

X86 Disassembly

[Wikibooks.org](https://en.wikibooks.org/wiki/X86_Disassembly)

March 13, 2013

On the 28th of April 2012 the contents of the English as well as German Wikibooks and Wikipedia projects were licensed under Creative Commons Attribution-ShareAlike 3.0 Unported license. An URI to this license is given in the list of figures on page 185. If this document is a derived work from the contents of one of these projects and the content was still licensed by the project under this license at the time of derivation this document has to be licensed under the same, a similar or a compatible license, as stated in section 4b of the license. The list of contributors is included in chapter Contributors on page 181. The licenses GPL, LGPL and GFDL are included in chapter Licenses on page 189, since this book and/or parts of it may or may not be licensed under one or more of these licenses, and thus require inclusion of these licenses. The licenses of the figures are given in the list of figures on page 185. This PDF was generated by the L^AT_EX typesetting software. The L^AT_EX source code is included as an attachment (`source.7z.txt`) in this PDF file. To extract the source from the PDF file, we recommend the use of <http://www.pdfplabs.com/tools/pdftk-the-pdf-toolkit/> utility or clicking the paper clip attachment symbol on the lower left of your PDF Viewer, selecting **Save Attachment**. After extracting it from the PDF file you have to rename it to `source.7z`. To uncompress the resulting archive we recommend the use of <http://www.7-zip.org/>. The L^AT_EX source itself was generated by a program written by Dirk Hünninger, which is freely available under an open source license from http://de.wikibooks.org/wiki/Benutzer:Dirk_Huenniger/wb2pdf. This distribution also contains a configured version of the `pdflatex` compiler with all necessary packages and fonts needed to compile the L^AT_EX source included in this PDF file.

Contents

1	Assemblers and Compilers	3
1.1	Assemblers	3
1.2	Assembler Concepts	3
1.3	Intel Syntax Assemblers	4
1.4	(x86) AT&T Syntax Assemblers	5
1.5	Other Assemblers	5
1.6	Compilers	6
1.7	Common C/C++ Compilers	7
2	Disassemblers and Decompilers	11
2.1	What is a Disassembler?	11
2.2	x86 Disassemblers	11
2.3	Disassembler Issues	15
2.4	Decompilers	16
2.5	Common Decompilers	17
2.6	Disassembly of 8 bit CPU code	17
2.7	Disassembly of 32 bit CPU code	18
2.8	A brief list of disassemblers	18
2.9	Further reading	19
3	Analysis Tools	21
3.1	Debuggers	21
3.2	Hex Editors	24
3.3	Other Tools for Windows	29
3.4	GNU Tools	30
3.5	Other Tools for Linux	31
4	Microsoft Windows	33
4.1	Microsoft Windows	33
4.2	Windows Versions	33
4.3	Virtual Memory	34
4.4	System Architecture	34
4.5	System calls and interrupts	34
4.6	Win32 API	35
4.7	Native API	35
4.8	ntoskrnl.exe	36
4.9	Win32K.sys	37
4.10	Win64 API	37
4.11	Windows Vista	37
4.12	Windows CE/Mobile, and other versions	37

4.13	"Non-Executable Memory"	37
4.14	COM and Related Technologies	38
4.15	Remote Procedure Calls (RPC)	38
5	Windows Executable Files	39
5.1	MS-DOS COM Files	39
5.2	MS-DOS EXE Files	39
5.3	PE Files	40
5.4	Relative Virtual Addressing (RVA)	40
5.5	File Format	41
5.6	Code Sections	46
5.7	Imports and Exports - Linking to other modules	48
5.8	Exports	49
5.9	Imports	50
5.10	Resources	52
5.11	Relocations	53
5.12	Alternate Bound Import Structure	53
5.13	Windows DLL Files	53
6	Linux	57
6.1	Linux	57
6.2	System Architecture	57
6.3	Configuration Files	58
6.4	Shells	58
6.5	GUIs	58
6.6	Debuggers	58
6.7	File Analyzers	59
7	Linux Executable Files	61
7.1	a.out Files	61
7.2	ELF Files	61
7.3	Relocatable ELF Files	62
8	The Stack	63
8.1	The Stack	63
8.2	Push and Pop	64
8.3	ESP In Action	65
8.4	Reading Without Popping	66
8.5	Data Allocation	66
9	Functions and Stack Frames	67
9.1	Functions and Stack Frames	67
9.2	Standard Entry Sequence	67
9.3	Standard Exit Sequence	69
9.4	Non-Standard Stack Frames	70
9.5	Local Static Variables	71
10	Functions and Stack Frame Examples	73
10.1	Example: Number of Parameters	73

10.2	Example: Standard Entry Sequences	74
11	Calling Conventions	75
11.1	Calling Conventions	75
11.2	Notes on Terminology	75
11.3	Standard C Calling Conventions	77
11.4	C++ Calling Convention	80
11.5	Note on Name Decorations	81
11.6	further reading	82
12	Calling Convention Examples	83
12.1	Microsoft C Compiler	83
12.2	GNU C Compiler	87
12.3	Example: C Calling Conventions	90
12.4	Example: Named Assembly Function	91
12.5	Example: Unnamed Assembly Function	91
12.6	Example: Another Unnamed Assembly Function	91
12.7	Example: Name Mangling	92
13	Branches	93
13.1	Branching	93
13.2	If-Then	93
13.3	If-Then-Else	95
13.4	Switch-Case	96
13.5	Ternary Operator ?:	101
14	Branch Examples	103
14.1	Example: Number of Parameters	103
14.2	Example: Identify Branch Structures	104
14.3	Example: Convert To C	106
15	Loops	109
15.1	Loops	109
15.2	Do-While Loops	109
15.3	While Loops	111
15.4	For Loops	112
15.5	Other Loop Types	113
16	Loop Examples	115
16.1	Example: Identify Purpose	115
16.2	Example: Complete C Prototype	115
16.3	Example: Decompile To C Code	116
17	Variables	119
17.1	Variables	119
17.2	How to Spot a Variable	119
17.3	.BSS and .DATA sections	120
17.4	"Static"Local Variables	120
17.5	Signed and Unsigned Variables	121

[Click here to download full PDF material](#)