Glossary

8087, 80287, or 80387 coprocessor Intel chips that perform high-speed floating-point and binary coded decimal number processing. Also called math coprocessors. Floating-point instructions are supported directly by the 80486 processor.

A

address The memory location of a data item or procedure. The expression can represent just the offset (in which case the default segment is assumed), or it can be in *segment***:***offset* format.

address constant In an assembly-language instruction, an immediate operand derived by applying the **SEG** or **OFFSET** operator to an identifier.

address range A range of memory bounded by two addresses.

addressing modes The various ways a memory address or device I/O address can be generated. See "far address," "near address."

aggregate types Data types containing more than one element, such as arrays, structures, and unions.

animate A debugging feature in which each line in a running program is highlighted as it executes. The Animate command from the CodeView debugger Run menu turns on animation.

API (application programming interface) A set of system-level routines that can be used in an application program for tasks such as basic input/output and file management. In a graphics-oriented operating environment like Microsoft Windows, high-level support for video graphics output is part of the Windows graphical API.

arg In PWB, a function modifier that introduces an argument or an editing function. The argument may be of any type and is passed to the next function as input. For example, the PWB command `Arg textarg Copy` passes the text argument `textarg` to the function `Copy`.

argument A value passed to a procedure or function. See "parameter."

array An ordered set of continuous elements of the same type.

ASCII (American Standard Code for Information Interchange) A widely used coding scheme where 1-byte numeric values represent letters, numbers, symbols, and special characters. There are 256 possible codes. The first 128 codes are standardized; the remaining 128 are special characters defined by the computer manufacturer.

assembler A program that converts a text file containing mnemonically coded microprocessor instructions into the corresponding binary machine code. MASM is an assembler. See "compiler."

assembly language A programming language in which each line of source code corresponds to a specific microprocessor instruction. Assembly language gives the programmer full access to the computer's hardware and produces the most compact, fastest executing code. See "high-level language."

assembly mode The mode in which the CodeView debugger displays the assembly-language equivalent of the high-level code being executed. CodeView obtains the assembly-language code by disassembling the executable file. See “source mode.”

B

base address The starting address of a stack frame. Base addresses are usually stored in the BP register.

base name The portion of the filename that precedes the extension. For example, SAMPLE is the base name of the file SAMPLE.ASM.

BCD (binary coded decimal) A way of representing decimal digits where 4 bits of 1 byte are a decimal digit, coded as the equivalent binary number.

binary Referring to the base-2 counting system, whose digits are 0 and 1.

binary expression A Boolean expression consisting of two operands joined by a binary operator and resolving to a binary number.

binary file A file that contains numbers in binary form (as opposed to ASCII characters representing the same numbers). For example, a program file is a binary file.

binary operator A Boolean operator that takes two arguments. The **AND** and **OR** operators in assembly language are examples of binary operators.

BIOS (Basic Input/Output System) The software in a computer’s ROM which forms a hardware-independent interface between the CPU and its peripherals (for example, keyboard, disk drives, video display, I/O ports).

bit Short for **bi**nary digi**t**. The basic unit of binary counting. Logically equivalent to decimal digits, except that bits can have a value of 0 or 1, whereas decimal digits can range from 0 through 9.

breakpoint A user-defined condition that pauses program execution while debugging. CodeView can set breakpoints at a specific line of code, for a specific value of a variable, or for a combination of these two conditions.

buffer A reserved section of memory that holds data temporarily, most often during input/output operations.

byte The smallest unit of measure for computer memory and data storage. One byte consists of 8 bits and can store one 8-bit character (a letter, number, punctuation mark, or other symbol). It can represent unsigned values from 0 to 255 or signed values between –128 and +127.

C

C calling convention The convention that follows the C standard for calling a procedure—that is, pushing arguments onto the stack from right to left (in reverse order from the way they appear in the argument list). The C calling convention permits a variable number of arguments to be passed.

chaining (to an interrupt) Installing an interrupt handler that shares control of an interrupt with other handlers. Control passes from one handler to the next until a handler breaks the chain by terminating through an **IRET** instruction. See “interrupt handler,” “hooking (an interrupt).”

character string See “string.”

clipboard In PWB, a section of memory that holds text deleted with the Copy, Ldelete, or Sdelete functions. Any text attached to the clipboard deletes text already there. The Paste function inserts text from the clipboard at the current cursor position.

.COM The filename extension for executable files that have a single segment containing both code and data. Tiny model produces .COM files.

combine type The segment-declaration specifier (**AT**, **COMMON**, **MEMORY**, **PUBLIC**, or **STACK**) which tells the linker to combine all segments of the same type. Segments without a combine type are private and are placed in separate physical segments.

compact A memory model with multiple data segments but only one code segment.

compiler A program that translates source code into machine language. Usually applied only to high-level languages such as Basic, FORTRAN, or C. See “assembler.”

constant A value that does not change during program execution. A variable, on the other hand, is a value that can—and usually does—change. See “symbolic constant.”

constant expression Any expression that evaluates to a constant. It may include integer constants, character constants, floating-point constants, or other constant expressions.

D

debugger A utility program that allows the programmer to execute a program one line at a time and view the contents of registers and memory in order to help locate the source of bugs or other problems. Examples are CodeView and Symdeb.

declaration A construct that associates the name and the attributes of a variable, function, or type. See “variable declaration.”

default A setting or value that is assumed unless specified otherwise.

definition A construct that initializes and allocates storage for a variable, or that specifies either code labels or the name, formal parameters, body, and return type of a procedure. See “type definition.”

description file A text file used as input for the NMAKE utility.

device driver A program that transforms I/O requests into the operations necessary to make a specific piece of hardware fulfill that request.

Dialog Command window The window at the bottom of the CodeView screen where dialog commands can be entered, and previously entered dialog commands can be reviewed.

direct memory operand In an assembly-language instruction, a memory operand that refers to the contents of an explicitly specified memory location.

directive An instruction that controls the assembler's state.

displacement In an assembly-language instruction, a constant value added to an effective address. This value often specifies the starting address of a variable, such as an array or multidimensional table.

DLL See "dynamic-link library."

double-click To rapidly press and release a mouse button twice while pointing the mouse cursor at an object on the screen.

double precision A real (floating-point) value that occupies 8 bytes of memory (MASM type **REAL8**). Double-precision values are accurate to 15 or 16 digits.

doubleword A 4-byte word (MASM type **DWORD**).

drag To move the mouse while pointing at an object and holding down one of the mouse buttons.

dump To display or print the contents of memory in a specified memory range.

dynamic linking The resolution of external references at load time or run time (rather than link time). Dynamic linking allows the called subroutines to be packaged, distributed, and maintained independently of their callers. Windows extends the dynamic-link mechanism to serve as the primary method by which all system and nonsystem services are obtained. See "linking."

dynamic-link library (DLL) A library file that contains the executable code for a group of dynamically linked routines.

dynamic-link routine A routine in a dynamic-link library that can be linked at load time or run time.

E

element A single member variable of an array of like variables.

environment block The section of memory containing the MS-DOS environment variables.

errorlevel code See "exit code."

.EXE The filename extension for a program that can be loaded and executed by the computer. The small, compact, medium, large, huge, and flat models generate .EXE files. See ".COM," "tiny."

exit code A code returned by a program to the operating system. This usually indicates whether the program ran successfully.

expanded memory Increased memory available after adding an EMS (Expanded Memory Specification) board to an 8086 or 80286 machine. Expanded memory can be simulated in software. The EMS board can increase memory from 1 megabyte to 8 megabytes by swapping segments of high-end memory into lower memory. Applications must be written to the EMS standard in order to make use of expanded memory. See "extended memory."

expression Any valid combination of

mathematical or logical variables, constants, strings, and operators that yields a single value.

extended memory Physical memory above 1 megabyte that can be addressed by 80286–80486 machines in protected mode. Adding a memory card adds extended memory. On 80386-based machines, extended memory can be made to simulate expanded memory by using a memory-management program.

extension The part of a filename (of up to three characters) that follows the period (**.**). An extension is not required but is usually added to differentiate similar files. For example, the source-code file MYPROG.ASM is assembled into the object file MYPROG.OBJ, which is linked to produce the executable file MYPROG.EXE.

external variable A variable declared in one module and referenced in another module.

F

far address A memory location specified with a segment value plus an offset from the start of that segment. Far addresses require 4 bytes—two for the segment and two for the offset. See “near address.”

field One of the components of a structure, union, or record variable.

fixup The linking process that supplies addresses for procedure calls and variable references.

flags register A register containing information about the status of the CPU and the results of the last arithmetic operation performed by the CPU.

flat A nonsegmented linear address space. Selectors in flat model can address the entire 4 gigabytes of addressable memory space. See “segment,” “selector.”

formal parameters The variables that receive values passed to a function when the function is called.

forward declaration A function declaration that establishes the attributes of a symbol so that it can be referenced before it is defined, or called from a different source file.

frame The segment, group, or segment register that specifies the segment portion of an address.

G

General-Protection (GP) fault An error that occurs in protected mode when a program accesses invalid memory locations or accesses valid locations in an invalid way (such as writing into ROM areas).

gigabyte 1,024 megabytes, or 1,073,741,824 bytes.

global See “visibility.”

global constant A constant available throughout a module. Symbolic constants defined in the module-level code are global constants.

global data segment A data segment that is shared among all instances of a dynamic-link routine; in other words, a single segment that is accessible to all processes that call a particular dynamic-link routine.

global variable A variable that is available (visible) across multiple modules.

granularity The degree to which library procedures can be linked as individual blocks of code. In Microsoft libraries, granularity is at the object-file level. If a single object file containing three procedures is added to a library, all three procedures will be linked with the main program even if only one of them is actually called.

group A collection of individually defined segments that have the same segment base address.

H

handle An arbitrary value that an operating system supplies to a program (or vice versa) so that the program can access system resources, files, peripherals, and so forth, in a controlled fashion.

handler See “interrupt handler.”

hexadecimal The base-16 numbering system whose digits are 0 through F (the letters A through F represent the decimal numbers 10 through 15). This is often used in computer programming because it is easily converted to and from the binary (base-2) numbering system the computer itself uses.

high-level language A programming language that expresses operations as mathematical or logical relationships, which the language’s compiler then converts into machine code. This contrasts with assembly language, in which the program is written directly as a sequence of explicit microprocessor instructions. Basic, C, COBOL, and FORTRAN are examples of high-level languages. See “assembly language,” “compiler.”

hooking (an interrupt) Replacing an address in the interrupt vector table with the address of another interrupt handler. See “interrupt handler,” “interrupt vector table,” **“**unhooking (an interrupt).”

huge A memory model (similar to large model) with more than one code segment and more than one data segment. However, individual data items can be larger than 64K, spanning more than one segment. See “large.”

I

identifier A name that identifies a register or memory location.

IEEE format A standard created by the Institute of Electrical and Electronics Engineers for representing floating-point numbers, performing math with them, and handling underflow/overflow conditions. The 8087 family of coprocessors and the emulator package implement this format.

immediate expression An expression that evaluates to a number that can be either a component of an address or the entire address.

immediate operand In an assembly-language instruction, a constant operand that is specified at assembly time and stored in the program file as part of the instruction opcode.

import library A pseudo library that contains addresses rather than executable code. The linker reads the addresses from an import library to resolve references to external dynamic-link library routines.

include file A text file with the .INC extension whose contents are inserted into the source-code file and immediately assembled.

indirect memory operand In an assembly-language instruction, a memory operand whose value is treated as an address that points to the location of the desired data. See “pointer.”

instruction The unit of binary information that a CPU decodes and executes. In assembly language, instruction refers to the mnemonic (such as **LDS** or **SHL**) that the assembler converts into machine code.

instruction prefix See “prefix.”

interrupt A signal to the processor to halt its current operation and immediately transfer control to an interrupt handler. Interrupts are triggered either by hardware, as when the keyboard detects a keypress, or by software, as when a program executes the **INT** instruction. See “interrupt handler.”

interrupt handler A routine that receives processor control when a specific interrupt occurs.

interrupt service routine See “interrupt handler.”

interrupt vector An address that points to an interrupt handler.

interrupt vector table A table maintained by the operating system. It contains addresses (vectors) of current interrupt handlers. When an interrupt occurs, the CPU branches to the address in the table that corresponds to the interrupt’s number. See “interrupt handler.”

K

keyword A word with a special, predefined meaning for the assembler. Keywords cannot be used as identifiers.

kilobyte (K) 1,024 bytes.

L

label A symbol (identifier) representing the address of a code label or data objects.

language type The specifier that establishes the naming and calling conventions for a procedure. These are **BASIC**, **C**, **FORTRAN**, **PASCAL**, **STDCALL**, and **SYSCALL**.

large A memory model with more than one code segment and more than one data segment, but with no individual data item larger than 64K (a single segment). See “huge.”

library A file that contains modules of compiled code. MS-DOS programs use normal run-time libraries, from which the linker extracts modules and combines them with other object modules to create executable program files. Windows-based programs can use dynamic-link libraries (see), which the operating system loads and links to calling programs. See also “import library.”

linked list A data structure in which each entry includes a pointer to the location of the adjoining entries.

linking In normal static linking, the process in which the linker resolves all external references by searching run-time and user libraries, and then computes absolute offset addresses for these references. Static linking results in a single executable file. In dynamic linking (see), the operating system, rather than the linker, provides the addresses after loading the modules into separate parts of memory.

local constant A constant whose scope is limited to a procedure or a module.

local variable A variable whose scope is confined to a particular unit of code, such as module-level code, or a procedure. See “module-level code.”

logical device A symbolic name for a device that can be mapped to a physical (actual) device.

logical line A complete program statement in source code, including the initial line of code and any extension lines.

logical segment A memory area in which a program stores code, data, or stack information. See “physical segment.”

low-level input and output routines Run-time library routines that perform unbuffered, unformatted input/output operations.

LSB (least-significant bit) The bit lowest in memory in a binary number.

M

machine code The binary numbers that a microprocessor interprets as program instructions. See “instruction.”

macro A block of text or instructions that has been assigned an identifier. When the assembler sees this identifier in the source code, it substitutes the related text or instructions and assembles them.

main module The module containing the point where program execution begins (the program’s entry point). See “module.”

math coprocessor See “8087, 80287, or 80387 coprocessor.”

medium A memory model with multiple code segments but only one data segment.

megabyte 1,024 kilobytes or 1,048,576 bytes.

member One of the elements of a structure or union; also called a field.

memory address A number through which a program can reference a location in memory.

memory map A representation of where in memory the computer expects to find certain types of information.

memory model A convention for specifying the number and types of code and data segments in a module. See “tiny,” “small,” “medium,” “compact,” “large,” “huge,” and “flat.”

memory operand An operand that specifies a memory location.

meta A prefix that modifies the subsequent PWB function.

mnemonic A word, abbreviation, or acronym that replaces something too complex to remember or type easily. For example, **ADC** is the mnemonic for the 8086’s add-with-carry instruction. The assembler converts it into machine (binary) code, so it is not necessary to remember or calculate the binary form.

module A discrete group of statements. Every program has at least one module (the main module). In most cases, a module is the same as a source file.

module-definition file A text file containing information that the linker uses to create a Windows-based program.

module-level code Program statements within any module that are outside procedure definitions.

MSB (most-significant bit) The bit farthest to the left in a binary number. It represents $2^{(n-1)}$, *where n* is the number of bits in the number.

multitasking operating system An operating system in which two or more programs, processes, or threads can execute simultaneously.

N

naming convention The way the compiler or assembler alters the name of a routine before placing it into an object file.

NAN Acronym for “not a number.” Math coprocessors generate NANs when the result of an operation cannot be represented in IEEE format. For example, if two numbers being multiplied have a product larger than the maximum value permitted, the coprocessor returns a NAN instead of the product.

near address A memory location specified by the offset from the start of the value in a segment register. A near address requires only 2 bytes. See “far address.”

nonreentrant See “reentrant procedure.”

null character The ASCII character encoded as the value 0.

null pointer A pointer to nothing, expressed as the value 0.

O

.OBJ Default filename extension for an object file.

object file A file (normally with the extension .OBJ) produced by assembling source code. It contains relocatable machine code. The linker combines object files with run-time and library code to create an executable file.

offset The number of bytes from the beginning of a segment to a particular byte within that segment.

opcode The binary number that represents a specific microprocessor instruction.

operand A constant or variable value that is manipulated in an expression or instruction.

operator One or more symbols that specify how the operand or operands of an expression are manipulated.

option A variable that modifies the way a program performs. Options can appear on the command line, or they can be part of an initialization file (such as TOOLS.INI). An option is sometimes called a switch.

output screen The CodeView screen that displays program output. Choosing the Output command from the View menu or pressing F4 switches to this screen.

overflow An error that occurs when the value assigned to a numeric variable is larger than the allowable limit for that variable’s type.

overlay A program component loaded into memory from disk only when needed. This technique reduces the amount of free RAM needed to run the program.

P

parameter The name given in a procedure definition to a variable that is passed to the procedure. See “argument.”

passing by reference Transferring the address of an argument to a procedure. This allows the procedure to modify the argument’s value.

passing by value Transferring the value (rather than the address) of an argument to a procedure. This prevents the procedure from changing the argument’s original value.

physical segment The true memory address of a segment, referenced through a segment register.

pointer A variable containing the address of another variable. See “indirect memory operand.”

precedence The relative position of an operator in the hierarchy that determines the order in which expression elements are evaluated.

preemptive Having the power to take precedence over another event.

prefix A keyword (**LOCK**, **REP**, **REPE**, **REPNE**, **REPNZ**, or **REPZ**) that modifies the behavior of an instruction. MASM 6.1 ensures the prefix is compatible with the instruction.

private Data items and routines local to the module in which they are defined. They cannot be accessed outside that module. See “public.”

privilege level A hardware-supported feature of the 80286–80486 processors that allows the programmer to specify the exclusivity of a program or process. Programs running at low-numbered privilege levels can access data or resources at higher-numbered privilege levels, but the reverse is not true. This feature reduces the possibility that malfunctioning code will corrupt data or crash the operating system.

privileged mode The term applied to privilege level 0. This privilege level should be used only by a protected-mode operating system. Special privileged instructions are enabled by .**286P**, .**386P**, and .**486P**. Privileged mode should not be confused with protected mode.

procedure call An expression that invokes a procedure and passes actual arguments (if any) to the procedure.

procedure definition A definition that specifies a procedure's name, its formal parameters, the declarations and statements that define what it does, and (optionally) its return type and storage class.

procedure prototype A procedure declaration that includes a list of the names and types of formal parameters following the procedure name.

process Generally, any executing program or code unit. This term implies that the program or unit is one of a group of processes executing independently.

Program Segment Prefix (PSP) A 256-byte data structure at the base of the memory block allocated to a transient program. It contains data and addresses supplied by MS-DOS that a program can read during execution.

protected mode The 80286–80486 operating mode that permits multiple processes to run and not interfere with each other. This feature should not be confused with privileged mode.

public Data items and procedures that can be accessed outside the module in which they are defined. See "private."

Q

qualifiedtype A user-defined type consisting of an existing MASM type (intrinsic, structure, union, or record), or a previously defined **TYPEDEF** type, together with its language or distance attributes.

R

radix The base of a number system. The default radix for MASM and CodeView is 10.

RAM (random-access memory) Computer memory that can be both written to and read from. RAM data is volatile; it is usually lost when the computer is turned off. Programs are loaded into and executed from RAM. See "ROM."

real mode The normal operating mode of the 8086 family of processors. Addresses correspond to physical (not mapped) memory locations, and there is no mechanism to keep one application from accessing or modifying the code or data of another. See "protected mode."

record A MASM variable that consists of a sequence of bit values.

reentrant procedure A procedure that can be safely interrupted during execution and restarted from its beginning in response to a call from a preemptive process. After servicing the preemptive call, the procedure continues execution at the point at which it was interrupted.

register operand In an assembly-language instruction, an operand that is stored in the register specified by the instruction.

register window The optional CodeView window in which the CPU registers and the flag register bits are displayed.

registers Memory locations in the processor that temporarily store data, addresses, and processor flags.

regular expression A text expression that specifies a pattern of text to be matched (as opposed to matching specific characters).

relocatable Not having an absolute address. The assembler does not know where the label, data, or code will be located in memory, so it generates a fixup record. The linker provides the address.

return value The value returned by a function.

ROM (read-only memory) Computer memory that can only be read from and cannot be modified. ROM data is permanent; it is not lost when the machine is turned off. A computer's ROM often contains BIOS routines and parts of the operating system. See "RAM."

routine A generic term for a procedure or function.

run-time dynamic linking The act of establishing a link when a process is running. See "dynamic linking."

run-time error A math or logic error that can be detected only when the program runs. Examples of run-time errors are dividing by a variable whose value is zero or calling a DLL function that doesn't exist.

S

scope The range of statements over which a variable or constant can be referenced by name. See "global constant," "global variable," "local constant," "local variable."

screen swapping A screen-exchange method that uses buffers to store the debugging and output screens. When you request the other screen, the two buffers are exchanged. This method is slower than flipping (the other screen-exchange method), but it works with most adapters and most types of programs.

scroll bars The bars that appear at the right side and bottom of a window and some list boxes. Dragging the mouse on the scroll bars allows scrolling through the contents of a window or text box.

segment A section of memory, limited to 64K with 16-bit segments or 4 gigabytes with 32-bit segments, containing code or data. Also refers to the starting address of that memory area.

sequential mode The mode in CodeView in which no windows are available. Input and output scroll down the screen, and the old output scrolls off the top of the screen when the screen is full. You cannot examine previous commands after they scroll off the top. This mode is required with computers that are not IBM compatible.

selector A value that indirectly references a segment address. A protected-mode operating system, such as Windows, assigns selector values to programs, which use them as segment addresses. If a program attempts to use an unassigned selector, it triggers a General-Protection fault (see).

shared memory A memory segment that can be accessed simultaneously by more than one process.

shell escape A method of gaining access to the operating system without leaving CodeView or losing the current debugging context. It is possible to execute MS-DOS commands, then return to the debugger.

sign extended The process of widening an integer (for example, going from a byte to a word, or a word to a doubleword) while retaining its correct value and sign.

signed integer An integer value that uses the most-significant bit to represent the value's sign. If the bit is one, the number is negative; if zero, the number is positive. See "two's complement," "unsigned integer," "MSB."

single precision A real (floating-point) value that occupies 4 bytes of memory. Single-precision values are accurate to six or seven decimal places.

single-tasking environment An environment in which only one program runs at a time. MS-DOS is a single-tasking environment.

small A memory model with only one code segment and only one data segment.

source file A text file containing symbols that define the program.

source mode The mode in which CodeView displays the assembly-language source code that represents the machine code currently being executed.

stack An area of memory in which data items are consecutively stored and removed on a last-in, first-out basis. A stack can be used to pass parameters to procedures.

stack frame The portion of a stack containing a particular procedure’s local variables and parameters.

stack probe A short routine called on entry to a function to verify that there is enough room in the program stack to allocate local variables required by the function.

stack switching Changing the stack pointers to point to another stack area.

stack trace A symbolic representation of the functions that are being executed to reach the current instruction address. As a function is executed, the function address and any function arguments are pushed on the stack. Therefore, tracing the stack shows the active functions and their arguments.

standard error The device to which a program can send error messages. The display is normally standard error.

standard input The device from which a program reads its input. The keyboard is normally standard input.

standard output The device to which a program can send its output. The display is normally standard output.

statement A combination of labels, data declarations, directives, or instructions that the assembler can convert into machine code.

status bar See “linking.”

static linking The line at the bottom of the PWB or CodeView screen. The status bar displays text position, keyboard status, current context of execution, and other program information.

STDCALL A calling convention that uses caller stack cleanup if the **VARARG** keyword is specified. Otherwise the called routine must clean up the stack.

string A contiguous sequence of characters identified with a symbolic name.

string literal A string of characters and escape sequences delimited by single quotation marks (`' '`) or double quotation marks (`" "`).

structure A set of variables that may be of different types, grouped under a single name.

structure member One of the elements of a structure. Also called a field.

switch See "option."

symbol A name that identifies a memory location (usually for data).

symbolic constant A constant represented by a symbol rather than the constant itself. Symbolic constants are defined with **EQU** statements. They make a program easier to read and modify.

SYSCALL A language type for a procedure. Its conventions are identical to C's, except no underscore is prefixed to the name.

T

tag The name assigned to a structure, union, or enumeration type.

task See "process."

text Ordinary, readable characters, including the uppercase and lowercase letters of the alphabet, the numerals 0 through 9, and punctuation marks.

text box In PWB, a box where you type information needed to carry out a command. A text box appears within a dialog box. The text box may be blank or contain a default entry.

tiny Memory model with a single segment for both code and data. This limits the total program size to 64K. Tiny programs have the filename extension .COM.

toggle A function key or menu selection that turns a feature off if it is on, or on if it is off. Used as a verb, "toggle" means to reverse the status of a feature.

TOOLS.INI A file containing initialization information for many of the Microsoft utilities, including PWB.

two's complement A form of base-2 notation in which negative numbers are formed by inverting the bit values of the equivalent positive number and adding 1 to the result.

type A description of a set of values and a valid set of operations on items of that type. For example, a variable of type **BYTE** can have any of a set of integer values within the range specified for the type on a particular machine.

type checking An operation in which the assembler verifies that the operands of an operator are valid or that the actual arguments in a function call are of the same types as the function definition's parameters.

type definition The storage format and attributes for a data unit.

U

unary expression An expression consisting of a single operand preceded or followed by a unary operator.

unary operator An operator that acts on a single operand, such as **NOT**.

underflow An error condition that occurs when a calculation produces a result too small for the computer to represent.

unhooking (an interrupt) The act of removing your interrupt handler and restoring the original vector. See “hooking (an interrupt).”

union A set of values (in fields) of different types that occupy the same storage space.

unresolved external See “unresolved reference.”

unresolved reference A reference to a global or external variable or function that cannot be found, either in the modules being linked or in the libraries linked with those modules. An unresolved reference causes a fatal link error.

unsigned integer An integer in which the most-significant bit serves as part of the number, rather than as an indication of sign. For example, an unsigned byte integer can have a value from 0 to 255. A signed byte integer, which reserves its eighth bit for the sign, can range from -127 to +128. See “signed integer,” “MSB.”

user-defined type A data type defined by the user. It is usually a structure, union, record, or pointer.

V

variable declaration A statement that initializes and allocates storage for a variable of a given type.

virtual disk A portion of the computer’s random access memory reserved for use as a simulated disk drive. Also called an electronic disk or RAM disk. Unless saved to a physical disk, the contents of a virtual disk are lost when the computer is turned off.

virtual memory Memory space allocated on a disk, rather than in RAM. Virtual memory allows large data structures that would not fit in conventional memory, at the expense of slow access.

visibility The characteristic of a variable or function that describes the parts of the program in which it can be accessed. An item has global visibility if it can be referenced in every source file constituting the program. Otherwise, it has local visibility.

W

watch window The window in CodeView that displays watch statements and their values. A variable or expression is watchable only while execution is occurring in the section of the program (context) in which the item is defined.

window A discrete area of the screen in PWB or CodeView used to display part of a file or to enter statements.

window commands Commands that work only in CodeView’s window mode. Window commands consist of function keys, mouse selections, CTRL and ALT key combinations, and selections from pop-up menus.

window mode The mode in which CodeView displays separate windows, which can change independently. CodeView has mouse support and a wide variety of window commands in window mode.

word A data unit containing 16 bits (2 bytes). It can store values from 0 to 65,535 (or -32,768 to +32,767).