$ ./hello_macho
Hello world

Dissection of an Intel 32-bits, 204 bytes, Mach-O file with 1 segment, 1 VM page and no libraries.

```
$ shasum hello_macho
 29866d22f3c262eb1ac96f520f78559311875281
```

http://seriot.ch/hello_macho.php — Nicolas Seriot, 2012-12 – 2013-01-03 17:45

	Offset	Actual bytes	Struct	Field	Value	Comment	Summary
Mach Header	0x00	**CE FA ED FE**	mach_header	magic	MH_MAGIC	mach magic number identifier	Mach-O executable file, 32 bits, i386
	0x04	**07** 00 00 00		cputype	CPU_TYPE_I386	cpu specifier	
	0x08	**03** 00 00 00		cpusubtype	CPU_SUBTYPE_I386_ALL	machine specifier	
	0x0C	**02** 00 00 00		filetype	MH_EXECUTE	type of file	
	0x10	**02** 00 00 00		ncmds	2	number of load commands	
	0x14	**88** 00 00 00		sizeofcmds	0x88 (136)	the size of all the load commands	
	0x18	**01** 00 00 00		flags	MH_NOUNDEFS	flags	
Load Commands / LC_SEGMENT (__TEXT)	0x1C	**01** 00 00 00	segment_command	cmd	LC_SEGMENT	LC_SEGMENT	one .text segment to be loaded in a 1kB memory page
	0x20	**38** 00 00 00		cmdsize	0x38 (56)	includes sizeof section structs	
	0x24	**5F 5F 54 45**		segname	__TEXT	segment name	
	0x28	**58 54** 00 00					
	0x2C	00 00 00 00					
	0x30	00 00 00 00					
	0x34	**00** 00 00 00		vmaddr	0x0	memory address of this segment	
	0x38	**00 10** 00 00		vmsize	0x1000	memory size of this segment	
	0x3C	00 00 00 00		fileoff	0x0	file offset of this segment	
	0x40	**40** 00 00 00		filesize	0x40 (64)	amount to map from the file	
	0x44	**07** 00 00 00		maxprot	rwx	maximum VM protection	
	0x48	**05** 00 00 00		initprot	r-x	initial VM protection	
	0x4C	**00** 00 00 00		nsects	0	number of sections in segment	
	0x50	00 00 00 00		flags		flags	
Load Commands / LC_UNIXTHREAD	0x54	**05** 00 00 00	thread_command	cmd	LC_UNIXTHREAD	LC_UNIXTHREAD	the initial state of the registers, the entry point $eip is at 0xA4
	0x58	**50** 00 00 00		cmdsize	0x50 (80)	total size of this command	
	0x5C	**01** 00 00 00		flavor	x86_THREAD_STATE32	flavor of thread state	
	0x60	**10** 00 00 00		count	0x10 (16)	count of longs in thread state	
	0x64	00 00 00 00	i386_thread_state	eax	0		
	0x68	00 00 00 00		ebx	0		
	0x6C	00 00 00 00		ecx	0		
	0x70	00 00 00 00		edx	0		
	0x74	00 00 00 00		edi	0		
	0x78	00 00 00 00		esi	0		
	0x7C	00 00 00 00		ebp	0		
	0x80	00 00 00 00		esp	0		
	0x84	00 00 00 00		ss	0		
	0x88	00 00 00 00		eflags	0		
	0x8C	**A4** 00 00 00		eip	0xA4		
	0x90	00 00 00 00		cs	0		
	0x94	00 00 00 00		ds	0		
	0x98	00 00 00 00		es	0		
	0x9C	00 00 00 00		fs	0		
	0xA0	00 00 00 00		gs	0		
Data / Section __TEXT	**0xA4**	**6A 0C**		push byte 12		text length	write(stdout, "Hello world\n", 12)
	0xA6	**68 C0 00 00 00**		push dword 0xC0		text address	
	0xAB	**6A 01**		push byte 1		stdout	
	0xAD	**B0 04**		mov byte eax, 4		code for 'write'	
	0xAF	**83 EC 04**		sub byte esp, 4		prepare syscall	
	0xB2	**CD 80**		int 0x80		syscall	
	0xB4	**83 C4 10**		add byte esp, 16		pop arguments	
	0xB7	**6A 00**		push byte 0		exit status	exit(0)
	0xB9	**B0 01**		mov byte eax, 0x1		code for 'exit'	
	0xBB	**83 EC 04**		sub byte esp, 4		prepare syscall	
	0xBE	**CD 80**		int 0x80		syscall	
	0xC0	**48 65 6C 6C 6F 20**		db 'Hello '		'Hello '	"Hello World\n" definition
	0xC6	**77 6F 72 6C 64 0A**		db 'world', 0Ah		'world\n'	
	Offset	Opcodes + arguments		Assembly Mnemonics + parameters		Comment	Summary