

Princeton University
COS 217: Introduction to Programming Systems
Assembler Output for hello

Symbol Table

Label	Section	Byte Offset	Local/Global	Type	Label Sequence #
pcGreeting	rodata	0	local	?	0
main	text	0	global	function	1
printf	?	?	local	?	2

Relocation Records

Section	Byte Offset	Relocation Type	Label Sequence #
text	3	32 bit absolute	0
text	8	32 bit displacement	2

Rodata Section

Byte Offset	Contents (hex)	Explanation
0	48	.asciz "Hello\n"
1	65	
2	6C	
3	6C	
4	6F	
5	0A	
6	00	

Text Section

Byte Offset	Contents (hex)	Explanation
0	55	pushl %ebp 01010101 This is a "pushl %ebp" instruction
1-2	89 E5	movl %esp, %ebp 10001001 11 100 101 This is a "movl" instruction whose source operand is a register The M field designates a register The source register is %esp The destination register is %ebp
3-7	68????????	pushl \$pcGreeting 01101000 ????????????????????????????????????? This is a "pushl" instruction with a 4 byte immediate operand This is the data to be pushed
8-12	E8????????	call printf 11101000 ????????????????????????????????????? This is a "call" instruction with a 4 byte immediate operand This is the displacement to the instruction to be called.
13-15	83C404	addl \$4, %esp 10000011 11 000 100 00000100 This is some "l" instruction that has a 1 byte immediate operand The M field designates a register This is an "add" instruction The destination register is %esp The immediate operand is 4
16-20	B800000000	movl \$0, %eax 10111000 00000000000000000000000000000000 This is an instruction of the form "movl 4-byte-immediate, %eax" The immediate operand is 0
21-22	89EC	movl %ebp, %esp 10001001 11 101 100 This is a "movl" instruction whose source operand is a register The M field designates a register The source register is %ebp The destination register is %esp
23	5D	popl %ebp 01011101 This is a "popl %ebp" instruction
24	C3	ret 11000011 This is a "ret" instruction