

GLOSSARY

- Abortive release** An abrupt termination of a transport connection, which may result in the loss of data.
- Ada** Named after the Countess of Lovelace, the nineteenth century mathematician and computer pioneer, Ada is a high-level general-purpose programming language developed under the sponsorship of the U.S. Department of Defense. Ada was developed to provide consistency among programs originating in different branches of the military. Ada features include packages that make data objects visible only to the modules that need them, task objects that facilitate parallel processing, and an exception handling mechanism that encourages well-structured error processing.
- ANSI standard** ANSI is the acronym for the American National Standards Institute. ANSI establishes guidelines in the computing industry, from the definition of ASCII to the determination of overall datacom system performance. ANSI standards have been established for both the Ada and FORTRAN programming languages, and a standard for C has been proposed.
- a.out file** **a.out** is the default file name used by the link editor when it outputs a successfully compiled, executable file. **a.out** contains object files that are combined to create a complete working program. Object file format is described in Chapter 11, "The Common Object File Format," and in **a.out(4)** in the *Programmer's Reference Manual*.
- application program** An application program is a working program in a system. Such programs are usually unique to one type of users' work, although some application programs can be used in a variety of business situations. An accounting application, for example, may well be applicable to many different businesses.
- archive** An archive file or archive library is a collection of data gathered from several files. Each of the files within an archive is called a member. The command **ar(1)** collects data for use as a library.

GLOSSARY

argument

An argument is additional information that is passed to a command or a function. On a command line, an argument is a character string or number that follows the command name and is separated from it by a space. There are two types of command-line arguments: options and operands. Options are immediately preceded by a minus sign (-) and change the execution or output of the command. Some options can themselves take arguments. Operands are preceded by a space and specify files or directories that will be operated on by the command. For example, in the command

pr -t -h Heading file

all of the elements after the **pr** are arguments. **-t** and **-h** are options, **Heading** is an argument to the **-h** option, and **file** is an operand.

For a function, arguments are enclosed within a pair of parentheses immediately following the function name. The number of arguments can be zero or more; if more than two are present they are separated by commas and the whole list enclosed by the parentheses. The formal definition of a function, such as might be found on a page in Section 3 of the *Programmer's Reference Manual*, describes the number and data type of argument(s) expected by the function.

ASCII

ASCII is an acronym for American Standard Code for Information Interchange, a standard for data representation that is followed in the operating system. ASCII code represents alphanumeric characters as binary numbers. The code includes 128 upper- and lower-case letters, numerals, and special characters. Each alphanumeric and special character has an ASCII code (binary) equivalent that is one byte long.

assembler

The assembler is a translating program that accepts instructions written in the assembly language of the computer and translates them into the binary representation of machine instructions. In many cases, the assembly language instructions map 1 to 1 with the binary machine instructions.

- assembly language** A programming language that uses the instruction set that applies to a particular computer.
- Asynchronous execution** The mode of execution in which Transport Interface routines will never block while waiting for specific asynchronous events to occur, but instead will return immediately if the event is not pending.
- BASIC** BASIC is a high-level conversational programming language that allows a computer to be used much like a complex electronic calculating machine. The name is an acronym for Beginner's All-purpose Symbolic Instruction Code.
- branch table** A branch table is an implementation technique for fixing the addresses of text symbols, without forfeiting the ability to update code. Instead of being directly associated with function code, text symbols label jump instructions that transfer control to the real code. Branch table addresses do not change, even when one changes the code of a routine. Jump table is another name for branch table.
- buffer** A buffer is a storage space in computer memory where data are stored temporarily into convenient units for system operations. Buffers are often used by programs, such as editors, that access and alter text or data frequently. When you edit a file, a copy of its contents are read into a buffer where you make changes to the text. For the changes to become part of the permanent file, you must write the buffer contents back into the permanent file. This replaces the contents of the file with the contents of the buffer. When you quit the editor, the contents of the buffer are flushed.
- byte** A byte is a unit of storage in the computer. On many systems, a byte is eight bits (binary digits), the equivalent of one character of text.
- byte order** Byte order refers to the order in which data are stored in computer memory.

GLOSSARY

- C** The C programming language is a general-purpose programming language that features economy of expression, control flow, data structures, and a variety of operators. It can be used to perform both high-level and low-level tasks. Although it has been called a system programming language, because it is useful for writing operating systems, it has been used equally effectively to write major numerical, text-processing, and data base programs.
- C compiler** The C compiler converts C programs into assembly language programs that are eventually translated into object files by the assembler.
- C preprocessor** The C preprocessor is a component of the C Compilation System. In C source code, statements preceded with a pound sign (#) are directives to the preprocessor. Command line options of the **cc(1)** command may also be used to control the actions of the preprocessor. The main work of the preprocessor is to perform file inclusions and macro substitution.
- CCS** CCS is an acronym for C Compilation System, which is a set of programming language utilities used to produce object code from C source code. The major components of a C Compilation System are a C preprocessor, C compiler, assembler, and link editor. The C preprocessor accepts C source code as input, performs any preprocessing required, then passes the processed code to the C compiler, which produces assembly language code that it passes to the assembler. The assembler in turn produces object code that can be linked to other object files by the link editor. The object files produced are in the Common Object File Format (COFF). Other components of CCS include a symbolic debugger, an optimizer that makes the code produced as efficient as possible, productivity tools, tools used to read and manipulate object files, and libraries that provide runtime support, access to system calls, input/output, string manipulation, mathematical functions, and other code processing functions.

Client	The transport user in connection-mode that initiates the establishment of a transport connection.
COBOL	COBOL is an acronym for Common Business Oriented Language. COBOL is a high-level programming language designed for business and commercial applications. The English-language statements of COBOL provide a relatively machine-independent method of expressing a business-oriented problem to the computer.
COFF	<p>COFF is an acronym for Common Object File Format. COFF refers to the format of the output file produced on some systems by the assembler and the link editor. This format is also used by other operating systems. The following are some of its key features:</p> <ul style="list-style-type: none">□ Applications may add system-dependent information to the object file without causing access utilities to become obsolete.□ Space is provided for symbolic information used by debuggers and other applications.□ Users may make some modifications in the object file construction at compile time.
command	A command is the term commonly used to refer to an instruction that a user types at a computer terminal keyboard. It can be the name of a file that contains an executable program or a shell script that can be processed or executed by the computer on request. A command is composed of a word or string of letters and/or special characters that can continue for several (terminal) lines, up to 256 characters. A command name is sometimes used interchangeably with a program name.
command line	A command line is composed of the command name followed by any argument(s) required by the command or optionally included by the user. The manual page for a command includes a command line synopsis in a notation designed to show the correct way to type in a command, with or without options and arguments.

GLOSSARY

- compiler** A compiler transforms the high-level language instructions in a program (the source code) into object code or assembly language. Assembly language code may then be passed to the assembler for further translation into machine instructions.
- Connection establishment** The phase in connection-mode that enables two transport users to create a transport connection between them.
- Connection-mode** A circuit-oriented mode of transfer in which data are passed from one user to another over an established connection in a reliable, sequenced manner.
- Connectionless-mode** A mode of transfer in which data are passed from one user to another in self-contained units with no logical relationship required among multiple units.
- Connection release** The phase in connection-mode that terminates a previously established transport connection between two users.
- core** Core is a (mostly archaic) synonym for primary memory.
- core file** A core file is an image of a terminated process saved for debugging. A core file is created under the name "core" in the current directory of the process when an abnormal event occurs resulting in the process' termination. A list of these events is found in the **signal(2)** manual page in section 2 of the *Programmer's Reference Manual*.
- core image** Core image is a copy of all the segments of a running or terminated program. The copy may exist in main storage, in the swap area. or in a core file.
- curses** **curses(3X)** is a library of C routines that are designed to handle input, output, and other operations in screen management programs. The name **curses** comes from the cursor optimization that the routines provide. When a screen management program is run, cursor optimization minimizes the amount of time a cursor has to move about a screen to update its contents. The program refers to the **terminfo(4)** data base at run time to obtain the information that it needs about the screen (terminal) being used. See **terminfo(4)** in the *Programmer's Reference Manual*.

data base	A data base is a bank of information on a particular subject or subjects. On-line data bases are designed so that by using subject headings, key words, or key phrases you can search for, analyze, update, and print out data.
Datagram	A unit of data transferred between two users of the connectionless-mode service.
data symbol	A data symbol names a variable that may or may not be initialized. Normally, these variables reside in read/write memory during execution. See text symbol.
Data transfer	The phase in connection-mode or connectionless-mode that supports the transfer of data between two transport users.
debug	Debugging is the process of locating and correcting errors in computer programs.
default	A default is the way a computer will perform a task in the absence of other instructions.
delimiter	A delimiter is an initial character that identifies the next character or character string as a particular kind of argument. Delimiters are typically used for option names on a command line; they identify the associated word as an option (or as a string of several options if the options are bundled). In the operating system command syntax, a minus sign (-) is most often the delimiter for option names, for example, <code>-s</code> or <code>-n</code> , although some commands also use a plus sign (+).
directory	<p>A directory is a type of file used to group and organize other files or directories. A directory consists of entries that specify further files (including directories) and constitutes a node of the file system. A subdirectory is a directory that is pointed to by a directory one level above it in the file system organization.</p> <p>The <code>ls(1)</code> command is used to list the contents of a directory. When you first log onto the system, you are in your home directory (<code>\$HOME</code>). You can move to another directory by using the <code>cd(1)</code> command and you can print the name of the current directory by using the <code>pwd(1)</code> command. You can also create new directories with the <code>mkdir(1)</code> command and remove empty directories with</p>

GLOSSARY

rmdir(1).

A directory name is a string of characters that identifies a directory. It can be a simple directory name, the relative path name or the full path name of a directory.

dynamic linking

Dynamic linking refers to the ability to resolve symbolic references at run time. Systems that use dynamic linking can execute processes without resolving unused references. See static linking.

environment

An environment is a collection of resources used to support a function. In the operating system, the shell environment is composed of variables whose values define the way you interact with the system. For example, your environment includes your shell prompt string, specifics for backspace and erase characters, and commands for sending output from your terminal to the computer.

An environment variable is a shell variable such as \$HOME (which stands for your login directory) or \$PATH (which is a list of directories the shell will search through for executable commands) that is part of your environment. When you log in, the system executes programs that create most of the environmental variables that you need for the commands to work. These variables come from `/etc/profile`, a file that defines a general working environment for all users when they log onto a system. In addition, you can define and set variables in your personal `.profile` file, which you create in your login directory to tailor your own working environment. You can also temporarily set variables at the shell level.

executable file

An executable file is a file that can be processed or executed by the computer without any further translation. That is, when you type in the file name, the commands in the file are executed. An object file that is ready to run (ready to be copied into the address space of a process to run as the code of that process) is an executable file. Files containing shell commands are also executable. A file may be given execute permission by using the `chmod(1)` command. In addition to being ready to run, a file needs to have execute permission.

- exit** A specific system call that causes the termination of a process. The `exit(2)` call will close any open files and clean up most other information and memory which was used by the process.
- exit status: return code** An exit status or return code is a code number returned to the shell when a command is terminated that indicates the cause of termination.
- Expedited data** Data that are considered urgent. The specific semantics of *expedited data* are defined by the transport protocol that provides the transport service.
- Expedited transport service data unit** The amount of expedited user data the identity of which is preserved from one end of a transport connection to the other (that is, an expedited message).
- exported symbol** A symbol that a shared library defines and makes available outside the library. See imported symbol.
- expression** An expression is a mathematical or logical symbol or meaningful combination of symbols. See regular expression.
- file** A file is an identifiable collection of information that, in the operating system, is a member of a file system. A file is known to the operating system as an inode plus the information the inode contains that tells whether the file is a plain file, a special file, or a directory. A plain file may contain text, data, programs or other information that forms a coherent unit. A special file is a hardware device or portion thereof, such as a disk partition. A directory is a type of file that contains the names and inode addresses of other plain, special or directory files.
- file and record locking** The phrase "file and record locking" refers to software that protects records in a data file against the possibility of being changed by two users at the same time. Records (or the entire file) may be locked by one authorized user while changes are made. Other users are thus prevented from working with the same record until the changes are completed.

GLOSSARY

- file descriptor** A file descriptor is a number assigned by the operating system to a file when the file is opened by a process. File descriptors 0, 1, and 2 are reserved; file descriptor 0 is reserved for standard input (**stdin**), 1 is reserved for standard output (**stdout**), and 2 is reserved for standard error output (**stderr**).
- file system** A file system is a hierarchical collection of directories and other files that are organized in a tree structure. The base of the structure is the root (**/**) directory; other directories, all subordinate to the root, are branches. The collection of files can be mounted on a block special file. Each file of a file system appears exactly once in the inode list of the file system and is accessible via a single, unique path from the root directory of the file system.
- filter** A filter is a program that reads information from standard input, acts on it in some way, and sends its results to standard output. It is called a filter because it can be used as a data transformer in a pipeline. Filters are different from editors and other commands because filters do not change the contents of a file. Examples of filters are **grep(1)** and **tail(1)**, which select and output part of the input; **sort(1)**, which sorts the input; and **wc(1)**, which counts the number of words, characters, and lines in the input. **sed(1)** and **awk(1)** are also filters but they are called programmable filters or data transformers because a program must be supplied as input in addition to the data to be transformed.
- flag** A flag or option is used on a command line to signal a specific condition to a command or to request particular processing. Operating system flags are usually indicated by a leading hyphen (**-**). The word option is sometimes used interchangeably with flag. Flag is also used as a verb to mean to point out or to draw attention to. See option.
- fork** **fork(2)** is a system call that divides a new process into two, the parent and child processes, with separate, but initially identical, text, data, and stack segments. After the duplication, the child (created) process is given a return code of 0 and the parent is given the process id of the newly created child as the return code.

- FORTRAN** **FORTRAN** is an acronym for **FOR**mula **TRAN**slator. **FORTRAN** is a high-level programming language originally designed for scientific and engineering calculations but now also widely adapted for many business uses.
- function** A function is a task done by a computer. In most modern programming languages, programs are made up of functions and procedures which perform small parts of the total job to be done.
- header file** A header file is used in programming and in document formatting. In a programming context, a header file is a file that usually contains shared data declarations that are to be copied into source programs as they are compiled. A header file includes symbolic names for constants, macro definitions, external variable references and inclusion of other header files. The name of a header file customarily ends with '.h' (dot-h). Similarly, in a document formatting context, header files contain general formatting macros that describe a common document type and can be used with many different document bodies.
- high-level language** A high-level language is a computer programming language such as C, FORTRAN, COBOL, or PASCAL that uses symbols and command statements representing actions the computer is to perform, the exact steps for a machine to follow. A high-level language must be translated into machine language by a compilation system before a computer can execute it. A characteristic of a high-level language is that each statement usually translates into a series of machine language instructions. The low-level details of the computer's internal organization are left to the compilation system.
- host machine** A host machine is the machine on which an **a.out** file is built.
- imported symbol** A symbol used but not defined by a shared library. See **exported symbol**.
- interpreted language** An interpreted language is a high-level language that is not translated by a compilation system and stored in an executable object file. The statements of a program in an interpreted language are translated each time the program is executed.

GLOSSARY

Interprocess Communication

Interprocess Communication describes software that enables independent processes running at the same time, to exchange information through messages, semaphores, or shared memory.

interrupt

An interrupt is a break in the normal flow of a system or program. Interrupts are initiated by signals that are generated by a hardware condition or a peripheral device indicating that a certain event has happened. When the interrupt is recognized by the hardware, an interrupt handling routine is executed. An interrupt character is a character (normally ASCII) that, when typed on a terminal, causes an interrupt. You can usually interrupt programs by pressing the delete or break keys, by typing Control-d, or by using the kill(1) command.

I/O (Input/Output)

I/O is the process by which information enters (input) and leaves (output) the computer system.

kernel

The kernel (comprising 5 to 10 percent of the operating system software) is the basic resident software on which the operating system relies. It is responsible for most operating system functions. It schedules and manages the work done by the computer and maintains the file system. The kernel has its own text, data, and stack areas.

lexical analysis

Lexical analysis is the process by which a stream of characters (often comprising a source program) is subdivided into its elementary words and symbols (called tokens). The tokens include the reserved words of the language, its identifiers and constants, and special symbols such as =, :=, and ;. Lexical analysis enables you to recognize, for example, that the stream of characters 'print("hello, universe")' is to be analyzed into a series of tokens beginning with the word 'print' and not with, say, the string 'print("h.' In compilers, a lexical analyzer is often called by the compiler's syntactic analyzer or parser, which determines the statements of the program (that is, the proper arrangements of its tokens).

library

A library is an archive file that contains object code and/or files for programs that perform common tasks. The library provides a common source for object code, thus saving space by providing one copy of the code instead of

requiring every program that wants to incorporate the functions in the code to have its own copy. The link editor may select functions and data as needed.

link editor

A link editor, or loader, collects and merges separately compiled object files by linking together object files and the libraries that are referenced into executable load modules. The result is an **a.out** file. Link editing may be done automatically when you use the compilation system to process your programs, but you can also link edit previously compiled files by using the **ld(1)** command.

Local management

The phase in either connection-mode or connectionless-mode in which a transport user establishes a transport endpoint and binds a transport address to the endpoint. Functions in this phase perform local operations, and require no transport layer traffic over the network.

magic number

The magic number is contained in the header of an **a.out** file. It indicates what the type of the file is, whether shared or non-shared text, and on which processor the file is executable.

makefile

A makefile is a file that lists dependencies among the source code files of a software product and methods for updating them, usually by recompilation. The **make(1)** command uses the makefile to maintain self-consistent software.

manual page

A manual page, or "man page", is the repository for the detailed description of a command, a system call, subroutine or other operating system component.

null pointer

A null pointer is a C pointer with a value of 0.

object code

Object code is executable machine-language code produced from source code or from other object files by an assembler or a compilation system. An object file is a file of object code and associated data. An object file that is ready to run is an executable file.

operating system

An operating system is the software on the computer under which all other software runs. **SYSTEM V/68** is a general-purpose, multiuser, interactive time-sharing operating system with two basic parts:

GLOSSARY

- The kernel is the program that is responsible for most operating system functions. It schedules and manages all the work done by the computer and maintains the file system. It is always running and is invisible to users.
- The shell is the program responsible for handling all interaction between users and the computer. It includes a powerful command language called shell language.

The utility programs or operating system commands are executed using the shell. They allow users to communicate with each other, edit and manipulate files, and write and execute programs in several programming languages.

optimizer

An optimizer, an optional step in the compilation process, improves the efficiency of the assembly language code. The optimizer reduces the space used by and speeds the execution time of the code.

option

An option is an argument used in a command line to modify program output by modifying the execution of a command. An option is usually one character preceded by a hyphen (-). When you do not specify any options, the command will execute according to its default options. For example, in the command line

ls -a -l directory

-a and **-l** are the options that modify the **ls(1)** command to list all **directory** entries, including entries whose names begin with a period (.), in the long format (including permissions, size, and date).

Orderly release

A procedure for gracefully terminating a transport connection with no loss of data.

parent process

A parent process occurs when a process is split into two, a parent process and a child process, with separate, but initially identical text, data, and stack segments.

parse	To parse is to analyze a sentence in order identify its components and to determine their grammatical relationship. In computer terminology the word has a similar meaning, but instead of sentences, program statements or commands are analyzed.
PASCAL	PASCAL is a multipurpose high-level programming language often used to teach programming. It is based on the ALGOL programming language and emphasizes structured programming.
path name	<p>A path name is a way of designating the exact location of a file in a file system. It is made up of a series of directory names that proceed down the hierarchical path of the file system. The directory names are separated by a slash character (/). The last name in the path is either a file or another directory. If the path name begins with a slash, it is called a full path name; the initial slash means that the path begins at the root directory.</p> <p>A path name that does not begin with a slash is known as a relative path name, meaning relative to the present working directory. A relative path name may begin either with a directory name or with two dots followed by a slash (../). One that begins with a directory name indicates that the ultimate file or directory is below the present working directory in the hierarchy. One that begins with ../ indicates that the path first proceeds up the hierarchy; ../ is the parent of the present working directory.</p>
Peer user	The user with whom a given user is communicating above the Transport Interface.
permissions	<p>Permissions are a means of defining a right to access a file or directory in the file system. Permissions are granted separately to you, the owner of the file or directory, your group, and all others. There are three basic permissions:</p> <ul style="list-style-type: none"> □ Read permission (r) includes permission to cat, pg, lp, and cp a file.

GLOSSARY

- Write permission (w) is the permission to change a file.
- Execute permission (x) is the permission to run an executable file.

Permissions can be changed with the operating system **chmod(1)** command.

pipe

A pipe causes the output of one command to be used as the input for the next command so that the two run in sequence. You can do this by preceding each command after the first command with the pipe symbol (`|`), which indicates that the output from the process on the left should be routed to the process on the right. For example, in the command

```
who | wc -l,
```

the output from the **who(1)** command, which lists the users who are logged on to the system, is used as input for the word-count command, **wc(1)**, with the `l` option. The result of this pipeline (succession of commands connected by pipes) is the number of people who are currently logged on to the system.

portable

Portability describes the degree of ease with which a program or a library can be moved or ported from one system to another. Portability is desirable because once a program is developed it is used on many systems. If the program writer must change the program in many different ways before it can be distributed to the other systems, time is wasted, and each modification increases the chances for an error.

preprocessor

Preprocessor is a generic name for a program that prepares an input file for another program. For example, **neqn(1)** and **tbl(1)** are preprocessors for **nroff(1)**. **grap(1)** is a preprocessor for **pic(1)**. **cpp(1)** is a preprocessor for the C compiler.

process

A process is a program that is at some stage of execution. It also refers to the execution of a computer environment, including contents of memory, register values, name of the current working directory, status of files, information recorded at login time, etc. Every time you type the name

of a file that contains an executable program, you initiate a new process. Shell programs can cause the initiation of many processes because they can contain many command lines.

The process id is a unique system-wide identification number that identifies an active process. The process status command, **ps(1)**, prints the process ids of the processes that belong to you.

program	A program is a sequence of instructions or commands that cause the computer to perform a specific task, for example, changing text, making a calculation, or reporting on the status of the system. A subprogram is part of a larger program and can be compiled independently.
regular expression	A regular expression is a string of alphanumeric characters and special characters that describe a character string. It is a shorthand way of describing a pattern to be searched for in a file. The pattern-matching functions of ed(1) and grep(1) , for example, use regular expressions.
routine	A routine is a discrete section of a program to accomplish a set of related tasks
semaphore	A semaphore is a sharable short unsigned integer maintained through a family of system calls which include calls for increasing the value of the semaphore, setting its value, and for blocking waiting for its value to reach some value. Semaphores are part of the operating system IPC facility.
Server	The transport user in connection-mode that offers services to other users (clients) and enables these clients to establish a transport connection to it.
Service indication	The notification of a pending event generated by the provider to a user of a particular service.
Service primitive	The unit of information passed across a service interface that contains either a service request or service indication.

GLOSSARY

- Service request A request for some action generated by a user to the provider of a particular service.
- shared library Shared libraries include object modules that may be shared among several processes at execution time.
- shared memory Shared memory is an IPC (interprocess communication) facility in which two or more processes can share the same data space.
- shell The shell is the operating system program—**sh(1)**—responsible for handling all interaction between you and the system. It is a command language interpreter that understands your commands and causes the computer to act on them. The shell also establishes the environment at your terminal. A shell normally is started for you as part of the login process. Three shells, the Bourne shell, the Korn shell and the C shell, are popular. The shell can also be used as a programming language to write procedures for a variety of tasks.
- signal: signal number A signal is a message that you send to processes or processes send to one another. The most common signals you might send to a process are ones that would cause the process to stop: for example, interrupt, quit, or kill. A signal sent by a running process is usually a sign of an exceptional occurrence that has caused the process to terminate or divert from the normal flow of control.
- source code Source code is the programming-language version of a program. Before the computer can execute the program, the source code must be translated to machine language by a compilation system or an interpreter.
- standard error Standard error is an output stream from a program. It is normally used to convey error messages. The default case is to associate standard error with the user's terminal.
- standard input Standard input is an input stream to a program. The default case is to associate standard input with the user's terminal.

standard output	Standard output is an output stream from a program. The default case is to associate standard output with the user's terminal.
stdio: standard input-output	stdio(3S) is a collection of functions for formatted and character-by-character input-output at a higher level than the basic read, write, and open operations.
static linking	Static linking refers to the requirement that symbolic references be resolved before run time. See dynamic linking.
stream	<ul style="list-style-type: none">□ A stream is an open file with buffering provided by the stdio package.□ A stream is a full duplex, processing and data transfer path in the kernel. It implements a connection between a driver in kernel space and a process in user space, providing a general character input/output interface for the user processes.
string	A string is a contiguous sequence of characters treated as a unit. Strings are normally bounded by white space(s), tab(s), or a character designated as a separator. A string value is a specified group of characters symbolized to the shell by a variable.
strip	strip(1) is a command that removes the symbol table and relocation bits from an executable file.
subroutine	A subroutine is a program that defines desired operations and may be used in another program to produce the desired operations. A subroutine can be arranged so that control may be transferred to it from a master routine and so that, at the conclusion of the subroutine, control reverts to the master routine. Such a subroutine is usually called a closed subroutine. A single routine may be simultaneously a subroutine with respect to another routine and a master routine with respect to a third.
symbol table	A symbol table describes information in an object file about the names and functions in that file. The symbol table and relocation bits are used by the link editor and by the debuggers.

GLOSSARY

symbol value The value of a symbol, typically its virtual address, used to resolve references.

Synchronous execution

The mode of execution in which Transport Interface routines may block while waiting for specific asynchronous events to occur.

syntax

- Command syntax is the order in which command names, options, option arguments, and operands are put together to form a command on the command line. The command name is first, followed by options and operands. The order of the options and the operands varies from command to command.
- Language syntax is the set of rules that describe how the elements of a programming language may legally be used.

system call

A system call is a request by an active process for a service performed by the operating system kernel, such as I/O, process creation, etc. All system operations are allocated, initiated, monitored, manipulated, and terminated through system calls. System calls allow you to request the operating system to do some work that the program would not normally be able to do. For example, the **getuid(2)** system call allows you to inspect information that is not normally available since it resides in the operating system's address space.

target machine

A target machine is the machine on which an **a.out** file is run. While it may be the same machine on which the **a.out** file was produced, the term implies that it may be a different machine.

TCP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols that support multi-network applications. It is the Department of Defense standard in packet networks.

terminal definition

A terminal definition is an entry in the **terminfo(4)** data base that describes the characteristics of a terminal. See **terminfo(4)** and **curses(3X)** in the *Programmer's Reference Manual*.

- terminfo**
- a group of routines within the `curses` library that handle certain terminal capabilities. For example, if your terminal has programmable function keys, you can use these routines to program the keys.
 - a data base containing the compiled descriptions of many terminals that can be used with `curses(3X)` screen management programs. These descriptions specify the capabilities of a terminal and how it performs various operations — for example, how many lines and columns it has and how its control characters are interpreted. A `curses(3X)` program refers to the data base at run time to obtain the information that it needs about the terminal being used.
- See `curses(3X)` in the *Programmer's Reference Manual*. `terminfo(4)` routines can be used in shell programs, as well as C programs.
- text symbol** A text symbol is a symbol, usually a function name, that is defined in the `.text` portion of an `a.out` file.
- tool** A tool is a program, or package of programs, that performs a given task.
- Transport address** The identifier used to differentiate and locate specific transport endpoints in a network.
- Transport connection** The communication circuit that is established between two transport users in connection-mode.
- Transport endpoint** The local communication channel between a transport user and a transport provider.
- Transport Interface** The library routines and state transition rules that support the services of a transport protocol.
- Transport provider** The transport protocol that provides the services of the Transport Interface.

GLOSSARY

Transport service data unit	The amount of user data whose identity is preserved from one end of a transport connection to the other (that is, a message).
Transport user	The user-level application or protocol that accesses the services of the Transport Interface.
trap	A trap is a condition caused by an error where a process state transition occurs and a signal is sent to the currently running process.
userid	A userid is an integer value, usually associated with a login name, used by the system to identify owners of files and directories. The userid of a process becomes the owner of files created by the process and descendent (forked) processes.
utility	A utility is a standard, permanently available program used to perform routine functions or to assist a programmer in the diagnosis of hardware and software errors, for example, a loader, editor, debugging, or diagnostics package.
Virtual circuit	A transport connection established in connection-mode.
variable	<ul style="list-style-type: none">□ A variable in a computer program is an object whose value may change during the execution of the program, or from one execution to the next.□ A variable in the shell is a name representing a string of characters (a string value).□ A variable normally set only on a command line is called a parameter (positional parameter and keyword parameter).□ A variable may be simply a name to which the user (user-defined variable) or the shell itself may assign string values.
white space	White space is one or more spaces, tabs, or newline characters. White space is normally used to separate strings of characters, and is required to separate the command from its arguments on a command line.

word A word is a unit of storage in a computer that is composed of bytes of information. The number of bytes in a word depends on the computer you are using. Computers based on the Motorola M68000 family of microprocessors, for example, have 16 bits or 2 bytes per word and 32 bits or 4 bytes per long word.

The following acronyms are used throughout this guide:

- CLTS** Connectionless Transport Service
- COTS** Connection Oriented Transport Service
- ETSDU** Expedited Transport Service Data Unit
- TSDU** Transport Service Data Unit

**G
L
O
S
S
A
R
Y**