

1. Record Nr.	UNINA9910458905603321
Autore	Hyde Randall
Titolo	The art of Assembly language [[electronic resource] /] / by Randall Hyde
Pubbl/distr/stampa	San Francisco, : No Starch Press, 2010
ISBN	1-59327-301-0
Edizione	[2nd ed.]
Descrizione fisica	1 online resource (764 p.)
Disciplina	005.13/6
Soggetti	Assembler language (Computer program language) Programming languages (Electronic computers) Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Contents in Detail; Acknowledgements; Chapter 1: Hello, World of Assembly Language; 1.1: The Anatomy of an HLA Program; 1.2: Running Your First HLA Program; 1.3: Some Basic HLA Data Declarations; 1.4: Boolean Values; 1.5: Character Values; 1.6: An Introduction to the Intel 80x86 CPU Family; 1.7: The Memory Subsystem; 1.8: Some Basic Machine Instructions; 1.9: Some Basic HLA Control Structures; 1.10: Introduction to the HLA Standard Library; 1.11: Additional Details About try..endtry; 1.12: High-Level Assembly Language vs. Low-Level Assembly Language; 1.13: For More Information Chapter 2: Data Representation 2.1: Numbering Systems; 2.2: The Hexadecimal Numbering System; 2.3: Data Organization; 2.4: Arithmetic Operations on Binary and Hexadecimal Numbers; 2.5: A Note About Numbers vs. Representation; 2.6: Logical Operations on Bits; 2.7: Logical Operations on Binary Numbers and Bit Strings; 2.8: Signed and Unsigned Numbers; 2.9: Sign Extension, Zero Extension, Contraction, and Saturation; 2.10: Shifts and Rotates; 2.11: Bit Fields and Packed Data; 2.12: An Introduction to Floating-Point Arithmetic; 2.13: Binary-Coded Decimal Representation; 2.14: Characters 2.15: The Unicode Character Set 2.16: For More Information; Chapter 3: Memory Access and Organization; 3.1: The 80x86 Addressing Modes; 3.2: Runtime Memory Organization; 3.3: How HLA Allocates Memory for

Variables; 3.4: HLA Support for Data Alignment; 3.5: Address Expressions; 3.6: Type Coercion; 3.7: Register Type Coercion; 3.8: The stack Segment and the push and pop Instructions; 3.9: The Stack Is a LIFO Data Structure; 3.10: Accessing Data You've Pushed onto the Stack Without Popping It; 3.11: Dynamic Memory Allocation and the Heap Segment; 3.12: The inc and dec Instructions
3.13: Obtaining the Address of a Memory Object
3.14: For More Information; Chapter 4: Constants, Variables, and Data Types; 4.1: Some Additional Instructions: intmul, bound, into; 4.2: HLA Constant and Value Declarations; 4.3: The HLA Type Section; 4.4: enum and HLA Enumerated Data Types; 4.5: Pointer Data Types; 4.6: Composite Data Types; 4.7: Character Strings; 4.8: HLA Strings; 4.9: Accessing the Characters Within a String; 4.10: The HLA String Module and Other String-Related Routines; 4.11: In-Memory Conversions; 4.12: Character Sets; 4.13: Character Set Implementation in HLA
4.14: HLA Character Set Constants and Character Set Expressions
4.15: Character Set Support in the HLA Standard Library; 4.16: Using Character Sets in Your HLA Programs; 4.17: Arrays; 4.18: Declaring Arrays in Your HLA Programs; 4.19: HLA Array Constants; 4.20: Accessing Elements of a Single-Dimensional Array; 4.21: Sorting an Array of Values; 4.22: Multidimensional Arrays; 4.23: Allocating Storage for Multidimensional Arrays; 4.24: Accessing Multidimensional Array Elements in Assembly Language; 4.25: Records; 4.26: Record Constants; 4.27: Arrays of Records
4.28: Arrays/Records as Record Fields

Sommario/riassunto

Widely respected by hackers of all kinds, *The Art of Assembly Language* teaches programmers how to understand assembly language and how to use it to write powerful, efficient code. Using the proven High Level Assembler (HLA) as its primary teaching tool, *The Art of Assembly Language* leverages your knowledge of high level programming languages to make it easier for you to quickly grasp basic assembly concepts. Among the most comprehensive references to assembly language ever published, *The Art of Assembly Language*, 2nd Edition has been thoroughly updated to reflect recent changes to the HLA lang
