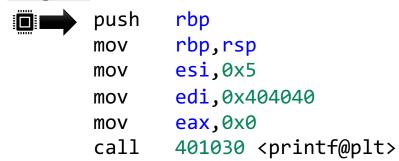
Practice: Disassembling a 64-bit Function Call Idiom

Given the information presented here about a *64-bit ELF* program, complete the chart and describe what the program prints. Reference information about the calling conventions for the System V (Linux) ABI can be found here: https://wiki.osdev.org/System V ABI#x86-64.

Registers

Register	Value	Register	Value
rax		rbp	0x7ff50
rbx		rsp	0x7ff20
rcx		rsi	
rdx		rdi	

<u>Program</u>



Stack Memory

Start Address	End Address	Contents
0x7ff30	0x7ff37	
0x7ff28	0x7ff2f	
0x7ff20	0x7ff27	
0x7ff18	0x7ff1f	
0x7ff10	0x7ff17	
0x7ff08	0x7ff0f	
0x7ff00	0x7ff07	
0x7fef8	0x7feff	
0x7fef0	0x7fef7	
0x7fee8	0x7feef	
0x7fee0	0x7fee7	
0x7fed8	0x7fedf	

On-Disk Data (in **hexdump** –**C** format)

00003010	20	3е	40	00	00	00	00	00	00	00	00	00	00	00	00	00	>@
00003020	00	00	00	00	00	00	00	00	36	10	40	00	00	00	00	00	6.@
00003030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	1
00003040	25	64	20	69	73	20	79	6f	75	72	20	6c	75	63	6b	79	%d is your lucky
00003050	20	6e	75	6d	62	65	72	20	2d	2d	20	6d	65	6e	74	69	number menti
00003060	6f	6e	20	69	74	20	61	6e	64	20	67	65	74	20	61	20	on it and $get a \mid$
00003070	53	74	61	72	62	75	63	6b	73	20	67	69	66	74	20	63	Starbucks gift c
00003080	61	72	64	2e	0a	00	47	43	43	3a	20	28	47	4e	55	29	ardGCC: (GNU)
00003090	20	31	32	2e	32	2e	31	20	32	30	32	32	31	31	32	31	12.2.1 20221121
000030a0	20	28	52	65	64	20	48	61	74	20	31	32	2e	32	2e	31	(Red Hat 12.2.1
000030b0	2d	34	29	00	80	00	00	00	10	00	00	00	00	01	00	00	-4)
000030c0	47	41	24	01	33	61	31	00	40	10	40	00	00	00	00	00	GA\$.3a1.@.@
		_	_			_											

ELF Header Information

There are 31 section headers, starting at offset 0x5a30:

Section Headers:

[Nr]	Name	Type	Addres	Offset		
	Size	EntSize	Flags	Link	Info	Align
[24]	.data	PROGBITS	000000	000040	4020	00003020
	0000000000000066	00000000000000000	WA	0	0	32