stream number 0.

**primary output stream**

In CMS Pipelines, the output stream defined by specifying a stage. The primary output stream is output stream number 0.

**primary paging device**

The DASD with the highest effective data rate available for CP use. CP always attempts to store pages on this paging device instead of on slower paging devices if space on the primary paging device is available.

**primary system operator**

The IBM-defined CP privilege class A user who is automatically logged on during z/VM system initialization. If the current primary system operator logs off, the next class A user to log on becomes the primary system operator. See also system operator.

**primary system operator console**

The device where the primary z/VM system operator's virtual machine is logged on.

**primary user**

In SCIF, a disconnected virtual machine that is controlled by another virtual machine (called the secondary user).

**primary-space mode**

An address translation mode in which storage operands are always considered to reside in the host-primary address space. Access registers are not used to resolve addresses when in primary-space mode. The SET ADDRESS SPACE CONTROL and SET ADDRESS SPACE CONTROL FAST instructions can be used to change the PSW to enter and exit primary-space mode. See also [access-register mode](#).

**principal name**

The unique name of a user (client) or service.

**print inhibit mode**

A function that lets the user enter information on a terminal or display without causing the data to be typed or displayed.

**printer image library**

See [image library](#).

**private gateway**

A gateway that programs outside a CS or TSAF collection can use to access private resources inside the collection, or that nonglobal resource manager programs can use to access resources outside a CS or TSAF collection. See also [global gateway](#) and [system gateway](#).

**private minidisk**

A nonshareable minidisk.

**private resource**

A resource accessible from anywhere within a CS or TSAF collection or SNA network and whose identity is known only within a single workstation or virtual machine. See also [global resource](#), [local resource](#), and [system resource](#).

**private resource manager**

An application that runs in a server workstation or virtual machine and provides a service for connecting programs, but that does not identify itself to the CS or TSAF collection. See also [global resource manager](#), [local resource manager](#), and [system resource manager](#).

**private storage**

A combination of application code and GCS code available to only one particular virtual machine. No virtual machine can access or share another's private storage area. See also [high private storage](#) and [low private storage](#).

**privilege class**

The authorization required to use a subset of the CP commands and DIAGNOSE codes and certain CP system functions. The privilege class corresponds to a type of user. Each CP command and DIAGNOSE code belongs to one or more privilege classes. A user is assigned one or more privilege classes in the virtual machine definition for that virtual machine. Using MODIFY system configuration statements and MODIFY commands, the privilege class structure can be extended from the eight IBM-defined classes to as many as 32 classes. Creating a more elaborate privilege class structure gives an installation more control over the functions that each user can use. The privilege classes can be made more focused and specialized, increasing system integrity and security.

**privileged instruction simulation**

The handling of privileged instruction execution for virtual machine operating systems that execute as if they were in supervisor state under z/VM.

**privileged program**

In GCS, a program called by a GCS application that operates in supervisor state and uses privileged functions. A privileged program is one that meets either of the following requirements:

- It runs in an authorized virtual machine.
- It is called through the AUTHCALL facility.

See also [nonprivileged program](#).

**problem state**

A state during which the processor cannot execute I/O and other privileged instructions. See also [privileged instruction simulation](#) and [supervisor state](#).

**process**

- (1) A function being performed, or waiting to be performed, and the resources associated with it.
- (2) In CMS multitasking, a collection of threads performing related work.
- (3) In OpenExtensions, a program created by a spawn() function or spawn callable service, or a program that requests OpenExtensions services.

**process group**

A collection of processes that permits the signaling of related processes. Each process in the system is a member of a process group that is identified by a process group ID. A newly created process joins the process group of its creator.

**process group ID**

The unique identifier representing a process group during its lifetime. A process group ID is a positive integer that is not reused by the system until the process group lifetime ends.

**process ID (PID)**

The unique identifier that represents a process. A process ID is a positive integer and is not reused until the process lifetime ends.

**processable file**

- (1) A file that contains either a compiled and link-edited program or a shell script.
- (2) A regular file acceptable as a new process image file by the equivalent of the POSIX.1 exec family of functions, and thus usable as one form of a utility. The standard utilities described in POSIX.1 as compilers can produce processable files, but other unspecified methods of producing processable files can also be provided. The internal format of a processable file is unspecified, but a conforming application shall not assume a processable file is a text file.

**processable program**

A program suitable for processing by a computer. The program can be an application or a shell script.

**processable statement**

A statement that causes an action to be taken by the program. For example, to calculate, to test conditions, or to alter usual sequential processing.

**processor**

The part of a computer that interprets and executes instructions. Two typical components of a processor are a control unit and an arithmetic logic unit. See also [central processor](#), [specialty processor](#), [virtual processor](#).

**processor complex**

See [central processor complex](#).

**processor configuration token**

A 64-byte token used to synchronize the hardware's view of the I/O configuration with the software's view of the I/O configuration.

**processor controller (PC)**

Hardware that initializes and provides support and diagnostic functions for the central processor complex.

**Processor Resource/Systems Manager (PR/SM)**

A feature that provides logical partitioning of the real machine, which allows the machine to support several system control programs (SCPs) simultaneously.

**processor storage**

The machine storage available to the central processors and specialty processors.

**prodid**

See [product identifier](#).

**PRODPART file**

A file included on a product's installation media that contains information used by VMSES/E to update entries in the system-level software inventory each time the product is loaded onto the system.

**product**

Any separately installable software program distinct from others and recognizable by a product identifier.

**product identifier (prodid)**

The 7- or 8-character alphanumeric identifier assigned to the product by IBM. A product identifier is unique to a given product, but does not identify the release level of that product.

**product installation medium**

A tape, DVD, or electronic envelope containing components or products to load and build.

**product number**

See [product identifier](#).

**product parameter file (PPF)**

A file containing installation and service parameters for a product such as control options, minidisk and file pool directory assignments, and component part type/function lists.

**product parameter override file**

A file containing one or more component override areas.

**product processing exit**

An interface used by program products to perform additional product installation tasks.

**product service medium**

A tape, DVD, or electronic envelope containing service changes for one or more products.

**product service upgrade (PSU)**

A procedure used to upgrade the service level of a product or component using a recommended service upgrade (RSU).

**product tape**

See [product installation medium](#).

**production system**

An operating system running work in a virtual machine.

**profile**

- (1) Data that describes the characteristics of a user, group, resource, program, device, or remote location.
- (2) A file containing customized settings for a system or user.
- (3) In security, a description of the characteristics of an entity to which access is controlled.

**profile definition**

See [profile entry](#).

**profile entry**

A source directory entry that begins with the PROFILE statement. This entry contains a set of directory statements that can be included in any virtual machine definition.

**PROFILE EXEC**

A special exec procedure with a file name of PROFILE that a user can create. The procedure is usually executed immediately after CMS is loaded into a virtual machine (through the IPL CMS command).

**PROFILE GCS**

A REXX exec that identifies load libraries and GCS applications. The exec is executed when GCS is loaded into a virtual machine.

**program function (PF) key**

In z/VM, a keyboard key that can invoke a function programmed by the user or determined by an application program. Used to refer to keys identified by *PF*n**; for example, PF1. See also [function key](#).

**program stack**

Temporary storage for data being exchanged by programs that execute under CMS. See also [console stack](#).

**program state**

A state associated with each partner in a conversation. This state defines the functions that a communication program can issue at a given time.

**program status word (PSW)**

A collection of information used to control instruction sequencing and to hold and indicate much of the status of the processor in relation to the program currently being executed.

**program status word address (PSWA)**

The address portion in the program status word.

**program temporary fix (PTF)**

The code changes supplied by IBM to provide a temporary solution or bypass for a problem reported in an authorized program analysis report (APAR) or to supply new function between releases of a product.

**program-event recording (PER)**

A hardware facility used in debugging programs.

**programmable operator facility**

A CMS facility that provides automatic filtering and routing of messages sent to the virtual machine in which the programmable operator is running (for example, in the z/VM system operator's virtual machine).

**programmed symbol set**

A group of symbols that can be used in a field in a virtual screen.

**programming request for price quotation (PRPQ)**

A customer request for a price quotation on alterations or additions to the functional capabilities of system control programming or licensed programs. The PRPQ might be used in conjunction with computing system RPQs to solve unique data processing problems.

**projected working set**

An estimate of the number of pages of real storage that must be allocated to an in-queue virtual machine if it is to avoid excessive paging. The estimate determines whether the virtual machine can be added to the run list from an eligible list.

**prompt**

A displayed message that describes required input or gives operational information.

**protected conversation**

An APPC conversation that is allocated (initiated) with the SYNC\_LEVEL=SYNCPT option between two application programs. When one of the application programs issues a commit (or roll back), the CRR sync point manager uses a protected conversation to notify the other application program to issue a commit (or roll back). CRR processing handles the actual committing (or rolling back) of both of the application programs' work. Applications that use protected conversations must follow the rules of the LU 6.2 sync point architecture.

**protected conversation adapter (PCA)**

In CRR, a resource adapter that uses a protected conversation to communicate with a participating resource manager.

**protected resource**

(1) A resource that is updated in a synchronized manner during resource recovery processing.

(2) A resource for which there is either a RACF<sup>®</sup> profile or a RACF option that is controlling access.

**protocol**

A set of rules controlling the communication and transfer of data between two or more devices or systems in a communication network.

**protocol data unit (PDU)**

A set of commands used by the SNMP agent to request management station data.

**protocol suite**

A set of protocols that cooperate to handle the transmission tasks for a data communication system.

**PRPQ** See programming request for price quotation.

**PSA** See prefix storage area.

**PSDN** See packet-switched data network.

**pseudo page fault**

A facility available with VM/VS handshaking that lets a VSE virtual machine dispatch another task while waiting for a page-in request to be completed for some other task. Without this facility, the entire virtual machine would wait until the page request was satisfied, even if higher priority tasks were ready to execute.

**pseudo timer**

A special z/VM timing facility that provides date, time, virtual processor, and total processor time information to a virtual machine.

**PSP bucket**

See Preventive Service Planning bucket.

**PSU** See product service upgrade.

**PSW** See program status word.

**PSWA** See program status word address.

**PTF** See program temporary fix.

**PTF number**

An alphanumeric string assigned by a service organization to uniquely identify a PTF. PTFs for different products or different releases of a product have different PTF numbers.

**PTF parts list**

A file that identifies the file name and file type of each part required to install a PTF. A PTF parts list is shipped on the product service medium for each PTF that is included.

**PU** See physical unit.

**PUB** See physical unit block.

**PUBLIC**

In an SFS file pool, an authorization that includes all valid users of the system.

**public object**

An object that all subjects can access but *only trusted subjects* can modify. Since all subjects are allowed to access them, public objects must contain innocuous data.

## Q

**QDIO** See queued direct I/O.

### QSAM

See queued sequential access method.

### qualifier

A modifier or attribute that makes an item unique.

**queue** (1) A line or list of items waiting to be processed; for example, virtual machines waiting to run or messages waiting to be displayed. See also first-in-first-out and last-in-first-out.

(2) To arrange in or form a queue.

### queue add

The action by the system scheduler of placing a runnable virtual machine on the list of virtual machines that can be given control of a processor.

### queue drop

The action by the system scheduler of removing a virtual machine from the list of virtual machines that can be given control of a processor.

### queued direct I/O (QDIO)

A hardware channel architecture for direct data exchange with I/O devices, in which both the I/O device and the program running on the server reference main storage directly through a set of data queues. The QDIO architecture is used by Open Systems Adapter-Express (OSA-Express), HiperSockets, and Fibre Channel Protocol (FCP) channels.

### queued sequential access method (QSAM)

An extended version of BSAM. When this method is used, a queue is formed of input data blocks awaiting processing or processed output data blocks awaiting transfer to auxiliary storage or to an output device.

### quiet recording mode

The mode of operation in which system recovery machine checks are not recorded.

## R

**R/O** See read-only access.

**R/W** See read/write access.

**RACF** See Resource Access Control Facility Security Server.

**RARP** See Reverse Address Resolution Protocol.

### raw mode

See noncanonical mode.

**RC** See return code.

**rdev** A number that identifies a real device control block.

### reach-ahead service

Corrective service or local service that has been applied to a product but is not available on a recommended service upgrade.

### read authority

The authority to read the contents of a file without being able to change them. For a directory, read authority lets the user view the names of the objects in the directory.

### read/write access (R/W)

An access mode associated with a virtual disk or file pool directory that lets a user read, update, or write any file on the disk or file pool directory.

**READ/WRITE Rule**

Requirements for a subject's gaining read/write access to a particular object, which are:

1. The subject's security level must exactly equal the security level of the object.
2. The security categories of the subject must be exactly the same as those of the object.

These requirements prevent a subject from both writing down and reading up. See also READ-ONLY Rule and WRITE-ONLY Rule.

**reading backward**

Reading a file sequentially starting with the last record and working toward the first record. In a file of  $n$  records, the first input operation would obtain record  $n$  and the last input operation (assuming the entire file is read) would obtain record 1.

**reading up**

An event in which a subject with a given security label reads data from an object whose security label dominates that of the subject. See also Domination Rule and writing down.

**read-only access (R/O)**

An access mode associated with a virtual disk or file pool directory that lets a user read, but not write or update, any file on the disk or file pool directory.

**READ-ONLY Rule**

A requirement that the security label of the subject dominate that of the object. This prevents a subject from reading up. See also READ/WRITE Rule and WRITE-ONLY Rule.

**real address**

An address that identifies a location in real storage. When a real address is used to access main storage, the real address is converted by means of prefixing to an absolute address.

**real device control block**

A control block that describes a device. CP uses these control blocks to manage real and logical devices.

**real GID**

The POSIX group ID (GID) assigned to a POSIX user's login ID. See also effective GID and saved set-GID.

**real machine**

The physical processor, channels, storage, and I/O devices required for z/VM operation.

**real storage**

Storage consisting of byte locations sequenced according to their real addresses. Real storage is synonymous with absolute storage except for the effects of prefixing. Prefixing provides the ability to assign the range of real addresses 0 - 8191 to a different block in absolute storage for each processor, thus permitting more than one processor sharing main storage to operate concurrently with a minimum of interference, especially in the processing of interruptions.

**real system operator**

See system operator.

**real UID**

The POSIX user ID (UID) assigned to a POSIX user's login ID. See also effective UID and saved set-UID.

**reason code**

A value used to indicate the specific reason for an event or condition.

**receive**

- (1) To bring data into the specified buffer. The data can be sent to the user's virtual machine from another virtual machine or from the user's own virtual machine.
- (2) To load product files from a product installation medium.

(3) To load service files from a product service medium.

(4) In CMS multitasking interprocess communication, the action of retrieving a message from a queue.

**receive message log**

The file in which the VMFREC command writes status and error messages during receive processing.

**receive status table**

See [system-level receive status table](#) and [service-level receive status table](#).

**receiving virtual machine**

The virtual machine that receives messages or data from a source virtual machine.

**recommended service upgrade (RSU)**

Media containing preventive service for upgrading the current release of a z/VM system after it has been installed.

**recomp**

To change the number of cylinders or blocks on the disk that are available to you.

**record** (1) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them.

(2) A group of related data, words, or fields treated as a unit.

**recording mode**

A mode of operation in which system recovery machine checks are recorded. See also [quiet recording mode](#).

**recovery**

To maintain or regain system operation after a failure occurs.

**recovery machine**

In GCS, the first machine to join a virtual machine group. It has responsibility for executing routines that were set with the GCS MACHEXT macro and cleaning up system resources when machines leave the group.

**recovery server**

See [CRR recovery server](#).

**recursion**

A process involving numerous steps, in which the output of each step is used for the successive step.

**redirect**

To divert data from a process to a file or device to which it would not usually go.

**redirection**

Changing the association between files and file descriptors for a program. A process inherits file descriptors from the process that created the program (usually the shell). A file descriptor's standard input and standard output are usually associated with the keyboard and display screen, respectively. The shell can arrange for these descriptors (or any others) to be associated with other files before creating the new process. In particular, <infile redirects the standard input from infile, while >outfile redirects the standard output to outfile.

**reentrant**

The attribute of a program or routine that allows the same copy of a program or routine to be used concurrently by two or more tasks.

**region table**

A table used in dynamic address translation. Each entry in the table indicates a block of sequential virtual addresses spanning 2 GB and beginning at a 2 GB boundary.

**register**

An internal computer component capable of storing a specified amount of data and accepting or transferring this data rapidly. See also [general register](#), [access register](#), and [floating-point register](#).

**regression**

In VMSES/E, to cause serviced parts to go back to earlier levels. Regression can occur when applying changes from an RSU to parts updated by corrective service or user modifications.

**regular expression**

(1) A pattern (sequence of characters or symbols) constructed according to the rules defined in POSIX.2.

(2) A set of characters, metacharacters, and operators that define a string or group of strings in a search pattern.

(3) A string containing wildcard characters and operations that define a set of one or more possible strings.

**regular file**

A BFS file that is a randomly accessible sequence of bytes, with no further structure imposed by the system.

**relative path name**

A BFS path name that does not begin with a slash (/), which means that the path name is relative to the working directory. See also [absolute path name](#).

**relocation domain**

A set of members of an SSI cluster among which guest virtual machines can relocate without regard to differences in machine models, firmware, or features. See also [virtual architecture level](#).

**remote**

Pertaining to a system, program, or device that is accessed through a communication line. See also [local](#).

**| Remote Direct Memory Access over Converged Ethernet (RoCE)**

A network protocol that allows remote direct memory access over an Ethernet network. RoCE is a link layer protocol that allows communication between any two hosts in the same Ethernet broadcast domain. The IBM 10GbE RoCE Express feature can be used for LPAR-to-LPAR communication on a single server or server-to-server communication in a multiple-CPC environment.

**Remote Execution Protocol (REXEC)**

A protocol that allows the execution of a command or program on a foreign host. The local host receives the results of the command execution. This protocol uses the REXEC command.

**remote host**

Any host on a network other than the host on which a user is working. See also [local host](#).

**remote logon**

The process by which a terminal user establishes a terminal session with a remote host.

**remote name**

The name of an entity residing outside a file pool. Within a file pool, remote names are contained in external objects.

**Remote Procedure Call (RPC)**

A facility that a client uses to request the execution of a procedure call from a server. This facility includes a library of procedures and an eXternal Data Representation.

**remote program**

The program at the other end of a conversation with respect to the reference program. See also [local program](#).

**Remote Spooling Communications Subsystem Networking (RSCS)**

An optional feature of z/VM that supports the reception and transmission of messages, files, commands, and jobs over a computer network.

**replacement part**

See serviceable part.

**replacement service**

Servicing a part by replacing the part with a new one. See also update service.

**reply** (1) A response to an inquiry.

(2) In CMS multitasking interprocess communication, the action of sending a response to a message received from a queue.

**Request for Comments (RFC)**

One of a series of documents that covers a topic affecting internetwork communication. Some RFCs are established as internet standards.

**request for price quotation (RPQ)**

A customer request for a price quotation on alterations or additions to the functional capabilities of a hardware product for a computing system or a device.

**requester**

A virtual machine containing a user program that requests a resource. See also server.

**requisite**

A requirement of a product or PTF.

**requisite relationships**

The interrelated requirements of a product or PTF.

**requisite table**

The software inventory table that contains the requisite relationships between products in the system level and between PTFs in the service level. The file type of the system level inventory table is SYSREQT and the file type of the service-level software inventory table is SRVREQT.

**reserved file type**

(1) A file type recognized by the CMS editor as having specific default attributes that include record size, tab settings, truncation column, and uppercase or lowercase characters associated with that particular file type. The CMS editor creates a file according to these attributes.

(2) A specific file type used by a CMS command. Some CMS commands search for and use particular file types or create one or more files with a particular file type.

(3) A file with a specific file type that is supplied as part of the z/VM product. For example, the file type HELPCP is reserved for help files for CP commands.

**reserved page frames option**

A virtual machine option that reserves a specific number of page frames for one virtual machine. Generally, this option lets the most recently active pages of storage in a virtual machine remain allocated in real storage.

**resolver**

A program or subroutine that obtains information from a name server or local table for use by the calling program.

**resource**

A program, file, specific set of files, device, or any other entity or a set of entities that can be uniquely identified for processing in a z/VM system.

**Resource Access Control Facility (RACF) Security Server**

An optional feature of z/VM that provides for access control by identifying and verifying the

users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

**resource adapter**

Resource manager code that runs in an application's virtual machine. This is the interface between an application program and resource manager. If the adapter represents a resource that is to be protected by the Coordinated Resource Recovery (CRR) facility, then the adapter must register the resource with the sync point manager (SPM), handle various coordination exits that are driven by the SPM, and pass return codes back to the SPM.

**resource ID**

A 1- to 8-character name that identifies a resource or transaction program.

**resource logical unit of work**

See logical unit of work.

**resource manager**

An application running in a server virtual machine that directly controls one or more z/VM resources. There are four categories of z/VM resource managers: global, local, private, and system. Also, a resource manager (such as the SFS file pool server), can participate in CRR (Coordinated Resource Recovery).

**resource records**

Individual records of data used by the Domain Name System. Examples of resource records include the following: a host's Internet Protocol addresses, preferred mail addresses, and aliases.

**response**

Normal command output, which provides information about the execution or effect of a command or which provides requested data. Unlike system messages, command responses do not have message identifiers. See also message.

**response time**

The time between the submission of an item of work to a computing system and the return of results.

**restricted saved segment**

A segment space or discontinuous saved segment that can be accessed by a virtual machine only if the virtual machine definition for the virtual machine contains a NAMESAVE control statement that specifies the name of the segment space or discontinuous saved segment.

**restricted saved system**

A saved system that can be accessed by a virtual machine only if the virtual machine definition for the virtual machine contains a NAMESAVE control statement that specifies the name of the saved system.

**Restructured Extended Executor (REXX) language**

A general-purpose programming language, particularly suitable for EXEC procedures, XEDIT macros, or programs for personal computing. Procedures, XEDIT macros, and programs written in this language can be interpreted by the REXX/VM Interpreter or compiled by the REXX compiler.

**resync** See resynchronization.

**resynchronization**

A CRR function that is performed by the CRR recovery server when there has been a failure during synchronization point processing for a transaction. Resynchronization, which involves exchanging log names and comparing logical unit of work states, automatically attempts to complete the sync point process for the transaction. The goal of resynchronization is to maintain a consistent state (data integrity) among the protected resources involved in a transaction. Resynchronization may complete after the application ends. In very rare cases, such as an irrecoverable media failure or an operator error, resynchronization cannot complete and CRR allows operator intervention to complete the transaction.

**resynchronization initialization**

The initial exchange of log names that a participating resource manager initiates with the CRR recovery server before participating in a sync point.

**resynchronization recovery**

The exchange of log names and compare states flows that the CRR recovery server initiates to ensure consistent completion of a sync point by all protected resources for which data was logged.

**return code (RC)**

A value returned by a program to indicate the result of its processing.

**reusable server kernel**

In CMS, a package that consists of text library routines and a macro library of function prototypes and constant definitions that can be used to construct a server program.

**Reverse Address Resolution Protocol (RARP)**

A protocol that maintains a database of mappings between physical hardware addresses and IP addresses.

**reverse video**

A form of highlighting a character, field, or cursor by reversing the color of the character, field, or cursor with its background.

**revoked ALE**

The state of a host access-list entry that exists if the entry designates an address space that was subsequently destroyed, or for which the owner has revoked access permission. If a virtual machine attempts to use a host access-list entry that is in the revoked state, an addressing-capability exception is recognized.

**revoked alias**

A file pool alias that no longer points to a base file because authorization on the base file was revoked from the owner of the alias. Aliases may also be revoked if the storage group in which the alias resides is restored and the base file, which resides in another storage group, does not exist.

**REXEC**

See Remote Execution Protocol.

**REXX** See Restructured Extended Executor language.

**REXX exec**

A CMS file with a file type of EXEC that contains a procedure consisting of statements in the Restructured Extended Executor language. The exec can be interpreted by the REXX/VM Interpreter or compiled by the REXX compiler. If the SET IMPEX setting is ON, the exec is treated as a command, and the procedure is invoked by entering the file name of the file; otherwise, the EXEC command must be used.

**REXX language**

See Restructured Extended Executor language.

**REXX program**

See REXX exec.

**REXX/VM**

A component of z/VM that contains the REXX/VM Interpreter, which processes the REXX language.

**REXX/VM Interpreter**

The language processor of the z/VM operating system that processes procedures, XEDIT macros, and programs written in the REXX language.

**RFC** See Request for Comments.

**right align**

See [right justify](#).

**right justify**

(1) To control the positions of characters on a page so that the right-hand margin of the printing is regular.

(2) To shift the contents of a register or field so that the character at the right-hand end of the data is at a specific position.

**ring of files**

The arrangement of files in virtual storage when multiple files are being edited by XEDIT.

**RIP** See [Routing Information Protocol](#).

**RMF PM**

A Linux performance data gatherer that supplies data that can be processed by the Performance Toolkit for VM.

**RoCE** See [Remote Direct Memory Access over Converged Ethernet](#).

**roll back**

In the context of file pools and CRR, undoing changes that were made to a resource (such as a file).

**rollback**

In CRR (Coordinated Resource Recovery), the action taken by CRR for an application program (or transaction program) to initiate CRR backout processing to undo updates to protected resources (such as file pools) during a transaction. See [backout](#).

**root** (1) See [root directory](#).

(2) The user name for the system user with the highest authority.

**root directory**

(1) The highest directory in the BFS file system hierarchy. It is referred to as “/”.

(2) The directory used to resolve path names that begin with a slash (/).

(3) See also [working directory](#).

**root file system**

The basic file system, onto which all other file systems can be mounted. The root file system contains the operating system files that get the rest of the system running.

**root process**

In CMS multitasking, a process that performs session initialization and creates a number of threads dedicated to performing standard system functions or managing shared facilities.

**rotational position sensing (RPS)**

A feature of IBM disk storage devices that lets these devices disconnect from a channel during rotational positioning operations, thereby letting the channel service other devices.

**route** A connection to another system by a logical link and one or more intermediate systems.

**RouteD**

A server in the TCP/IP suite that uses the Routing Information Protocol to route packets dynamically. z/VM does not supply a RouteD server. Instead, z/VM provides dynamic routing through the MPRoute server.

**router** A device that connects networks. A router is protocol-dependent and connects only networks operating the same protocol. Routers do more than transmit data; they also select the best transmission paths and optimum sizes for packets. See also [gateway](#).

**Routing Information Protocol (RIP)**

The protocol that maintains routing table entries for gateways, routers, and hosts.

### **routing table**

(1) The table holding a list of valid paths through which hosts can communicate with other hosts. The routing table can hold static routes and dynamic routes.

(2) A Conversational Monitor System (CMS) file that contains the information that controls the operation of the programmable operator facility. The routing table determines which actions, if any, the programmable operator facility takes when receiving a message. The routing table also specifies authorized users of the programmable operator functions.

**RPC** See Remote Procedure Call.

**RPQ** See request for price quotation.

**RPS** See rotational position sensing.

**RSCS** See Remote Spooling Communications Subsystem Networking.

**RSU** See recommended service upgrade.

### **RUNNING**

On a display used as a virtual console under z/VM, a screen status indicating that the user's virtual machine is in control (but not necessarily executing a program or command) and that the terminal can receive messages. See also screen status area.

## **S**

**SAA** See Systems Application Architecture.

### **SAA Communications Interface**

See Common Programming Interface for Communications.

### **SAA Resource Recovery Interface**

The routines that allow applications to initiate a sync point to either commit or back out (roll back) all updates. The SRRCMIT routine commits and the SRRBACK routine backs out the updates.

**SAC** See storage access component.

**SAN** See storage area network.

**SAPL** See stand-alone program loader.

### **saved segment**

An area of virtual storage that is assigned a name and saved. Segment spaces, member saved segments, and discontinuous saved segments are defined by CP and saved in system data files. Logical saved segments are defined by CMS. A saved segment can be attached to and detached from a virtual machine and can be shared by many virtual machines. See also discontinuous saved segment, member saved segment, logical saved segment, physical saved segment, and segment space.

### **saved set-GID**

The POSIX group ID (GID) of a file or program, assumed from the owner, which gives the file or program the owner's privileges. When a user uses the file or program, the user's effective GID is changed to the saved set-GID of the file or program while it is being used. See also effective GID and real GID.

### **saved set-UID**

The POSIX user ID (UID) of a file or program, assumed from the owner, which gives the file or program the owner's privileges. When a user uses the file or program, the user's effective UID is changed to the saved set-UID of the file or program while it is being used. See also effective UID and real UID.

### **saved system**

See named saved system.

**SBCS** See single-byte character set.

**scale** A line on the XEDIT full-screen display, used for column reference.

**SCIF** See Single Console Image Facility.

**SCMBK**

See subchannel measurement block.

**SCP** See system control program.

**screen attribute byte**

A character position on the screen of a display terminal that defines the characteristics of the next field displayed on the screen; for example, protected, not protected, displayable, or nondisplayable.

**screen status area**

On a display terminal used as a virtual console under z/VM, an area located in the lower right of the display screen that indicates the current status of the display screen.

**SCSI** See Small System Computer Interface.

**SDF** See system data file.

**SDLC** See Synchronous Data Link Control.

**SDO** See z/VM System Delivery Offering.

**search order**

(1) The rules by which a system searches for names, files, or objects.

(2) See CMS search order.

**search path**

The sequence of directories that a command interpreter searches to find the program that the user wants to run.

**secondary input stream**

In CMS Pipelines, an input stream defined for a stage that has a label definition by the first reference to the label, or an input stream defined by the ADDSTREAM pipeline subcommand. The secondary input stream has input stream number 1.

**secondary output stream**

In CMS Pipelines, an output stream defined for a stage that has a label definition by the first reference to the label, or an output stream defined by the ADDSTREAM pipeline subcommand. The secondary output stream has output stream number 2.

**secondary user**

In SCIF, a user who is designated to receive a disconnected user's console messages and to enter commands to the disconnected user's console.

**second-level**

Pertaining to an operating system running in a virtual machine. See also first-level and guest operating system.

**second-level storage**

The storage that appears real to a virtual machine. See also first-level storage and third-level storage.

**Secure Sockets Layer (SSL)**

A cryptographic protocol that provides communication security over the Internet. With SSL, client/server applications can communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. See also Transport Layer Security (TLS).

**security administrator**

A programmer who manages, protects, and controls access to sensitive information.

**security category**

One of the two components of a security label that specifies which area of information a subject is permitted to access or an object is permitted to contain. See also [security level](#).

**security label**

A security level and any security categories associated with a subject or object. It is either a precise statement of the level of sensitivity and confidentiality of the contents of an object, or a precise statement of the power and privilege of a subject over objects.

**security labeling**

Assigning a label denoting security sensitivity to each subject and object in a system. The system uses security labels to enforce mandatory access control.

**security level**

One of the two components of a security label that specifies into which general class of sensitivity and confidentiality a subject or object falls. See also [security category](#).

**segment**

See [architected segment](#) or [saved segment](#).

**segment interface**

A method of reading and writing records in which the application is responsible for assembling input segments into logical records and for disassembling logical output records into segments. The segment interface is used under BSAM and QSAM.

**segment number**

The part of a virtual storage address needed to refer to a segment.

**segment space**

A special type of discontinuous saved segment that begins and ends on a megabyte boundary and contains 1-64 nonoverlapping member saved segments. A segment space is created by CP when member saved segments are defined. Access to a segment space provides access to all of its members.

**segment table**

A table used in DAT to control user access to virtual storage segments. Each entry indicates the length, location, and availability of a corresponding page table.

**select data file**

In VMSES/E, a file containing a list of the parts serviced by the VMFAPPLY command.

**selective line editing**

A feature of XEDIT that allows editing of a specified collection of lines while excluding other lines from the screen.

**selective preventive service**

The selective application of PTFs from an RSU. See also [preventive service](#).

**semaphore**

In CMS multitasking, a variable with an associated wait queue used by threads to control access to a shared resource. CMS multitasking blocks and unblocks threads on the semaphore's wait queue, but it does not control access to the resource. See also [mutex](#).

**separator**

A punctuation character that separates parts of a command or file, or that delimits character strings.

**sequence number**

In a source code file, a number in the last eight columns of each line. In an update file, the number identifies the lines in the source code file to update. See also [support identification code](#).

**serial line**

A network medium that is a de facto standard, not an international standard, commonly used for point-to-point TCP/IP connections. Generally, a serial line consists of an RS-232 connection into a modem and over a telephone line.

**serialization**

A method used to prevent the asynchronous altering of variables, whose validity must be checked, until after the operation for which they are validated is complete.

**server** (1) A functional unit that provides services to one or more clients over a network. Examples include a file server, a print server, and a mail server. See also client.

(2) The general name for a virtual machine that provides a service for a requesting virtual machine. See also communication server, CRR recovery server, file pool server machine, intermediate server, requester, and trusted server.

(3) A physical computer.

**Server Time Protocol (STP)**

A feature designed to provide the capability for multiple servers and Coupling Facilities to maintain time synchronization with each other, without requiring a Sysplex Timer.

**service**

(1) Changing a product after installation.

(2) Work performed by a server or service virtual machine.

**service level**

A number indicating the most recent RSU that has been applied.

**service machine**

A virtual machine running a program that provides system-wide services.

**service pool**

A set of service pool virtual machines, designed to do work with a conversation management routine (CMR).

**service pool virtual machine**

A virtual machine that is one of a set of identical virtual machines, created through the use of the POOL directory statement.

**service representative**

The user who is permitted to set or change the level of detail of I/O device error recording. This user type corresponds to IBM-defined CP privilege class F. This privilege class is reserved for IBM use only. See also privilege class.

**service routine**

A CP or CMS routine used for addressing and updating directories, formatting or initializing disks, or doing disk, tape, or terminal I/O functions.

**service tape**

See product service medium.

**service virtual machine**

A virtual machine that provides a system service, such as accounting, error recording, or monitoring.

**serviceable part**

In VMSES/E, an individual part of a product that can be serviced separately. A serviceable part has the file name of the source or replacement part and a file type in the form *ttnnnnnn*, where *ttt* is a unique three-character abbreviation for the part type and *nnnnn* is the PTF number.

Serviceable parts are maintained by both source updates and replacement service. See also base file type and usable form.

**service-level apply status table**

A software inventory table that indicates the apply status of all PTFs for a product on the system. The file type of this table is SRVAPPS.

**service-level build status table**

A software inventory table that indicates the status of all objects that have been built or need to be built for a product on the system. The file type of the table is SRVBLDS.

**service-level description table**

A software inventory table that contains the abstract information for an APAR that has been received on the system. The file type of this table is SRVDESCT.

**service-level inventory**

See [service-level software inventory](#).

**service-level receive status table**

A software inventory table that indicates the receive status of all PTFs for a product on the system. The file type of this table is SRVRECS.

**service-level software inventory**

A file maintained by VMSES/E that shows requisite relationships between PTFs, the status of the PTFs installed, the level of each part of the product serviced, and the status of objects built for the product. See also [system-level software inventory](#).

**session**

(1) See [terminal session](#).

(2) In CMS multitasking, a collection of processes sharing a common environment. The environment includes such things as traditional CMS assets and facilities, the virtual configuration of processors and I/O devices, and a primary address space.

(3) In OpenExtensions, a collection of process groups established for job control purposes.

(4) In network architecture, an association of facilities that establish, maintain, and release connections for communication between stations.

(5) In SNA, a logical connection established between two network addressable units (NAUs) that allows them to communicate. For routing purposes each session is identified by the local or network addresses of the session partners.

**session limit**

In SNA, the maximum number of sessions that can occur between a pair of LUs.

**session lock**

See [explicit session lock](#).

**sever** (1) To end communication with another virtual machine or with the user's own virtual machine.

(2) In CMS Pipelines, to cause a stream to become unconnected.

**SFS** See [shared file system](#).

**SFS communication adapter**

The part of CMS in a user's virtual machine that communicates with file pool server machines.

**SFS directory**

A directory in the CMS shared file system. SFS directories can be arranged to form a hierarchy in which one directory can contain one or more subdirectories as well as files. Also see [directory control directory](#) and [file control directory](#).

**sh\_history**

See [history file](#).

**share lock**

See [explicit share lock](#) and [implicit share lock](#).

**shared file system (SFS)**

A part of CMS that lets users organize their files into groups known as directories and selectively share those files and directories with other users.

**shared segment**

One or more segments of real storage, saved in a named saved system, member saved segment, or discontinuous saved segment, that can be shared among many virtual machines. For example, in a CMS named saved system, the CMS nucleus is shared in real storage by all virtual machines that loaded CMS by name; that is, every CMS virtual machine maps a 1 MB segment of virtual storage to the same 1 MB of real storage.

**shared system**

See named saved system.

**shell** In OpenExtensions, a command processor modeled after the UNIX System V shell with some of the features found in the KornShell. Through this command processor, the user can call shell commands or utilities that request services from z/VM, write shell scripts using the shell programming language, and run shell scripts and C/C++ language programs. See also kernel.

**shell procedure**

See shell script.

**shell program**

See shell.

**shell prompt**

The character string on the command line indicating that the system can accept a command.

**shell script**

A file containing shell commands. If the file is made executable, it can be executed by specifying its name as a simple command, which causes the shell to execute the commands within the script. Alternatively, a shell can be requested to execute the commands in a shell script by specifying the name of the shell script as the operand of the **sh** utility.

**shell variables**

Facilities of the shell program for assigning variable values to constant names.

**short** (1) In CMS Pipelines, to cause a stream to bypass a stage by connecting the output of the preceding stage to the input of the following stage. The shorted stage can neither read from the input stream nor write to the output stream.

(2) In CMS Pipelines, a bypass made by shorting.

**SID code**

See support identification code.

**side information**

System-defined values that are used for the initial values of the *partner\_LU\_name*, *mode\_name*, and *TP\_name* conversation characteristics, among others. z/VM implements side information in CMS communications directory files.

**signal** (1) A means of informing processes of asynchronous events. A mechanism by which a process may be notified of, or affected by, an event occurring in the system. Examples of such events include hardware exceptions and specific actions by processes. The term *signal* is also used to refer to the event itself.

(2) A method of interprocess communication that simulates software interrupts. See also interrupt.

**signal attention**

To press the attention key to present an attention interruption (I/O interruption) to CP or to the user's virtual machine.

**Simple Mail Transfer Protocol (SMTP)**

An Internet application protocol used to transfer mail between users on different systems. SMTP specifies how mail systems interact and the format of control messages they use to transfer mail.

**Simple Network Management Protocol (SNMP)**

A set of protocols for monitoring systems and devices in complex networks. Information about managed devices is defined and stored in a Management Information Base (MIB).

**simultaneous peripheral operations online (SPOOL)**

(1) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor.

(2) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

**Single Console Image Facility (SCIF)**

A CP function that enables a virtual machine operator to control multiple disconnected virtual machines from one physical terminal. See also primary user and secondary user.

**single key protection**

The association between each 4 KB block of real storage with one storage key. Only one key is needed to protect a 4 KB frame.

**single key storage**

Real storage composed of storage frames that are protected by one storage key per 4 KB frame rather than two storage keys per 4 KB frame.

**single precision**

(1) The specification that causes a floating-point value to be stored internally in the short format.

(2) The use of one computer word to represent a number, in accordance with the required precision.

**single system image cluster**

See z/VM single system image cluster.

**single user group**

In GCS, a single virtual machine that runs applications that do not require group communications. This allows an application to run without the overhead of group initialization and multiple virtual machines. Multiple users cannot IPL the same named saved system if it was built for a single user environment. See also virtual machine group.

**single-byte character set (SBCS)**

A character set in which each character is represented by a one-byte code. See also double-byte character set.

**single-configuration virtual machine definition**

A virtual machine definition that consists of a user entry and any included profile entry. This virtual machine definition allows only one virtual machine instance to be defined. In an SSI cluster, the user ID can be logged on to only one member of the cluster at a time.

**single-density DASD volume**

A DASD volume with the standard number of cylinders for its device type. See also double-density DASD volume and triple-density DASD volume.

**sink virtual machine**

See receiving virtual machine.

**sleep mode**

A mode in which the virtual machine is in a dormant state; that is, the virtual machine is not running, but connect time still accumulates, and messages can be displayed at the terminal. The virtual machine is restarted either at the end of a specified interval or when the user signals attention to CP.

- slot** (1) A contiguous area of a paging device in which a page can be stored.  
(2) In z/VM, a placeholder identified by a number in the list of CP-owned volumes.

**Small Computer System Interface (SCSI)**

An ANSI-standard electronic interface that allows personal computers to communicate with peripheral hardware, such as disk drives, tape drives, CD-ROM drives, printers, and scanners, faster and more flexibly than previous interfaces.

**small programming enhancement (SPE)**

Any new function being shipped in the service stream as an APAR, including rollback of new release line items to previous releases.

**SMAPI**

See systems management API.

**SMI** See Structure for Management Information.

**SMSG function**

A CP function that lets a virtual machine send a special message to another virtual machine programmed to accept and process the message.

**SMTP** See Simple Mail Transfer Protocol.

**SNA** See Systems Network Architecture.

**SNA/CCS terminal**

Any terminal accessing z/VM that is managed by a VTAM service machine.

**snap dump**

A dump, initiated by a command or macro, of the storage directly owned by CP. A snap dump is very similar to a hard abend dump but does not result in system termination.

**SNMP**

See Simple Network Management Protocol.

**SOA** See start of authority record.

- socket** (1) A method of communication between two processes. Sockets allow communication in two directions, in contrast to pipes, which allow communication in only one direction.  
(2) A means for directing data to an application in a TCP/IP network by using a unique identifier that is a combination of an IP address and a port number.  
(3) The system call that creates a socket and its associated data structure.

**socket address**

An address that results when the port identification number is combined with an Internet Protocol address.

**socket interface**

An application interface that allows users to write their own applications to supplement those supplied by TCP/IP.

**soft abend dump**

An automatic dump of a small amount of the storage directly owned by CP. The dump is created when CP encounters a problem for which system integrity is not jeopardized by the error, or when CP can isolate an error to a virtual machine. The dump does not result in a system termination.

| **Soft Fence**

| A DS8000<sup>®</sup> function exploited by GDPS 3.10 that protects the integrity of the old primary  
| volumes in the GDPS-managed consistency group during a HyperSwap scenario by putting them  
| into soft fence state and limiting most I/O operations directed to those volumes.

**soft requisite**

In VMSES/E, a PTF that is not required by another PTF, but affects any of the same modules. The relationship exists because the pre-built replacement parts that are shipped with PTFs are built with all prior PTFs.

**software inventory**

A set of files containing control and status information that is used during product installation and service. The software inventory has two levels: the system-level software inventory and the service-level software inventory.

**source code**

The input to a compiler or assembler, written in a source language. See also [object code](#).

**source directory**

The form of the user directory that consists of one or more CMS files containing directory statements coded according to the z/VM user directory language rules. The object directory is created from the source directory by the DIRECTXA command.

**source file**

A file of programming code that is not compiled into machine language.

**source product parameter file**

In VMSES/E, a file supplied with a product containing recommended values for the options that control VMSES/E processing for the product, formats of product installation and service media, and the list of build lists used to build the product. The file name is the ID of the product and the file type is \$PPF.

**source update**

A change to the source code for a product contained in a source update file. See also [auxiliary control file](#) and [control file](#).

**source update file**

A file containing a single change to a statement in a source file. The source update file can also include requisite information for applying the change.

**source virtual machine**

The virtual machine that initiates the sending of messages or data to another virtual machine (the receiving virtual machine).

**spanned record**

A logical record that spans across one or more data blocks on a storage device.

**sparse block**

A data block or pointer block that contains all sparse records. Sparse blocks are not physically stored in a file. See also [logically sparse file](#).

**sparse record**

For files with fixed-length records, a logical record with a position number between the position numbers of two records that have been written to the file. For example, if the last record written to the file, DOG DATA, has a position number of 55, you can write a record with a position number of 60. Records 56, 57, 58, and 59 are the sparse records. When a sparse record is read, a record of X'00's is returned. See also [logically sparse file](#) and [sparse block](#).

**SPE** See [small programming enhancement](#).

**special character**

A character other than a letter or number. For example, \*, +, and % are special characters.

**special message**

A data transmission, made up of instructions or commands, sent from one virtual machine to another by means of the SMSG function. A special message is processed by the receiving virtual machine and does not appear on the receiver's console.

**special variable**

A reserved variable name assigned a value during processing by the REXX/VM Interpreter, the EXEC 2 processor, or the CMS EXEC processor. These variables can be tested within an EXEC procedure, edit macro, or XEDIT macro.

**specialty processor**

A System z processor devoted to a specific workload. Examples are the IBM Integrated Facility for Linux (IFL), the IBM Internal Coupling Facility (ICF), the IBM System z Application Assist Processor (zAAP), and the IBM System z Integrated Information Processor (zIIP). See also [central processor](#).

**spin lock**

A lock for which a program waits in a loop, repeatedly checking until the lock becomes available.

**SPM** See [synchronization point manager](#).

**spoofing**

An act of forging and inserting data that is incorrect or not valid. It is most commonly used in reference to IP source spoofing, where the source address in an IP packet header is replaced with a false one, effectively masking the source of the packet (making it difficult to trace back to the originator).

**SPOOL**

See [simultaneous peripheral operations online](#).

**spool file**

A collection of data along with CCWs for processing on a unit record device. See also [system data file](#).

**spool file block**

A 4 KB buffer that contains control information, in addition to records.

**spool file buffer**

See [spool file block](#).

**spool file class**

A one-character class associated with each virtual unit record device and each spool file. For input spool files, the spool file class lets the user control which input spool files are read next. For output spool files, the spool file class lets the spooling operator better control or reorder the printing or punching of spool files having similar characteristics or priorities. The spool file class value can be A through Z, 0 through 9, or \*.

**spool file recovery**

To recover from disk the checkpoint data that was previously recorded for closed spool files. If a warm start is not possible, the z/VM system operator can attempt a checkpoint start or a force start. See also [checkpoint start](#) and [force start](#).

**spool file system**

A CP file system that manages spool files in a temporary storage area.

**spool file tag**

In z/VM, a 136-character data field that can be associated with each output spool file generated. The use, content, and format of this field is a bilateral decision between the originator and receiver of the file.

**spool ID**

The identification number automatically assigned by CP to a spool file when the file is closed. The spool ID number can be 0001 - 9999; it is unique for each spool file for each user. To identify a given spool file, a user must specify the owner's user ID, the virtual device type, and the spool ID.

**spooling**

The processing of files created by or intended for readers, punches, and printers. The spool files can be sent from one virtual device to another, from one virtual machine to another, and to real devices. See also [virtual console spooling](#).

**spooling device**

An I/O device (reader, punch, or printer) that reads input and writes output.

**spooling operator**

The user who controls spool files and the z/VM system's real unit record devices allocated to spooling use. This user type corresponds to IBM-defined CP privilege class D. See also [privilege class](#).

**SQL** See [Structured Query Language](#).

**SR** See [symptom record](#).

**SSCP** See [system services control point](#).

**SSI cluster**

See [z/VM single system image cluster](#).

**SSI cluster member**

A z/VM system participating in an SSI cluster.

**SSI-enabled source directory**

A source directory that can contain single-configuration virtual machine definitions and multiconfiguration virtual machine definitions and that has the **SSI** option specified on the DIRECTORY statement. In each multiconfiguration virtual machine definition, the identity entry can include multiple BUILD statements, each of which must specify the system name of a member of the SSI cluster. Virtual machine instances can be created on multiple members from a multiconfiguration virtual machine definition when the object directory is created from this type of source directory.

**SSI-ready source directory**

A source directory that can contain single-configuration virtual machine definitions and multiconfiguration virtual machine definitions, but does not have the **SSI** option on the DIRECTORY statement. In each multiconfiguration virtual machine definition, the identity entry can include at most one BUILD statement, on which the system name field must be an asterisk (\*). Only one virtual machine instance can be created from a multiconfiguration virtual machine definition when the object directory is created from this type of source directory.

**SSL** See [Secure Sockets Layer](#).

**S-STAT**

A block of storage that contains the file status tables (FSTs) associated with the S-disk. The FSTs are sorted so that a binary search can search for files. The S-STAT usually resides in the CMS nucleus so the S-STAT can be shared. Only files with a file mode of 2 have their associated FSTs in the S-STAT. See also [Y-STAT](#).

**stable access**

A minidisk access mode that means the user holding the stable access can be assured that the minidisk remains stable, unchanged by others, until the access is released. See also [exclusive access](#).

**stack** A list of items constructed and maintained so that the last item stored is the first item retrieved (LIFO). See also [console stack](#) and [program stack](#).

**stage** In CMS Pipelines, a processing step in a pipeline. A pipeline consists of one or more stages, delimited by stage separators.

**stage separator**

In CMS Pipelines, a character that indicates the end of one stage of a pipeline and the beginning of the next stage. The default stage separator is the solid vertical bar (|).

**stall** In CMS Pipelines, a condition in a multistream pipeline in which no stage can run.

**stand-alone dump**

A dump acquired without regular system functions. For example, to obtain a CP dump when the regular system is unable to dump the machine, the stand-alone dump utility can be used to create a CP stand-alone dump.

**stand-alone program loader (SAPL)**

A program installed in cylinder 0 of a disk volume. The stand-alone loader is a generic loader that can load any program designed to work with it.

**standard error (stderr)**

(1) The place where many programs place error messages, which is the display screen unless another place is specified with redirection.

(2) An output stream usually intended to be used for diagnostic messages.

(3) The conventional name for file descriptor 2. By convention, programs write diagnostics and error messages to this descriptor. Usually, the descriptor refers to the display screen, but it may be changed by redirection. This descriptor is separate from standard output so that error diagnostics are still visible when the output is redirected.

**standard input (stdin)**

(1) The primary source of data going into a command. Standard input comes from the keyboard unless redirection or piping is used, in which case standard input can be from a file or the output from another command.

(2) An input stream usually intended to be used for primary data input.

(3) The conventional name for file descriptor 0. By convention, programs read input from this descriptor. Usually, the descriptor refers to the keyboard, but it may be changed by redirection.

**standard output (stdout)**

(1) The primary destination of data coming from a command. Standard output goes to the display unless redirection or piping is used, in which case standard output can be to a file or another command.

(2) An output stream usually intended to be used for primary data output.

(3) The conventional name for file descriptor 1. By convention, programs write output to this descriptor. Usually, the descriptor refers to the display screen, but may be changed by redirection.

**standard processor**

See central processor.

**start of authority (SOA) record**

In the Domain Name System, the resource record that defines a zone.

**state transition**

The change from one program state to another in a conversation.

**status area**

The area in the bottom right corner of the display screen that contains messages about the status of a virtual machine.

**stderr** See standard error.

**stdin** See standard input.

**stdout** See standard output.

**sticky bit**

A file access permission bit that allows multiple users to share a single copy of an executable file.

**storage access component (SAC)**

The part of a file pool server machine that accesses catalogs and user files and provides support for locking, catalog indexes, logging, file pool recovery, and file pool generation. See also [data access component](#).

**storage area network**

A network dedicated solely to disk, tape, and optical storage, which is based on a fabric of fibre, switches, and hubs that connects storage devices to a heterogeneous set of servers on a many-to-many basis.

**storage control**

The component in a DASD subsystem that connects the DASD to the host channels. It does channel commands and controls the DASD devices.

**storage group**

A subset of minidisks within a file pool. Each storage group is identified by a number.

**storage group 1**

See [catalog storage group](#).

**storage key**

An indicator associated with one or more storage blocks, which requires that tasks have a matching protection key to use the blocks.

**STP** See [Server Time Protocol](#).**stream**

(1) In CMS Pipelines, a flow of data records into or out of a stage. One end of a stream enters or leaves a stage. The other end can be connected to streams of other stages so that data can flow between stages. A stage's input and output streams are numbered separately, starting at zero. Streams can also be given an identifier.

(2) In REXX, characters transmitted to a destination or from a source, such as a human user, a display, or a file. A stream can be manipulated character by character or line by line. A stream is transient (for example, data sent to or received from a display) or persistent (for example, data written to or read from a file).

(3) In TCP/IP, the collection of datagrams needed to represent a message transmitted by a process. The number and size of the datagrams needed to contain the message may vary according to network capacity. The process sees only the message, not the datagrams.

(4) In OpenExtensions, sequential input or output from an open file descriptor.

(5) See [data stream](#).

**stream editor**

In OpenExtensions, an editor called by the **sed** command, which modifies lines from a specified file, according to an edit script, and writes them to a standard output.

**stream identifier**

In CMS Pipelines, a symbolic name for a stream, consisting of up to four alphabetic characters or a combination of alphabetic characters and digits that includes at least one alphabetic character, with no intervening blanks. Built-in stages, subcommands, and pipeline macros can refer to a stream by number, where 0 is the primary stream, 1 is the secondary stream, and so on.

**string** (1) A linear sequence of entities such as characters or physical elements. Examples of strings are alphabetic string, binary element string, bit string, character string, search string, and symbol string.

(2) In VMSES/E, a group of minidisks defined for a specific function in the product parameter file; for example, the BASE2 string, which holds source code.

**structurally sparse file**

A file that contains sparse blocks.

**Structure for Management Information (SMI)**

The rules used to define the objects that can be accessed through a network management protocol. See also Management Information Base.

**Structured Query Language (SQL)**

A fourth generation English-like programming language used to perform queries on relational databases.

**subchannel**

A logical function of a channel subsystem associated with the management of a single device.

**subchannel measurement block (SCMBK)**

A control block that can be associated with a given device. The control block contains measurement data for the device such as the I/O rate and the timing values of service time for various components. The hardware has responsibility to update this information. From the measurement block information, performance products can compute device service time, I/O rate, and utilization.

**subchannel number**

A system-unique 16-bit value used to address a subchannel. See also I/O address.

**subcommand**

A command that can be specified only within the environment created by a certain command. For example, the XEDIT subcommands can be specified only within the environment that is initiated by specifying the XEDIT command.

**subconfiguration entry**

A source directory entry that begins with the SUBCONFIG statement. This entry contains a set of directory statements in a multiconfiguration virtual machine definition that are specific to one of its virtual machine instances.

**subdirectory**

A directory contained within another directory in a file system hierarchy.

**subject**

In computer security, an active entity that causes information to flow among objects or changes the system's state. A subject is generally a person, process, or device. See also object.

**subnet**

A networking scheme that divides a single logical network into smaller physical networks to simplify routing.

**subnet address**

The portion of the host address that identifies a subnet.

**subnet mask**

A mask used in the IP protocol layer to separate the subnet address from the host portion of the address.

**subnetwork**

See subnet.

**subroutine pipeline**

In CMS Pipelines, the operand of the CALLPIPE pipeline subcommand.

**subshell**

An instance of the shell program started from an existing shell program.

**suffix** A character string attached to the end of a file name that helps identify the type of file.

**supersede**

For a given PTF, to include all the APARs, parts, and requisite relationships of another PTF.

**superuser**

In OpenExtensions, a system user who has the special rights and privileges needed to manage OpenExtensions processes and files.

**supervisor call instruction (SVC)**

An instruction that interrupts a program being executed and passes control to the supervisor so that it can do a specific service indicated by the instruction.

**supervisor state**

A state during which the processor can execute I/O and other privileged instructions. See also problem state.

**support identification (SID) code**

In an update file, a character string in the first nine columns of the last 17 columns of each line that can be used to identify who made the change and why. See also sequence number.

**SVC** See supervisor call instruction.

**SVC 76**

A supervisor call interruption code used for virtual device errors. z/VM provides a means to record the error incidents met by certain operating systems running in a virtual machine.

**SVM** See service virtual machine.

**switched point-to-point channel path configuration**

In an ESCON I/O interface, a configuration that consists of a link between a channel and an ESCON Director (ESCD) and one or more links from the ESCD, each of which attaches to a control unit. This configuration depends on the capabilities of the ESCD for establishing and removing connections between channels and control units. See also point-to-point channel path configuration.

**symbolic destination name**

A name that an APPC/VM or CPI-Communications connection uses. Symbolic destination names index SNA routing and security parameters, which are stored in communications directory NAMES files, to complete a connection.

**symbolic link**

A type of BFS file that contains the path name of, and acts as a pointer to, another BFS file or directory. The path name in the symbolic link can be relative or absolute. If a symbolic link contains a relative path name, the path name is relative to the directory containing the symbolic link.

**symptom record**

A collection of data conveying basic information about a z/VM software problem. The Dump Viewing Facility or VM Dump Tool can display this data.

**sync point**

See synchronization point processing.

**sync point manager**

See synchronization point manager.

**sync point processing**

See synchronization point processing.

**sync point tree**

A conceptual structure used to organize all the protected resources and protected conversations that are coordinated by a synchronous point manager in a transaction. The tree consists of nodes that contain the protected resources and nodes that are the targets of protected conversations.

**synchronization point manager (SPM)**

The part of the Coordinated Resource Recovery (CRR) facility that resides in the application's virtual machine. A protected resource registers with the SPM to participate in Coordinated Resource Recovery (CRR). The SPM coordinates the updating of protected resources and distributes the coordination of protected conversations to other SPMs.

**synchronization point processing**

The coordination function of CRR. The synchronization point manager (SPM) drives the participating resource adapters through the following SPM exits:

- Precoordination - checks participating resources to ensure they are ready for a sync point.
- Coordination - the actual sync point, which implements the one-phase and two-phase commit protocols.
- Postcoordination - performs cleanup processing after a sync point.

**synchronous**

(1) Pertaining to two or more processes that depend upon the occurrences of specific events, such as a common timing signal. See also asynchronous.

(2) Occurring with a regular or predictable time relationship.

**synchronous communication**

A method of communication in a conversation by which a user's virtual machine is put into a WAIT state and cannot issue any communication functions on any conversations until a particular function completes. See also asynchronous communication.

**Synchronous Data Link Control (SDLC)**

A protocol for managing synchronous information transfer over a data link connection.

**synonym**

(1) In CMS, an alternative command name defined by the user as equivalent to an existing CMS command name.

(2) In XEDIT, an alternative name defined by the user for an XEDIT subcommand.

**SYSGEN**

See system generation.

**SYSHIGH**

The security label associated with the most important and most sensitive data handled by a given trusted computing base. It is a security label combining the system's highest security level with all of the system's security categories. See also SYSLOW.

**SYSLOW**

The security label associated with the least important and least sensitive data handled by a given trusted computing base. It is a security label consisting of the system's lowest security level with none of the system's security categories. See also SYSHIGH.

**sysplex**

A set of z/OS systems that communicate with each other through certain multisystem hardware components and software services.

**SYSRES volume**

See CP system residence volume.

**system administrator**

The person responsible for maintaining a computer system.

**system analyst**

The user who examines and saves z/VM system operation data in specified z/VM storage areas. This user type corresponds to IBM-defined CP privilege class E. See also privilege class.

**SYSTEM CONFIG**

See system configuration file.

**system configuration**

- (1) The devices and programs that form a particular data processing system.
- (2) A process that specifies the devices and programs that form a particular data processing system.

**system configuration file**

A file that resides on the parm disk and contains the primary system definition required at system IPL. The file name is usually SYSTEM CONFIG.

**system console**

The console used by a production system.

**system control program (SCP)**

Programming that is fundamental to the operation of the system. SCPs include the BCP element of the z/OS operating system and the CP component of the z/VM operating system, plus any other programming that is used to operate and maintain the system.

**system data file (SDF)**

A file defined by CP to hold a collection of data associated with a particular system function, such as a saved segment, a saved system, a printer image library, user class restructure definitions (in z/VM releases prior to V6.3), a message repository, or system trace data. Because a system data file contains no CCWs, it cannot be processed on a unit record device. See also [spool file](#).

**system DDR**

See [z/VM system DDR](#).

**System Delivery Offering**

See [z/VM System Delivery Offering](#).

**system disk**

- (1) In VMSES/E, a minidisk or file pool directory containing the other products that are required during service.
- (2) In CMS, the minidisk (usually at virtual address 190) that holds CMS executables.

**system event**

In CMS multitasking, an event that can be defined, monitored, and signaled by CMS or an application.

**system execution area**

A part of host logical storage that contains the CP nucleus, the prefix pages, dynamically allocated free storage and other CP-use pages, aliases (references to guest pages), and routines loaded with the CPXLOAD system configuration statement or the CP CPXLOAD command. Host logical storage addresses in the CP nucleus and the prefix pages are identity mapped to real storage (that is, the host logical storage addresses and host real storage addresses are identical). The rest of the system execution area is dynamic (not identity mapped).

**system gateway**

A gateway that APPC programs can use to access global or private resources on a specific system within a CS or TSAF collection, or that APPC programs in a TSAF collection can use to access resources in an adjacent CS collection, or that APPC programs in a CS collection can use to access resources in an adjacent TSAF collection. See also [global gateway](#) and [private gateway](#).

**system generation (SYSGEN)**

The process of creating a particular system tailored to the requirements of a data processing installation.

**system identifier**

A unique name that identifies a particular z/VM system running in an LPAR, on hardware with a particular serial number, or in a virtual machine.

**system integrity**

The property of a system that is designed, implemented, and maintained to protect itself from unauthorized access.

**system load**

The combination of active devices, programs, and users that use the system resources of the processor and storage.

**system name**

See system identifier.

**system name table**

In CP, the table that contains the name and location of saved systems, including shared and nonshared segments.

**system object**

An object, such as a saved segment, that may contain objects or parts supplied by more than one component or product.

**system operator**

The user who controls the z/VM system and is responsible for the availability of the z/VM system and its resources. In addition, a system operator controls system accounting, broadcast messages, virtual machine performance options, and other options that affect the overall performance of z/VM. This user type corresponds to IBM-defined CP privilege class A. See also privilege class, primary system operator and virtual machine operator.

**system operator console**

See primary system operator console.

**system profile**

An exec (SYSPROF EXEC) that resides in a saved system or on a system disk and is called by CMS initialization. The exec contains some initialization functions, and provides a means for installations to override the default CMS environment by tailoring the exec to suit the installation.

**system programmer**

The user who updates or changes system-wide parameters of the z/VM system. This user type corresponds to IBM-defined CP privilege class C. See also privilege class.

**system reset**

In z/VM, to perform a system-reset-normal function on a virtual machine and to place all processors in the virtual machine configuration in a hard stop.

**system residence volume**

See CP system residence volume.

**system resource**

A resource that is unique on the system on which it is defined (similar to a local resource) but is accessible (similar to a global resource) from remote systems in a CS or TSAF collection, or in the SNA network. See also global resource, local resource, and private resource.

**system resource manager**

An application that runs in a server workstation or virtual machine and provides a service for connecting programs. The system resource identified is not known globally throughout the TSAF or CS collection. However, by specifying the system gateway name of the system on which the system resource manager resides, APPC programs within the collection can access the system resource. See also global resource manager, local resource manager, and private resource manager.

**system resource operator**

The user who controls all the real resources of the z/VM system, except those controlled by system operators and spooling operators. This user type corresponds to IBM-defined CP privilege class B. See also privilege class.

**system restart**

In z/VM, to simulate a hardware system restart.

**system segment identification file**

A file (SYSTEM SEGID) that identifies the logical saved segments on the system and the physical saved segments in which they reside.

**system services control point (SSCP)**

The focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network.

**system startup**

See initial program load.

**system string**

In VMSES/E, the set of system disks.

**system trace file**

A type of system data file that contains CP or virtual machine trace data.

**system-level apply status table**

A software inventory table that indicates the apply status for all products on the system. The file type of this table is SYSAPPS.

**system-level build status table**

A software inventory table that indicates the build status for all products on the system. The file type of this table is SYSBLDS.

**system-level description table**

A software inventory table that contains the descriptions of the products that have been received on the system. The file type of this table is SYSDESCCT.

**system-level inventory**

See system-level software inventory.

**system-level receive status table**

A software inventory table that indicates the receive status for all products on the system. The file type of this table is SYSRECS.

**system-level software inventory**

A file maintained by VMSES/E that contains requisite relationships between products or components, a mapping of the product identifier to the name of the \$PPF file used during installation, the status of the product or component on the system, and a mapping of the PTF file type abbreviation to the real CMS file type. See also service-level software inventory.

**Systems Application Architecture (SAA)**

A defined set of interfaces, conventions, and protocols that can be used by various IBM systems.

**systems management API**

An API that provides a standard, platform-independent client interface that reduces the amount of VM-specific programming skills required to manage resources for virtual systems (guests).

**Systems Network Architecture (SNA)**

The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and for controlling their configuration and operation.

**T**

**tag** (1) A mechanism used to identify certain attributes having some bearing on handling of character data.

(2) To associate descriptive information with system spool files or spooled devices.

**tailorable file**

Any source-level product file that requires user input in order for the product to work correctly. Examples of tailorable files are the SYSTEM CONFIG file and a PROFILE EXEC for a user.

**tailorings**

Changes made to a source-level product file to customize it for a customer environment.

**tape descriptor file**

A file containing a directory of the products on a product service tape.

**tape document**

A document describing the service procedure for a product service tape.

**target** (1) In XEDIT, one of many ways to locate a line. A target can be specified as an absolute line number, a relative displacement from the current line, a line name, or a string expression.

(2) In CMS Pipelines, a way to locate a record. A target can be a group of characters, a delimited string, a number of records, or a displacement value.

**target disk**

In VMSES/E, a minidisk or file pool directory that receives files from which the objects are built.

**target processor**

The processor that controls execution during a program restart, instruction trace, stand-alone dump, or IPL.

**target string**

(1) In VMSES/E, the set of target disks.

(2) A character string that is the object of a search or locate function.

**task control block (TCB)**

A control block that is used to communicate information about tasks within an address space that is connected to a subsystem.

**task ID**

A 2-byte field that uniquely defines a task within a GCS virtual machine. Task ID is sometimes combined with machine ID to uniquely identify a task within a virtual machine group.

**Tb (terabit)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 40, or 1,099,511,627,776, bits.

(2) For disk storage capacity and communications volume, 10 to the power of 12, or 1,000,000,000,000, bits.

**TB (terabyte)**

(1) For processor storage, real and virtual storage, and channel volume, 2 to the power of 40, or 1,099,511,627,776, bytes.

(2) For disk storage capacity and communications volume, 10 to the power of 12, or 1,000,000,000,000, bytes.

**TCB** (1) See task control block.

(2) See transmission control block.

(3) See trusted computing base.

**TCP** See Transmission Control Protocol.

**TCP/IP**

See Transmission Control Protocol/Internet Protocol.

**TCP/IP for z/VM**

A facility supplied with z/VM that provides support for the TCP/IP protocol suite.

**T-disk** See [temporary minidisk](#).

**Telnet** A TCP/IP application protocol for remote terminal connection service that allows a user at one site to gain access to a foreign host as if the user's terminal were connected directly to that foreign host.

**template file**

A file that defines the number, data type, and length of parameters that a callable services library routine requires.

**temporary minidisk**

An area on a DASD available to the user for newly created or stored files until logoff, at which time the area is released. Temporary disk space is allocated to the user during logon (through the MDISK directory statement) or by using the CP DEFINE command. A temporary disk is nonshareable (private).

**temporary product parameter file**

In VMSES/E, the output of the VMFOVER command, which contains overrides for a source product parameter file. The file name is either the file name of the last override product parameter file in the chain of overrides, or the file name of the source product parameter file. The file type is \$PPFTEMP.

**terabit** See [Tb](#).

**terabyte**

See [TB](#).

**terminal**

A device, usually equipped with a keyboard and a display, capable of sending and receiving information.

**terminal emulator**

A program that simulates the function of a particular kind of terminal.

**terminal input buffer**

A buffer used to hold lines entered at the user's terminal until CMS processes them. See also [console stack](#).

**terminal session**

The time from logon to logoff when a user and the virtual machine can use the facilities of z/VM. This also includes any time that the virtual machine is running in disconnected mode.

**tertiary input stream**

In CMS Pipelines, an input stream defined for a stage that has a label definition by the second reference to the label, or an input stream defined by the ADDSTREAM pipeline subcommand. The tertiary input stream has input stream number 2.

**tertiary output stream**

In CMS Pipelines, an output stream defined for a stage that has a label definition by the second reference to the label, or an input stream defined by the ADDSTREAM pipeline subcommand. The tertiary output stream has output stream number 2.

**text deck**

See [text file](#).

**text deck shell**

A file that contains requisite information for a text deck but no additional code.

**text file**

(1) A file with a filetype of TEXT that contains a relocatable object module created after assembling or compiling a program.

(2) A file that contains only printable characters.

**text library**

A CMS file that contains text files and a directory that indicates their location within the library.

**third-level storage**

The virtual storage created and controlled by a virtual machine. See also [first-level storage](#) and [second-level storage](#).

**thread** In CMS multitasking, the basic dispatchable entity in the system. A thread is an instance of execution of a unit of program code. The thread environment is characterized by a PSW, a set of register values, and a save area stack.

**throttle**

To limit or control the number of I/O operations that a guest operating system can initiate to a specific real device, thus preventing the guest from interfering with or dominating the I/O resources of that real device.

**time bomb**

See [virtual machine time bomb](#).

**time sharing**

A function of a computing system that lets many users execute programs concurrently and interact with the programs during execution.

**time stamp**

A value of an object that is an indication of the system time at some critical point in the history of the object.

**time-of-day (TOD) clock**

A hardware feature required by z/VM. The TOD clock is incremented once every microsecond, and provides a consistent measure of elapsed time suitable for the indication of date and time. The TOD clock runs regardless of the processor state (running, wait, or stopped).

**timeout**

- (1) A time interval that is allotted for an event to occur or complete before operation is interrupted.
- (2) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event.

**TLS** See [Transport Layer Security](#).

**TN3270**

An standard protocol for transmitting 3270 data streams over Telnet.

**TOD clock**

See [time-of-day clock](#).

- token**
- (1) A value passed as a parameter for the purpose of uniquely identifying objects.
  - (2) A particular message or bit pattern that signifies permission or temporary control to transmit over a network.
  - (3) An eight-character symbol created by the CMS EXEC processor when it scans an EXEC procedure or EDIT macro statements. Symbols longer than eight characters are truncated to eight characters.
  - (4) In REXX, a unit of low-level syntax from which clauses are built. Examples are literal strings, hexadecimal strings, binary strings, symbols, numbers, operator characters, and special characters.

**tokenized PLIST (parameter list)**

A string of doubleword aligned parameters occupying successive doublewords. See also [untokenized parameter list](#).

**top directory**

The directory created for a user when the user is enrolled in a file pool. The name of the top directory is the same as the person's user ID.

**top of file**

When editing a file with XEDIT, the line above the file contents area.

**topmost window**

In full-screen CMS, the highest window in the display order such that the window name is not WM or STATUS and the window currently displays at least one virtual screen data line or reserved line. See also windowing.

**total CPU time**

The processor time used by a virtual machine, which includes virtual processor time plus the CP overhead used to service the virtual machine.

**TPN** See transaction program name.

- trace**
- (1) A record that exhibits the sequence of instructions executed by a computer program.
  - (2) The process of recording the sequence in which the instructions in a program are executed and, optionally, the values of the program variables used in the instructions.

**trace table**

See CP trace table.

**transaction**

- (1) A unit of processing consisting of one or more application programs, affecting one or more objects, that is initiated by a single request.
- (2) See also logical unit of work and logical unit of work identifier.

**transaction program**

- (1) An application that runs transactions within a particular LU in an SNA network.
- (2) In CRR, an application that executes one or more transactions or CRR logical units of work.

**transaction program name (TPN)**

A symbolic name given to a particular transaction program in an SNA-defined network.

**transient program area**

In CMS, the virtual storage area occupying locations X'E000' - X'10000'. Some CMS commands and user programs execute in this area of CMS storage.

**transition**

See state transition.

**translate table**

See translation table.

**translation mode**

An addressing mode that controls the handling of addresses through dynamic address translation (DAT). Examples are access-register mode and primary-space mode.

**translation table**

- (1) A table used to replace one or more characters with alternative characters—for example, to convert characters representing an event to those representing a procedure call, characters of a code set to those of another code set, or characters representing a relocated address to those representing an absolute address.
- (2) A table that maps virtual addresses with real addresses.
- (3) A table that specifies the mapping of events or event sequences to procedure names.

**transmission control block (TCB)**

An internal control block within the TCP/IP address space.

**Transmission Control Protocol (TCP)**

In the Internet suite of protocols, a protocol that provides reliable, process-to-process data stream delivery between nodes in interconnected computer networks. TCP assumes that Internet Protocol (IP) is the underlying protocol.

**Transmission Control Protocol/Internet Protocol (TCP/IP)**

A suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network.

**transparency mode**

A BSC mode that permits transmission of any data, bypassing regular BSC control character scanning.

**transparent mode**

See [TN3270](#).

**Transparent Services Access Facility (TSAF)**

A component of z/VM that runs in a virtual machine and routes communication between APPC application programs on interconnected z/VM systems. When a TSAF virtual machine on one z/VM system establishes a link to a TSAF virtual machine on another z/VM system, a TSAF collection is formed.

**transport layer**

Layer 4 of the Open Systems Interconnection (OSI) model that defines protocols governing message structure and some error checking.

**Transport Layer Security (TLS)**

A cryptographic protocol that provides communication security over the Internet. TLS is the successor to the Secure Sockets Layer (SSL) protocol, and offers additional cipher suites.

**trap** (1) A function that intercepts exception conditions.

(2) In REXX, to recognize that a currently enabled condition occurred and to perform the CALL or SIGNAL instruction specified when the condition trap was enabled.

(3) An unsolicited message that is sent by an SNMP agent to an SNMP network management station to report an exception condition.

**triple-density DASD volume**

A DASD volume with three times the standard number of cylinders for its device type. See also [single-density DASD volume](#) and [double-density DASD volume](#).

**truncation**

The shortening of a field, value, statement, or string. See also [minimum truncation](#).

**truncation setting**

In XEDIT, the value that determines the maximum length of input lines.

**trusted computing base (TCB)**

In computer security, all of the protection mechanisms within a computer system, including hardware, software, and firmware, the combination of which enforces a security policy. It creates a basic protection environment and provides additional user services required for a trusted computer system.

**trusted server**

(1) A server that presents a certificate from a valid certification authority.

(2) In z/VM, a virtual machine that runs programs necessary for the operation of the system. These programs provide services such as accounting, error recording, security, network management, printing, and many others. A trusted server always runs disconnected, and it never performs work on behalf of some other user. Though not all trusted servers are privileged, most are able to issue CP privileged commands. A RACF virtual machine is an example of a trusted server.

**TSAF** See Transparent Services Access Facility.

**TSAF collection**

A group of up to eight z/VM systems that are interconnected through a TSAF virtual machine running in each system.

**TSAF virtual machine**

A virtual machine in which TSAF is installed and running.

**TSAF VTAM line driver**

A portion of TSAF that handles APPC communications across VTAM-controlled links.

**two-phase commit protocol**

A two-step process by which recoverable resources and an external subsystem are committed. During the first step, the database manager subsystems are polled to ensure that they are ready to commit. If all subsystems respond positively, the database manager instructs them to commit. On z/VM, this protocol is implemented by SFS and CRR. See also one-phase commit protocol.

**two-word command**

A command that resolves to a one-word module or program name by using the first two tokens of its tokenized parameter list.

**U**

**UCR** See user class restructure.

**UCS** See universal character set.

**UCW** See unit control word.

**UDP** See User Datagram Protocol.

**UID** See POSIX user ID.

**uncommitted work**

Operations associated with a work unit that have not yet been committed or rolled back.

**unformatted file**

A file displayed with data that is not arranged with particular characteristics, such as line spacing, headings, and number of characters and lines per page. See also formatted file.

**Unified Resource Manager**

See IBM zEnterprise Unified Resource Manager.

**unit address**

The last two hexadecimal digits of an I/O address or a device number.

**unit control word (UCW)**

An identifier that contains the control information necessary for a channel to perform input/output operations to an attached device.

**unit record device**

A reader, line printer, or punch. z/VM provides virtual unit record devices for each virtual machine.

**universal character set (UCS)**

A printer feature that permits a variety of character arrays.

**universal class card reader**

A virtual card reader that can read any class of reader, printer, or punch files spooled or transferred to it.

**UNIX** A highly portable operating system originally developed by Bell Laboratories that features multiprogramming in a multiuser environment. UNIX is implemented in the C language. UNIX was originally developed for use on minicomputers but has been adapted on mainframes and

microcomputers. It is especially suitable for multiprocessor, graphics, and vector-processing systems. Many of the commands in the OpenExtensions shell are based on similar commands available with UNIX System V.

**unmount**

To logically disassociate a mountable file system from another file system.

**unnamed pipe**

In OpenExtensions, a pipe that typically sends data from one process to another; the two ends of a pipe can be used in a single program task. A pipe does not have a name in the byte file system, and the pipe vanishes when the last process using it closes it. The receiving process reads the data first-in-first-out (FIFO). See also FIFO special file.

**unparked**

The state of a logical processor that is in use. See also parked.

**unresolved alias**

An alias pointing to a base file that does not exist or for which the creator does not have the necessary authorizations. Such aliases are resolved when the base file is created, provided the authorizations exist. Administrator authority is required to create an unresolved alias.

**untokenized parameter list**

A parameter list in which no restrictions are placed on the structure of the items in the list. See also token and tokenized PLIST.

**unused ALE**

The state of a host access-list entry that exists when the entry does not designate an address space. If a virtual machine attempts to use a host access-list entry that is in the unused state, an ALEN-translation exception is recognized.

**update file**

See source update file.

**update lock**

See explicit update lock.

**update service**

Servicing a part by applying a change to a source file statement, then assembling or compiling the source file to produce a new object file.

**update shell**

A file that contains requisite information for applying an update but does not contain the update code.

**upgrade installation**

A technique for upgrading a z/VM V6.2 or later system to a newer release. An upgrade installation is performed using a two-stage approach, with two separate sets of changes being defined and then made on the current system. The first set of changes, STAGE1, can be made to the current system without disrupting normal system operation. After the STAGE1 changes are completed, the current system is backed up and all normal production work is stopped. Then the more disruptive STAGE2 changes can be done. In a z/VM SSI cluster environment, production Linux workloads can be relocated from the current system to other members of the cluster before the STAGE2 activities are performed.

**UPSI** See user program switch indicator.

**usable form**

A built part of a product. The service level of a usable form cannot be determined from the file identifier; for instance, an exec file with the file type of EXEC. See also base file type and serviceable part.

**usable form product parameter file**

In VMSES/E, a product parameter file produced by applying all override product parameter files

to a source product parameter file, and used by most VMSES/E execs during installation. The file name is either the file name of the last override product parameter file in the chain of overrides, or the file name of the source product parameter file if there are no overrides. The file type is PPF.

- user** (1) Anyone who requests the services of a computing system.  
(2) In z/VM, a virtual machine.

**user address space**

An address space that has at least one shell task known to the kernel address space. This address space can contain a shell or an application program that uses OpenExtensions.

**user class**

See privilege class.

**user class restructure (UCR)**

In z/VM releases prior to V6.3, the use of class override files to extend the CP privilege class structure.

**user data**

In a file pool, any data that resides in storage groups 2 - 32767.

**User Datagram Protocol (UDP)**

An Internet protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process.

**user definition**

See virtual machine definition.

**USER DIRECT**

See user directory.

**user directory**

A collection of specifications, called virtual machine definitions, that define the names (user IDs), configurations, and operating characteristics of virtual machines. The user directory has two forms: a source form and an object form.

**user entry**

A source directory entry that begins with the USER statement. This entry, plus any included profile entry, contains the set of directory statements for a single-configuration virtual machine definition.

**user exit**

A documented z/VM programming interface that can be used by an application program to transfer control to a user exit routine. Generally, a user exit affects only the particular application specifying the user exit and is run as part of the application program. See also installation-wide exit.

**user ID**

- (1) A string of 1 - 8 characters that uniquely identifies a virtual machine to the z/VM system and to an external security manager (ESM).  
(2) See also POSIX user ID.

**user identification card**

See ID card.

**user memo**

- (1) At the system level, special instructions for installing a product.  
(2) At the service level, special instructions for installing a PTF.  
(3) See also memo-to-users.

**user modification**

See local modification.

**user name**

(1) See POSIX user ID.

(2) In RACF, 1 - 20 alphanumeric characters that represent a RACF-defined user.

(3) See user ID.

**user profile**

(1) In OpenExtensions, a description of a user, including user ID, user name, default group name, password, owner, access authority, and other attributes obtained at logon.

(2) In OpenExtensions, a file in the user's home directory named `.profile` that contains shell commands that set initial user-defined characteristics and defaults for the session.

**user program**

A transaction program that requests a service from a resource manager program. User programs reside in requester virtual machines.

**user program switch indicator (UPSI)**

Switches that can be set by the CMS SET command. The user can set the switches (1 byte) to a desired value, which can be tested by a program in CMS/DOS.

**user-written program**

A module, exec, or XEDIT macro that is written by a user.

**user-written stage**

In CMS Pipelines, a stage that is not supplied by CMS Pipelines. A user-written stage is written in REXX or Assembler language. A REXX user-written stage can contain pipeline subcommands, REXX instructions, and host commands. An Assembler user-written stage can contain pipeline subcommands and pipeline Assembler macros. Assembler user-written stages provide increased performance over REXX user-written stages, especially when used to process large amounts of data. See also built-in stage.

**utility** (1) A computer program in general support of computer processes. Examples of a utility are a diagnostic program, a trace program, and a sort program.

(2) A CP program initiated by a command that must be invoked from CMS.

(3) A CMS program initiated by a command that is not intended for general users and typically requires additional authorization.

**V**

**V=V** See virtual=virtual.

**valid ALE**

The state of a host access-list entry that exists when the entry designates an address space and access permission for the space is still in effect. A valid host access-list entry can be used to access the subject address space when in access-register mode.

**variable symbol**

In an exec procedure, a symbol beginning with an ampersand (&) character, the value of which is assigned by the user, or sometimes the EXEC 2 processor or CMS EXEC processor. The value of a variable symbol can be tested and changed using control statements. See also special variable.

**variable-length record**

A record having a length independent of the length of other records with which it is logically or physically associated. The length of the record is included as part of the record. See also fixed-length record.

**VCIT** See virtual configuration identification token.

**VEPA** See Virtual Edge Port Aggregator.

**verification setting**

A setting in the CMS Editor or XEDIT that determines whether new lines entered, or the results of editing changes entered by the user, are displayed at the user's terminal.

**version vector table**

A software inventory table that contains a history of all PTFs that have been applied to a product at a specific maintenance level. The filetype of this table is *VVTlvlid*, where *lvlid* identifies the maintenance level and is obtained from the AUX record of the control file that has been identified for the product.

**vertical polarization mode**

An operational mode of a partition in which PR/SM increases memory cache benefits by mapping logical processors to real processors and by changing these mappings as little as possible. In vertical polarization mode, entitlements are not distributed evenly among the logical processors in the partition. See also horizontal polarization mode.

**vertical-high (Vh) processor**

A type of logical processor that is entitled to 100% of a real processor. See also vertical-low processor and vertical-medium processor.

**vertical-low (Vl) processor**

A type of logical processor that is entitled to 0% of a real processor. See also vertical-high processor and vertical-medium processor.

**vertical-medium (Vm) processor**

A type of logical processor that is entitled to 50-100% of a real processor, except in cases for which the logical partition's entitlement is less than 50%. See also vertical-high processor and vertical-low processor.

**Vh** See vertical-high processor.

**virtual adapter**

See network interface card.

**virtual address**

The address of a location in virtual storage. A virtual address must be translated into a real address to process the data in processor storage.

**virtual address space**

See virtual storage.

**virtual architecture level**

In an SSI cluster, a value that CP assigns to a guest virtual machine at logon that is the maximal common subset of the architectural features (hardware architecture facilities and z/VM-supplied features) of the cluster members that belong to the virtual machine's assigned or default relocation domain. The guest can use only the architectural features that are included in the virtual architecture level. The guest can be freely relocated to other members in the same relocation domain because those members provide the same architectural features.

**virtual block multiplexer mode**

A virtual machine option that allows the interleaving of data to different virtual devices on the same virtual channel path.

**virtual card reader**

A card reader simulated by CP for a virtual machine. The virtual device type and I/O device address are defined in the virtual machine definition or by the DEFINE command.

**virtual configuration**

See virtual machine configuration.

**virtual configuration identification token (VCIT)**

An 8-byte token that uniquely identifies a particular virtual machine. The VCIT is a system-wide, unique identification token—after a VCIT value has been assigned to a virtual machine, that VCIT value is not used again within the scope of a z/VM system IPL. The VCIT for a virtual machine is equal to the ASIT that identifies the virtual machine's host-primary address space.

**virtual console**

A console simulated by CP on a terminal for a virtual machine. The virtual device type and I/O device address are defined in the virtual machine definition or by the DEFINE command.

**virtual console function**

A Diagnose code (X'08') used to issue CP commands from programs running in a virtual machine.

**virtual console spooling**

The writing of console I/O on disk as a printer spool file instead of, or in addition to, having it typed or displayed at the virtual machine console. The console data includes messages, responses, commands, and data from or to CP and the virtual machine operating system. The user can invoke or terminate console spooling at any time. When the console spool file is closed, it becomes a printer spool file.

**virtual CPU**

See virtual processor.

**virtual CPU time**

The time used to execute the instructions of the virtual machine.

**virtual device**

The simulation of a device by CP. See also virtual device control block.

**virtual device control block**

A control block that describes the status of a real or virtual I/O device that can be accessed by a virtual machine. A virtual device control block defines the device to the virtual machine, whereas the real device control block defines the device to the system.

**virtual disk**

See minidisk.

**virtual disk in storage**

A temporary virtual disk mapped to storage (memory) rather than the cylinders of a real disk, which avoids the need for disk I/O. A virtual disk in storage can be allocated to the user during logon, in which case the disk is shareable, or when entering the CP DEFINE command, in which case the disk is nonshareable (private). In an SSI cluster, a virtual disk in storage is always local (limited to one member of the cluster).

**virtual disk initialization program**

A program supplied by VSE that can initialize virtual as well as real disks for use by VSE running under z/VM.

**Virtual Edge Port Aggregator (VEPA)**

A mode of operation that provides the capability to take all virtual machine traffic sent by the server and send it to an adjacent network switch. This removes all frame relay switching from the hypervisor virtual switch to the (external) adjacent switch. With the adjacent switch handling the frame relay for virtual switch guest port to guest port communications, imbedded network appliances in the adjacent switch, such as firewalls, Access Control Lists (ACLs), Quality of Service (QoS), and port mirroring, are available to be deployed for this guest port to guest port switching. VEPA eliminates the need to provide and support these network based appliances in the hypervisor or LPAR.

**virtual local area network (VLAN)**

A mechanism described by IEEE standard 802.1q that enables groups of hosts to be logically connected into a single local area network (LAN) even though they are connected to different switches in different physical locations.

**virtual machine**

(1) A virtual data processing system that appears to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real data processing system.

(2) In z/VM, the virtual processors, virtual storage, virtual devices, and virtual channel subsystem that the Control Program (CP) allocates to a single user. A virtual machine also includes any expanded storage dedicated to it. See also guest virtual machine, service pool virtual machine, and service virtual machine.

**virtual machine communication facility (VMCF)**

A CP function that provides a method of communication and data transfer between virtual machines operating under the same z/VM system.

**virtual machine configuration**

The specification of a virtual machine instance. A single-configuration virtual machine definition contains one virtual machine configuration, which consists of a user entry and any included profile entry. A multiconfiguration virtual machine definition can contain multiple configurations, each of which provides the specification for the instance of that virtual machine on a particular member of an SSI cluster. Each configuration consists of the common identity entry, any included profile entry, and a member-specific subconfiguration entry.

**virtual machine definition**

The set of statements in the source directory that defines the configuration and operating characteristics of a virtual machine. There are two types of virtual machine definitions: single-configuration virtual machine definitions and multiconfiguration virtual machine definitions.

**virtual machine descriptor block (VMDBK)**

The primary control block for many activities related to a single virtual machine. This block contains the following types of information: the dispatch and priority level of the virtual machine, the virtual machine's processor registers, preferred virtual machine options currently in effect, and information concerning all other significant activities.

**virtual machine dump**

A dump limited to any single virtual machine running on the z/VM system.

**virtual machine group**

In GCS, two or more virtual machines associated with each other through the same named saved system (for example, GCS1). Virtual machines in a group share common read/write storage and can communicate with one another through facilities provided by GCS. See also single user group.

**virtual machine instance**

The instantiation of a virtual machine from its configuration in the object directory. CP creates the virtual machine instance when the user ID is logged on.

**virtual machine mode**

A designation that indicates the architecture of a virtual machine. See also ESA virtual machine, XA virtual machine, and XC virtual machine.

**virtual machine operator**

The user who logs onto a virtual machine. See also system operator.

**virtual machine operator's console**

The virtual machine operator's logon terminal, which simulates the hardware system console

functions for the guest operating system. Messages destined for the hardware system console are presented to the virtual machine operator's console.

**virtual machine reset**

A collective term for the simulation of the system-reset-normal or system-reset-clear processes defined by the hardware architectures. Among other things, virtual machine reset is an event that causes CP to recover many resources associated with a virtual machine, such as IUCV or APPC paths and data spaces. Virtual machine reset can result from issuing such CP commands as SYSTEM CLEAR, SYSTEM RESET, IPL or LOGOFF.

**virtual machine resource manager (VMRM)**

In z/VM, a set of functions that dynamically tune the system by managing the performance goals of workloads, which are groups of virtual machines.

**Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E)**

A component of z/VM that provides the tools for installing and servicing the various components of the z/VM product. The component is also the strategic installation and service tool for all of the other products that run on those z/VM platforms.

**virtual machine time bomb**

A mechanism for allowing a virtual machine to cause an action to be performed if the virtual machine becomes unresponsive.

**virtual network adapter**

The collection of virtual I/O devices and logic connecting host programs to the network.

**virtual printer**

A printer simulated by CP for a virtual machine. The virtual device type and I/O address are defined in the virtual machine definition or by the DEFINE command.

**virtual processor**

A representation of a processor that is dispatched by CP on a real processor. The virtual processor includes the contents of all registers and the state of the processor. z/VM supports various processor types, including CP (central processor) and IFL (IBM Integrated Facility for Linux).

**virtual punch**

A card punch simulated by CP for a virtual machine. The virtual device type and I/O address are defined in the virtual machine definition or by the DEFINE command.

**virtual reader**

See [virtual card reader](#).

**virtual reserve/release**

A function that provides data protection on a minidisk. The function prevents users of the same data file from simultaneously accessing the same data, particularly when that data is being updated.

**virtual screen**

In full-screen CMS, a functional simulation of a physical screen. A virtual screen is a *presentation space* where data is maintained. The user can view pieces of the virtual screen through a window on the physical screen.

**virtual server**

(1) In z/VM, a virtual machine and a guest operating system or application running in that virtual machine, taken as a whole.

(2) In an ensemble, a logical construct that appears to comprise processor, memory, and I/O resources conforming to a particular architecture. A virtual server can support an operating system, associated middleware, and applications. A hypervisor creates and manages virtual servers.

**virtual spooling device**

A unit record device simulated by CP. The virtual device type and I/O address of the virtual

spooling device are defined in the z/VM user directory or by the CP DEFINE command. The CP spooling facilities permit multiple virtual machines to share the real unit record devices. In addition, z/VM has extended the concept of spooling to include the logging of all I/O to and from the virtual machine console.

**virtual storage**

Storage space that can be regarded as addressable main storage in which virtual addresses are mapped into real addresses.

**Virtual Storage Access Method (VSAM)**

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

**virtual supervisor state**

A condition, controlled by a virtual machine's current PSW, during which the control program lets the virtual machine issue I/O and other privileged instructions. When these instructions are not emulated, the control program intercepts these instructions and simulates their functions for the virtual machine.

**virtual switch**

See z/VM virtual switch.

**virtual unit record device**

See virtual spooling device.

**virtual wait time**

The period during which the control program suspends the processing of a program while a required resource is unavailable.

**virtual=virtual (V=V)**

An attribute of a virtual machine indicating that it runs in the dynamic paging area. CP pages the virtual machine's guest real storage in and out of host real storage.

**VI** See vertical-low processor.

**VLAN** See virtual local area network.

**Vm** See vertical-medium processor.

**VM configuration token**

A 64-byte token created by z/VM that is compared to a similar 64-byte token created by the hardware to synchronize both views of the I/O configuration during dynamic I/O changes.

**VM data spaces**

A facility of z/VM that uses ESA/390 interpretive execution (SIE) enhancements to provide the ESA/XC virtual machine architecture, which allows virtual machines to create and access multiple host-managed address spaces.

**VM directory**

See user directory.

**VM domain controller**

A part of the ISFC function of CP that allows user programs to connect to and communicate with resources on different VM systems in a Communication Services (CS) collection.

**VM Dump Tool**

A CP facility that assists in analyzing data from CP stand-alone dumps, CP ABEND dumps, and virtual machine dumps of a CP system. The VM Dump Tool allows the user to interactively display, locate, and format CP dump data. Analysis of all other virtual machine dumps is performed by the Dump Viewing Facility. See also Dump Viewing Facility.

**VM LAN**

A LAN (local area network) that is simulated by z/VM. A VM LAN connects virtual machines in a z/VM system environment using virtual network adapters instead of dedicated hardware.

**VM READ**

On a display terminal used as a virtual console under z/VM, a screen status that indicates the user's virtual machine is not executing, but is waiting for a response or a request for work from the user. See also [screen status area](#).

**VM/VS handshaking**

A communication interface between CP and a VSE operating system running in a virtual machine that allows them to make each other aware of mutual capabilities and requirements.

**VMCF**

See [virtual machine communication facility](#).

**VMDBK**

See [virtual machine descriptor block](#).

**VMLIB**

The name of the callable services library supplied with z/VM and that contains routines to do various VM functions.

**VMRM**

See [virtual machine resource manager](#).

**VMRM cooperative memory management (VMRM-CMM)**

A part of z/VM's virtual machine resource manager that helps Linux guests manage memory constraints in the system.

**VMRM-CMM**

See [VMRM cooperative memory management](#).

**VMSES/E**

See [Virtual Machine Serviceability Enhancements Staged/Extended](#).

**volid** [volume identifier](#).

**volume**

The DASD space identified by a common label and accessed by any of a set of related addresses.

**volume identifier (volid)**

The volume identification label for a DASD volume.

**volume table of contents (VTOC)**

A table on a direct access volume that describes each data set on the volume.

**VSAM**

See [Virtual Storage Access Method](#).

**VSM** See [VTAM service machine](#).

**VTAM service machine (VSM)**

A virtual machine that acts as an interface between SNA/CCS and the SNA network. The VTAM service machine runs programs that work with Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM) to manage the SNA network.

**VTOC** See [volume table of contents](#).

**W****wait state**

A situation in which a program or process is suspended. See also [disabled wait state](#) and [enabled wait state](#).

**WAN** See [wide area network](#).

**warm start**

A z/VM system restart that does not erase previous system data. Closed spool files and the z/VM accounting information are not lost. See also [checkpoint start](#), [clean start](#), [cold start](#), and [force start](#).

**well-known port**

A port number that has been preassigned for specific use by a specific protocol or application. Clients and servers using the same protocol can communicate over the same well-known port.

**white space**

In OpenExtensions, space characters, tab characters, newline characters, and comments.

**wide area network (WAN)**

A network that provides communication services to a geographic area larger than that served by a local area network, and that might use or provide public communication facilities. (T) See also [local area network](#).

**widget**

A reusable user interface component such as a button, scrollbar, control area, or text edit area, that can receive input from the keyboard or mouse and can communicate with an application or with another widget. Widgets are part of an X Window System toolkit.

**wildcard character**

(1) In z/VM, a symbol used in a command to represent zero or more characters. The wildcard characters are the asterisk (\*), representing zero or more characters, and the percent symbol (%), representing exactly one character.

(2) In OpenExtensions, a symbol that can be used in a file specification for zero or more characters. The wildcard characters are the asterisk (\*), representing zero or more characters, and the question mark (?), representing exactly one character. See also [regular expression](#).

**window**

In full-screen CMS, an area on the physical screen where virtual screen data can be displayed.

**windowing**

In full-screen CMS, a set of functions that allow the user to view and manipulate data in user-defined areas of the physical screen called *windows*. Windowing support lets the user define, position, and overlay windows; scroll backward and forward through data; and write data into virtual screens.

**work unit**

In CMS, a group of related operations that can be either committed or rolled back as a unit. When the operations associated with a work unit are committed or rolled back, new operations can be associated with the same work unit. Multiple work units can be active. See also [active work unit](#), [inactive work unit](#), and [logical unit of work](#).

**work unit ID**

A nonnegative integer value, unique within a virtual machine, that identifies a work unit.

**working directory**

The directory that is currently in use by an operating system or application. If no path is specified, this is the directory to which data is written, from which data is deleted, or in which data is searched. The active directory used to resolve path names that do not begin with a slash. See also [root directory](#).

**working set**

The estimated number of pages of real storage that the virtual machine needs to execute.

**write access**

Permission to write to an object.

**write authority**

The authority to read or change the contents of a file or directory. Write authority implies read authority.

**write permission**

See write access.

**WRITE-ONLY Rule**

A requirement that the security label of the object dominates that of the subject. This prevents a subject from writing down. See also READ-ONLY Rule and READ/WRITE Rule.

**writing down**

An event in which a subject with a given security label writes in an object whose security label the subject does not dominate. See also Domination Rule and reading up.

**X****X client**

A program that uses the X protocol to communicate windowing and graphics requests to an X server.

**X Intrinsic**

See X Toolkit Intrinsic.

**X protocol**

A client-server protocol that is part of the X Window System.

**X server**

A program that interprets the X protocol and controls one or more screens, a pointing device, a keyboard, and various resources associated with the X Window System, such as graphics contexts, pixmaps, and color tables.

**X Toolkit Intrinsic**

A library used in the X Window System that uses the low-level Xlib library and provides an API to develop X Window System software with graphical widgets. X Toolkit Intrinsic can be used in the C or C++ languages.

**X Window System**

A computer software system and network protocol designed to support network transparent windowing and graphics. TCP/IP for z/VM provides client support for the X Window System API.

**X Window System API**

In z/VM, a set of application programming interfaces from the X Window System Version 11, Release 4 and Release 1.1 of the OSF/Motif-based widget set.

**X Window System toolkit**

A set of library functions layered on top of the X Window System Xlib functions that allows you to simplify the design of applications by providing an underlying set of common user interface functions.

**X Windows**

See X Window System.

**XA virtual machine**

A virtual machine designation, supported for compatibility with previous versions of VM, that is functionally equivalent to an ESA virtual machine.

**XC virtual machine**

A virtual machine that provides Enterprise Systems Architecture/Extended Configuration (ESA/XC) functions.

**xCAT** See Extreme Cloud Administration Toolkit.

**XDR** See [eXternal Data Representation](#).

## **XEDIT**

The CMS facility that provides the editor environment in which a user can create, change, and manipulate CMS files.

### **XEDIT macro**

An exec invoked from the XEDIT environment. An XEDIT macro must have a file type of XEDIT and can contain XEDIT subcommands, exec instructions, and CMS and CP commands.

### **XEDIT profile macro**

A special XEDIT macro with a file name of PROFILE that a user can create. It is automatically executed when the XEDIT command or XEDIT subcommand is entered, unless the NOPROFILE option is specified.

### **XEDIT subcommand**

A command that is valid only in the XEDIT environment.

**XRC** See [extended remote copy](#).

## **Y**

### **Y-STAT**

A block of storage that contains the FSTs associated with file mode Y. The FSTs are sorted so that a binary search can search for files. The Y-STAT usually resides in the CMS nucleus so it can be shared. Only files with file mode of 2 will have their associated FSTs in the Y-STAT. See also [S-STAT](#).

## **Z**

### **z/Architecture**

An IBM mainframe computer and operating system architecture that includes most of the facilities of ESA/390 and provides significant extensions such as 64-bit registers and addressing.

### **z/Architecture CMS (z/CMS)**

A version of CMS that runs in z/Architecture 31-bit addressing mode in an ESA or XA virtual machine and enables CMS programs to use z/Architecture instructions, including those that operate on 64-bit registers, while permitting existing ESA/390 architecture CMS programs to continue to function without change. See also [ESA/390 CMS](#).

### **z/CMS**

See [z/Architecture CMS](#).

### **z/VM HiperDispatch**

A z/VM function designed to improve a workload's performance in the memory subsystem, in particular, by taking advantage of the memory cache structure.

### **z/VM single system image (SSI) cluster**

A z/VM multisystem environment in which the member systems can be managed, serviced, and administered as one system. An SSI cluster includes capabilities such as multisystem installation, single point of maintenance, autonomic minidisk cache management, and live guest relocation.

### **z/VM system DDR**

A tape or DVD containing the image of a built z/VM system for various types of DASD.

### **z/VM System Delivery Offering**

An IBM offering that includes a single deliverable containing z/VM and a group of fully serviced IBM licensed program products.

### **z/VM virtual switch**

A virtualized representation of a hardware LAN switch. A virtual switch is capable of bridging a z/VM guest LAN to an associated real LAN connected by an OSA-Express adapter.

**zAAP** See [IBM System z Application Assist Processor](#).

- zap** To modify or dump an individual text file, using the ZAP or ZAPTEXT command.
- zBX** See IBM zEnterprise BladeCenter Extension.
- zEDC** See IBM zEnterprise Data Compression.
- zEnterprise**  
See IBM zEnterprise System.
- zIIP** See IBM System z Integrated Information Processor.
- zone** (1) In the Domain Name System, a logical grouping of domain names that is assigned to a particular organization.
- (2) In XEDIT, a starting and ending column position of each record to be scanned for target searches and that is set through the SET ZONE subcommand. The editor searches for targets only within the zone.

---

## Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Site Counsel  
2455 South Road  
Poughkeepsie, NY 12601-5400  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The

sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

---

## Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see the IBM Online Privacy Policy at <http://www.ibm.com/privacy> and the IBM Online Privacy Statement at <http://www.ibm.com/privacy/details>, in particular the section entitled "Cookies, Web Beacons and Other Technologies", and the IBM Software Products and Software-as-a-Service Privacy Statement at <http://www.ibm.com/software/info/product-privacy>.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at IBM copyright and trademark information - United States ([www.ibm.com/legal/us/en/copytrade.shtml](http://www.ibm.com/legal/us/en/copytrade.shtml)).

PostScript is a registered trademark or trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java is a trademark or registered trademark of Oracle and/or its affiliates.



---

## Bibliography

See the following publications for additional information about z/VM. For abstracts of the z/VM publications, see *z/VM: General Information*, GC24-6193

---

### Where to get z/VM information

z/VM product information is available from the following sources:

- IBM Knowledge Center z/VM welcome page ([www.ibm.com/support/knowledgecenter/SSB27U/welcome](http://www.ibm.com/support/knowledgecenter/SSB27U/welcome))
- IBM Publications Center ([www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss](http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss))
- *IBM Online Library: z/VM Collection*, SK5T-7054
- IBM: z/VM Internet Library ([www.ibm.com/vm/library/](http://www.ibm.com/vm/library/))

---

### z/VM base library

#### Overview

- *z/VM: General Information*, GC24-6193
- *z/VM: Glossary*, GC24-6195
- *z/VM: License Information*, GC24-6200

#### Installation, migration, and service

- *z/VM: Installation Guide*, GC24-6246
- *z/VM: Migration Guide*, GC24-6201
- *z/VM: Service Guide*, GC24-6247
- *z/VM: VMSES/E Introduction and Reference*, GC24-6243

#### Planning and administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6167
- *z/VM: CMS Planning and Administration*, SC24-6171
- *z/VM: Connectivity*, SC24-6174
- *z/VM: CP Planning and Administration*, SC24-6178
- *z/VM: Enabling z/VM for OpenStack (Support for OpenStack Icehouse Release)*, SC24-6248
- *z/VM: Getting Started with Linux on System z*, SC24-6194
- *z/VM: Group Control System*, SC24-6196

- *z/VM: I/O Configuration*, SC24-6198
- *z/VM: Running Guest Operating Systems*, SC24-6228
- *z/VM: Saved Segments Planning and Administration*, SC24-6229
- *z/VM: Secure Configuration Guide*, SC24-6230
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6236
- *z/VM: TCP/IP Planning and Customization*, SC24-6238
- *z/OS and z/VM: Hardware Configuration Manager User's Guide*, SC33-7989

#### Customization and tuning

- *z/VM: CP Exit Customization*, SC24-6176
- *z/VM: Performance*, SC24-6208

#### Operation and use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6166
- *z/VM: CMS Pipelines Reference*, SC24-6169
- *z/VM: CMS Pipelines User's Guide*, SC24-6170
- *z/VM: CMS Primer*, SC24-6172
- *z/VM: CMS User's Guide*, SC24-6173
- *z/VM: CP Commands and Utilities Reference*, SC24-6175
- *z/VM: System Operation*, SC24-6233
- *z/VM: TCP/IP User's Guide*, SC24-6240
- *z/VM: Virtual Machine Operation*, SC24-6241
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6244
- *z/VM: XEDIT User's Guide*, SC24-6245

#### Application programming

- *z/VM: CMS Application Development Guide*, SC24-6162
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6163
- *z/VM: CMS Application Multitasking*, SC24-6164
- *z/VM: CMS Callable Services Reference*, SC24-6165
- *z/VM: CMS Macros and Functions Reference*, SC24-6168
- *z/VM: CP Programming Services*, SC24-6179
- *z/VM: CPI Communications User's Guide*, SC24-6180

- z/VM: *Enterprise Systems Architecture/Extended Configuration Principles of Operation*, SC24-6192
- z/VM: *Language Environment User's Guide*, SC24-6199
- z/VM: *OpenExtensions Advanced Application Programming Tools*, SC24-6202
- z/VM: *OpenExtensions Callable Services Reference*, SC24-6203
- z/VM: *OpenExtensions Commands Reference*, SC24-6204
- z/VM: *OpenExtensions POSIX Conformance Document*, GC24-6205
- z/VM: *OpenExtensions User's Guide*, SC24-6206
- z/VM: *Program Management Binder for CMS*, SC24-6211
- z/VM: *Reusable Server Kernel Programmer's Guide and Reference*, SC24-6220
- z/VM: *REXX/VM Reference*, SC24-6221
- z/VM: *REXX/VM User's Guide*, SC24-6222
- z/VM: *Systems Management Application Programming*, SC24-6234
- z/VM: *TCP/IP Programmer's Reference*, SC24-6239
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- z/OS: *IBM Tivoli Directory Server Plug-in Reference for z/OS*, SA76-0148
- z/OS: *Language Environment Concepts Guide*, SA22-7567
- z/OS: *Language Environment Debugging Guide*, GA22-7560
- z/OS: *Language Environment Programming Guide*, SA22-7561
- z/OS: *Language Environment Programming Reference*, SA22-7562
- z/OS: *Language Environment Run-Time Messages*, SA22-7566
- z/OS: *Language Environment Writing Interlanguage Communication Applications*, SA22-7563
- z/OS MVS *Program Management: Advanced Facilities*, SA22-7644
- z/OS MVS *Program Management: User's Guide and Reference*, SA22-7643

## Diagnosis

- z/VM: *CMS and REXX/VM Messages and Codes*, GC24-6161
- z/VM: *CP Messages and Codes*, GC24-6177

- z/VM: *Diagnosis Guide*, GC24-6187
- z/VM: *Dump Viewing Facility*, GC24-6191
- z/VM: *Other Components Messages and Codes*, GC24-6207
- z/VM: *TCP/IP Diagnosis Guide*, GC24-6235
- z/VM: *TCP/IP Messages and Codes*, GC24-6237
- z/VM: *VM Dump Tool*, GC24-6242
- z/OS and z/VM: *Hardware Configuration Definition Messages*, SC33-7986

---

## z/VM facilities and features

### Data Facility Storage Management Subsystem for VM

- z/VM: *DFSMS/VM Customization*, SC24-6181
- z/VM: *DFSMS/VM Diagnosis Guide*, GC24-6182
- z/VM: *DFSMS/VM Messages and Codes*, GC24-6183
- z/VM: *DFSMS/VM Planning Guide*, SC24-6184
- z/VM: *DFSMS/VM Removable Media Services*, SC24-6185
- z/VM: *DFSMS/VM Storage Administration*, SC24-6186

### Directory Maintenance Facility for z/VM

- z/VM: *Directory Maintenance Facility Commands Reference*, SC24-6188
- z/VM: *Directory Maintenance Facility Messages*, GC24-6189
- z/VM: *Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190

### Open Systems Adapter/Support Facility

- zEnterprise System and System z10: *Open Systems Adapter-Express Customer's Guide and Reference*, SA22-7935
- System z9 and eServer zSeries 890 and 990: *Open Systems Adapter-Express Integrated Console Controller User's Guide*, SA22-7990
- System z: *Open Systems Adapter-Express Integrated Console Controller 3215 Support*, SA23-2247
- System z10: *Open Systems Adapter-Express3 Integrated Console Controller Dual-Port User's Guide*, SA23-2266

### Performance Toolkit for VM

- z/VM: *Performance Toolkit Guide*, SC24-6209
- z/VM: *Performance Toolkit Reference*, SC24-6210

## **RACF Security Server for z/VM**

- *z/VM: RACF Security Server Auditor's Guide, SC24-6212*
- *z/VM: RACF Security Server Command Language Reference, SC24-6213*
- *z/VM: RACF Security Server Diagnosis Guide, GC24-6214*
- *z/VM: RACF Security Server General User's Guide, SC24-6215*
- *z/VM: RACF Security Server Macros and Interfaces, SC24-6216*
- *z/VM: RACF Security Server Messages and Codes, GC24-6217*
- *z/VM: RACF Security Server Security Administrator's Guide, SC24-6218*
- *z/VM: RACF Security Server System Programmer's Guide, SC24-6219*
- *z/VM: Security Server RACROUTE Macro Reference, SC24-6231*

## **Remote Spooling Communications Subsystem Networking for z/VM**

- *z/VM: RSCS Networking Diagnosis, GC24-6223*

- *z/VM: RSCS Networking Exit Customization, SC24-6224*
- *z/VM: RSCS Networking Messages and Codes, GC24-6225*
- *z/VM: RSCS Networking Operation and Use, SC24-6226*
- *z/VM: RSCS Networking Planning and Configuration, SC24-6227*
- *Network Job Entry: Formats and Protocols, SA22-7539*

---

## **Prerequisite products**

### **Device Support Facilities**

- *Device Support Facilities: User's Guide and Reference, GC35-0033*

### **Environmental Record Editing and Printing Program**

- *Environmental Record Editing and Printing Program (EREP): Reference, GC35-0152*
- *Environmental Record Editing and Printing Program (EREP): User's Guide, GC35-0151*







Product Number: 5741-A07

Printed in USA

GC24-6195-04

