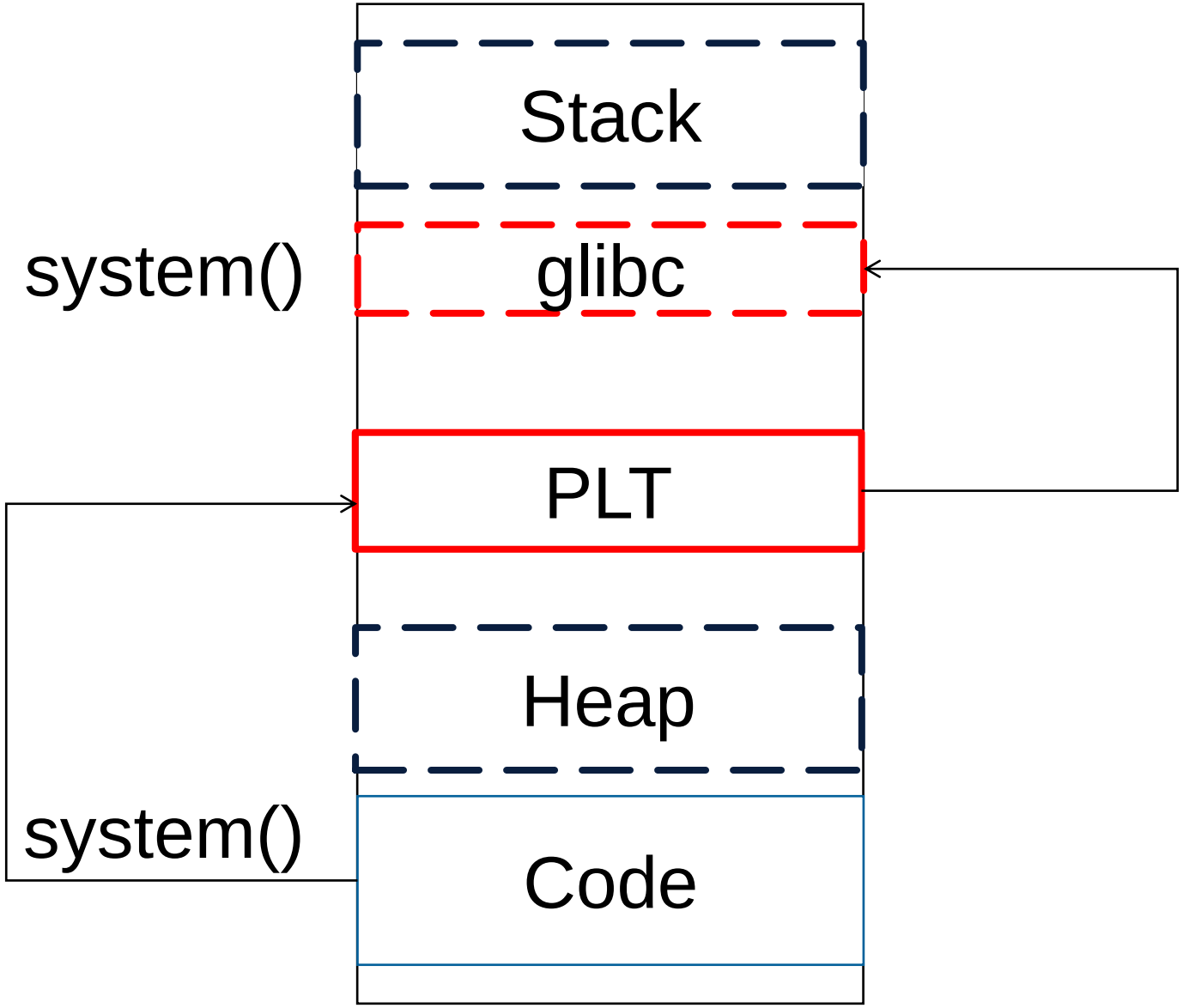
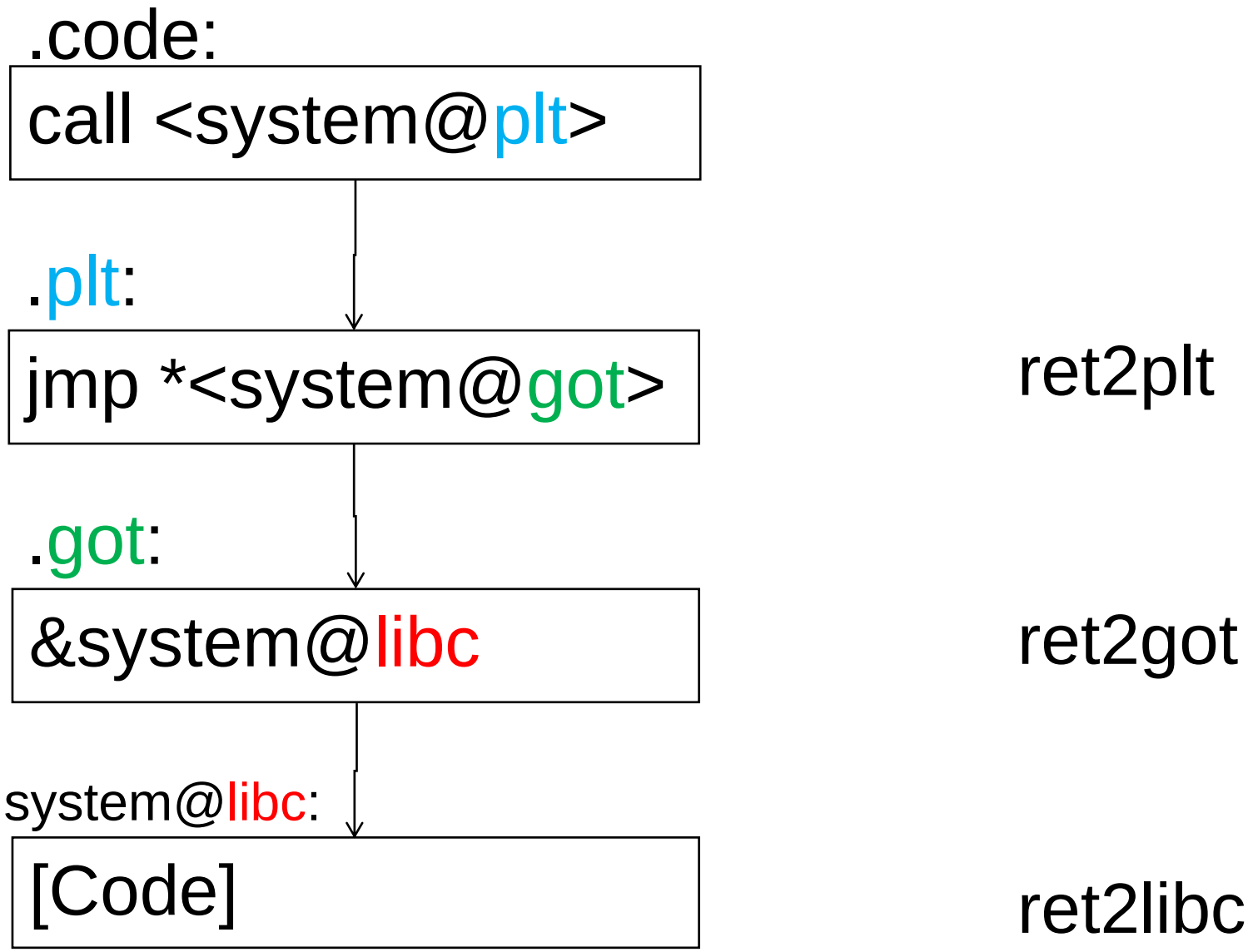
Recap ret2plt

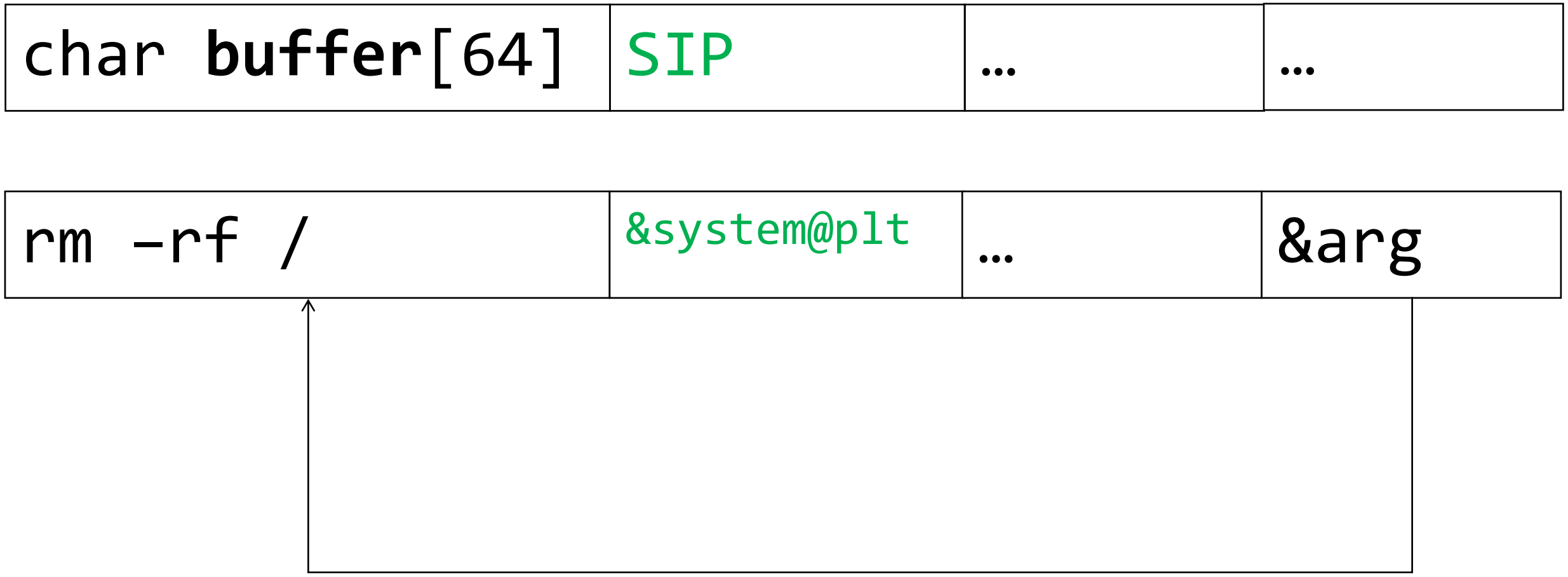
Defeating DEP – Shared Libraries Intro



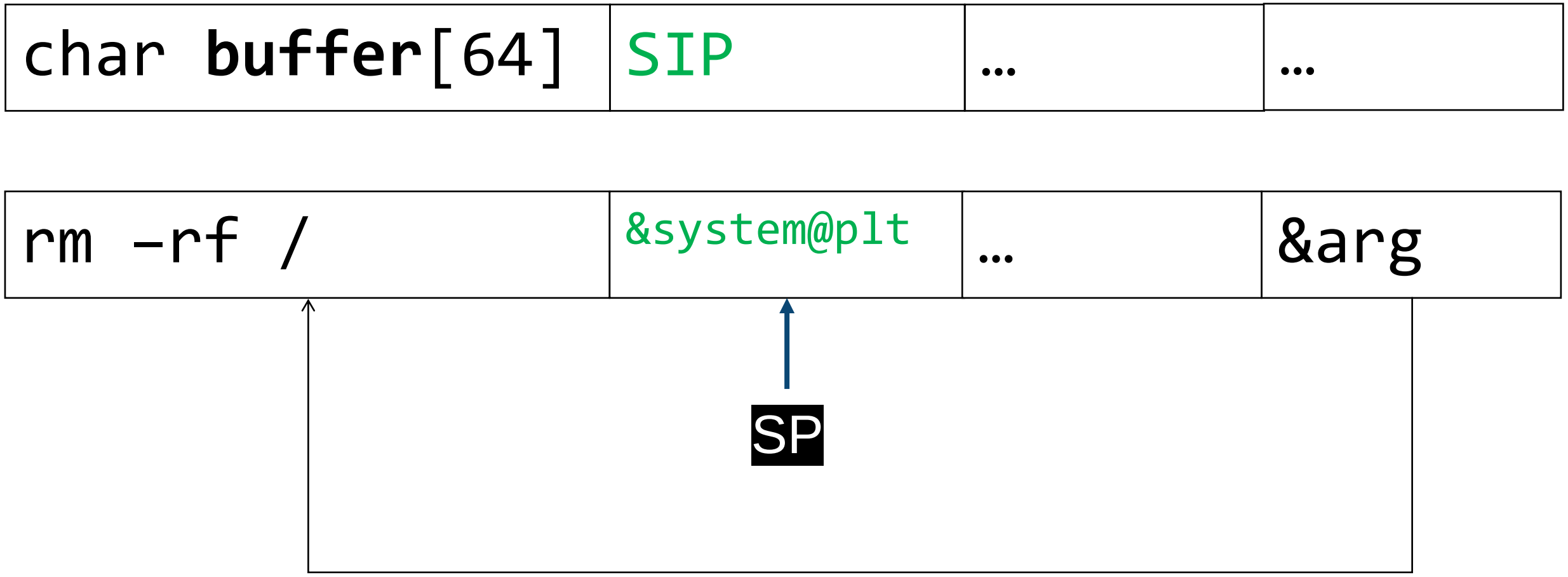
Return to X


.code:
call <system@plt>
.plt:
jmp *<system@got>
.got:
&system@libc
system@libc:
[Code]
ret2plt
ret2got
ret2libc

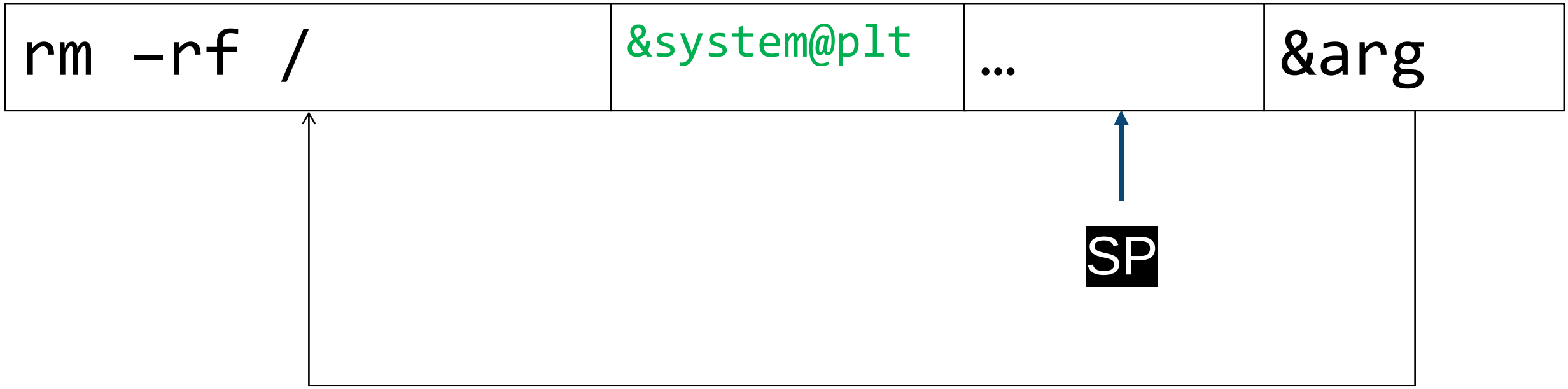
Ret2plt exploit in 32 bit



Ret2plt in 32 bit – before ret



Ret2plt in 32 bit – after ret



Stack Layout – after call

Saved IP (&__libc_start)

Saved Frame Pointer

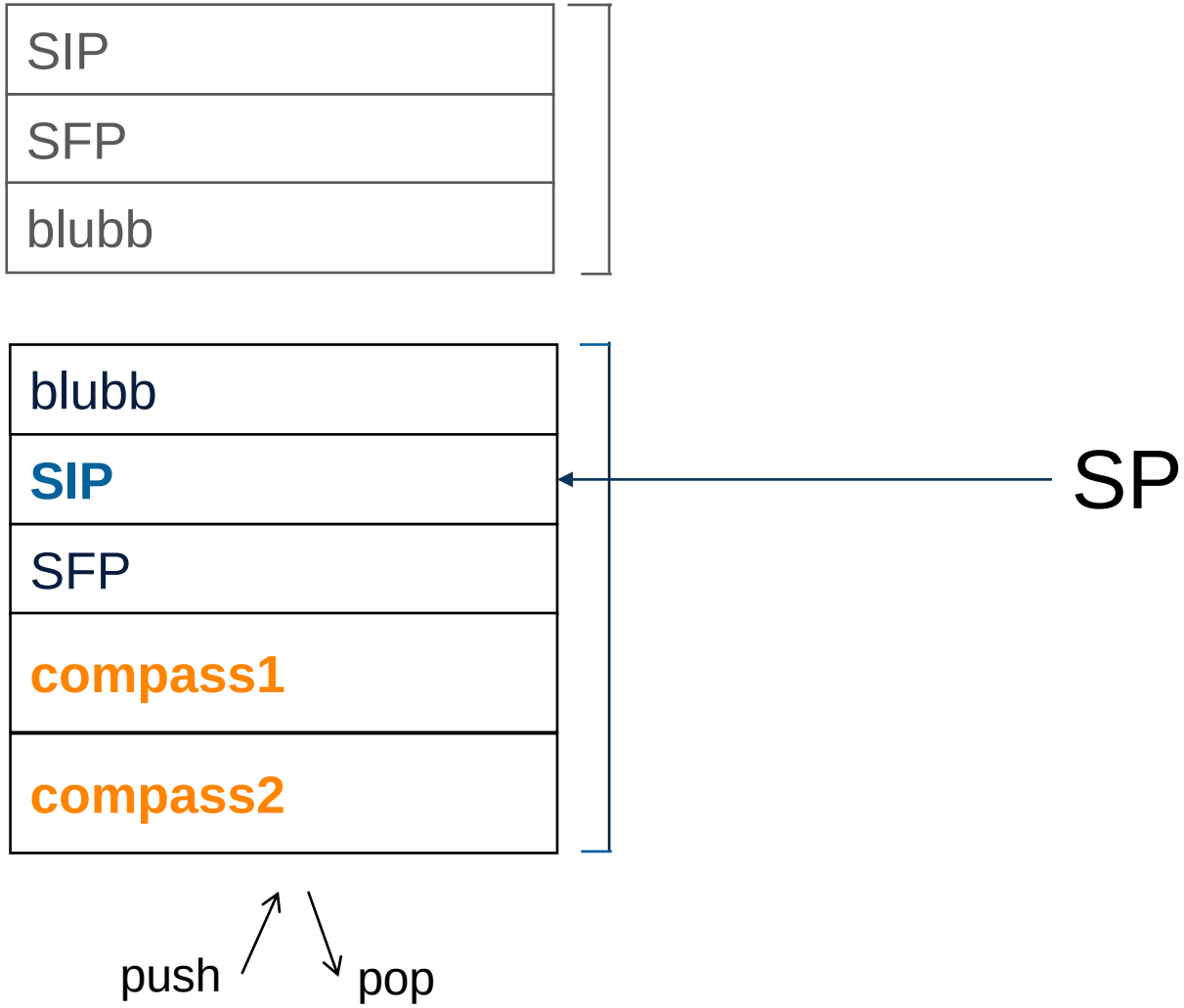
Local Variables <main>

Argument for <foobar>

Saved IP (&return)

Saved Frame Pointer

Local Variables <foobar>



Ret2plt in 64 bit

Function Call Convention Cheat Sheet

x32	Parameter	Syscall nr in
x32 userspace	`stack`	
x32 syscalls	`ebx, ecx, edx, esi, edi, ebp`	eax

x64	Parameter	Syscall nr in
x64 userspace	`rdi, rsi, rdx, rcx, r8, r9`	
x64 syscall	`rdi, rsi, rdx, r10, r8, r9`	rax

http://stackoverflow.com/questions/2535989/what-are-the-calling-conventions-for-unix-linux-system-calls-on-x86-64

Ret2plt exploit in 64 bit