

COMP201 Lab 7

Fall 2020

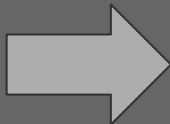
Runtime Stack



**KOÇ
UNIVERSITY**

Example Code

```
int foo1()
{
    int i = 2;
    return i;
}
int foo()
{
    int i = 5;
    return foo1();
}
```



```
0x0000000000400546 <foo1>:
    push rbp
    movq rsp, rbp
    sub 16, rsp
    movl $2, -0x4(rbp)
    movl -0x4(rbp), eax
    movq rbp, rsp
    pop rbp
    ret
0x0000000000400626 <foo>:
    push rbp
    movq rsp, rbp
    sub 16, rsp
    movl $5, -0x4(rbp)
    call 0x400546 <foo1>
    movq rbp, rsp
    pop rbp
    ret
```

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

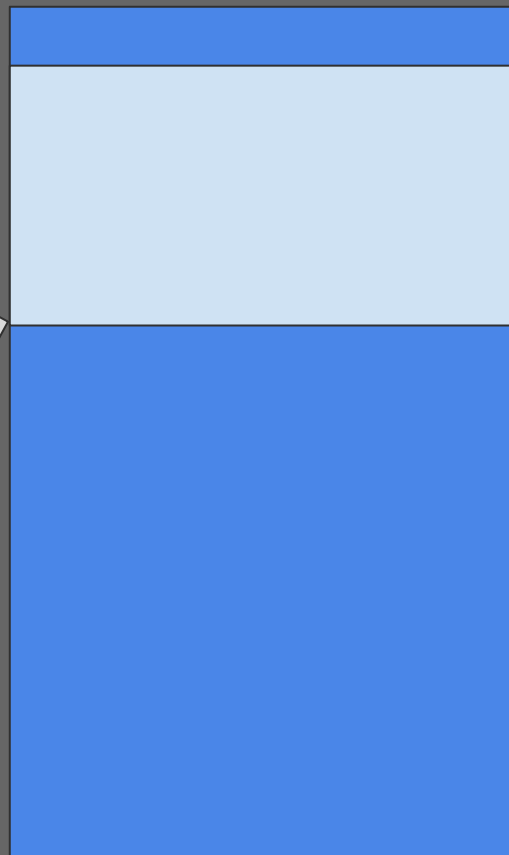
0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

rbp1

foo()

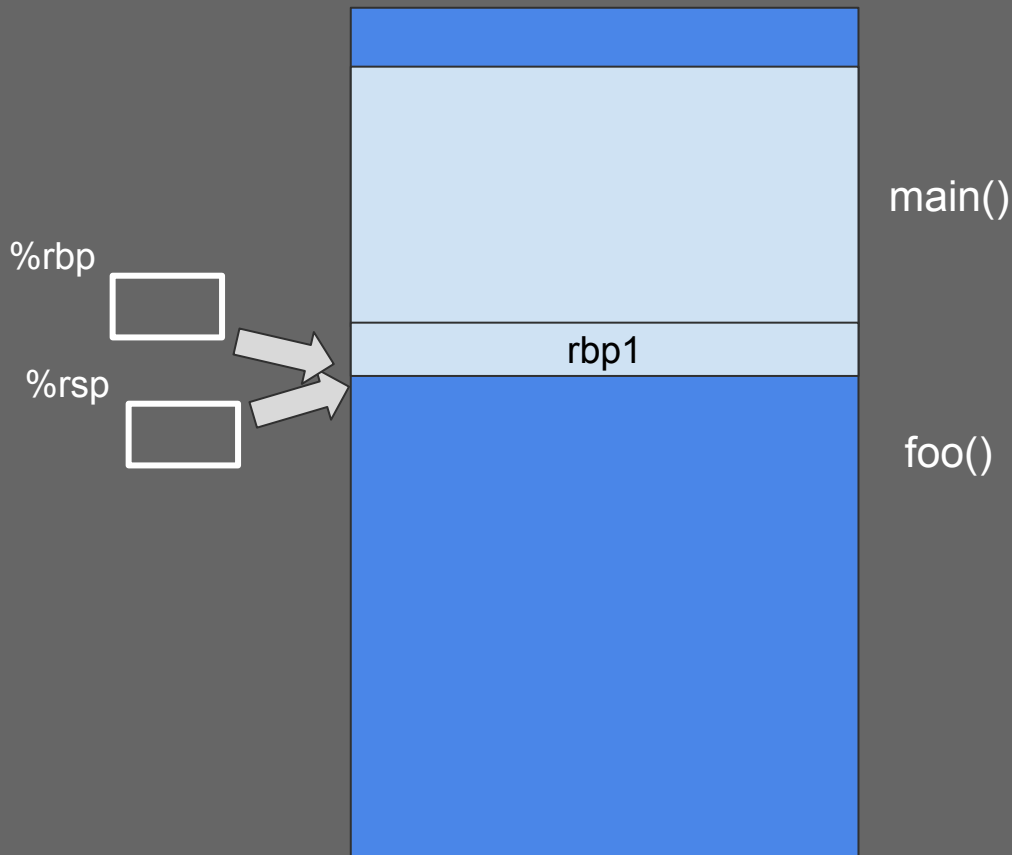
Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```



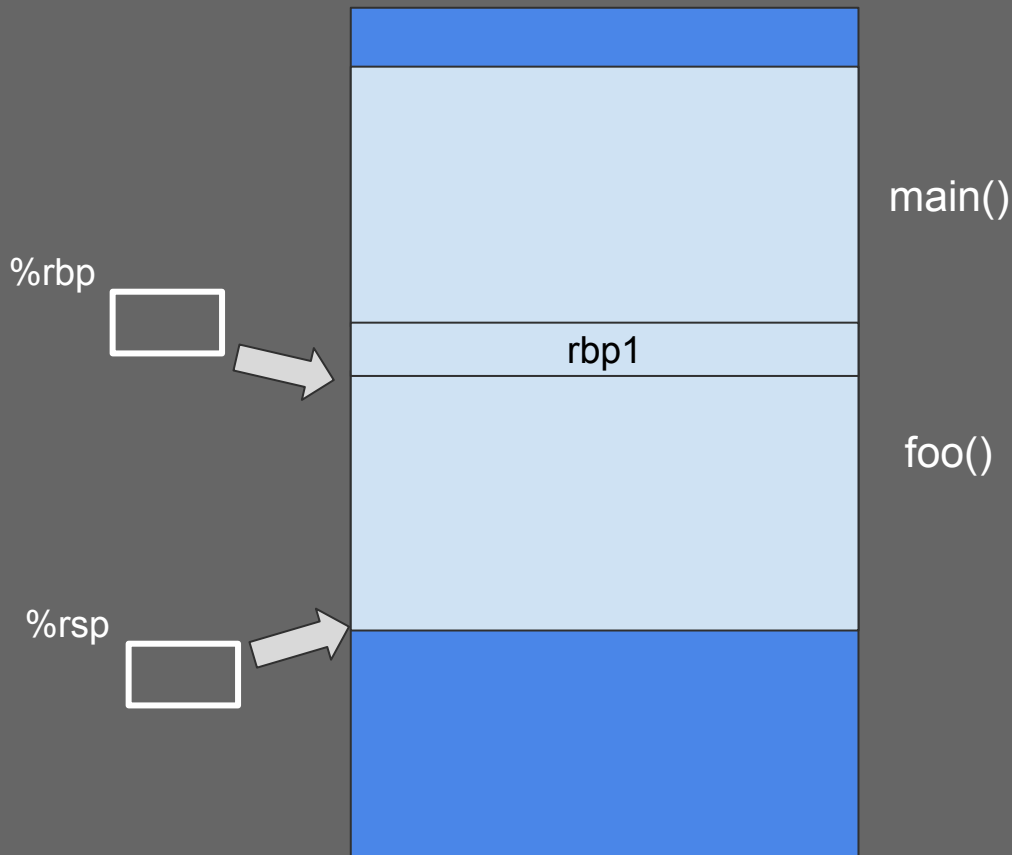
Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```



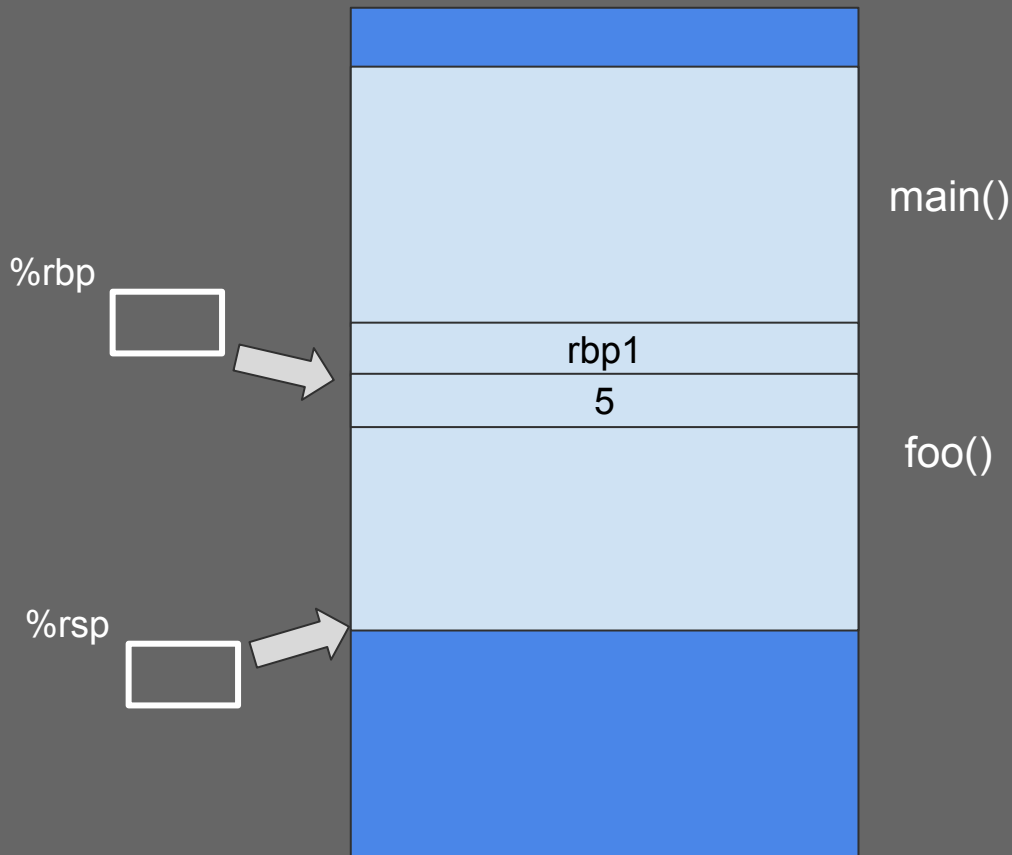
Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```



Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp ← ra1
pop rbp
ret
```

%rbp



%rsp



main()

foo()

call pushes the address of the next instruction (ra1) to the stack and puts the address of foo1 label to the program counter (%rip)

Execution Flow

0x0000000000400546 <foo1>:

push rbp

movq rsp, rbp

sub 16, rsp

movl \$2, -0x4(rbp)

movl -0x4(rbp), eax

movq rbp, rsp

pop rbp

ret

0x0000000000400626 <foo>:

push rbp

movq rsp, rbp

sub 16, rsp

movl \$5, -0x4(rbp)

call 0x400546 <foo1>

movq rbp, rsp

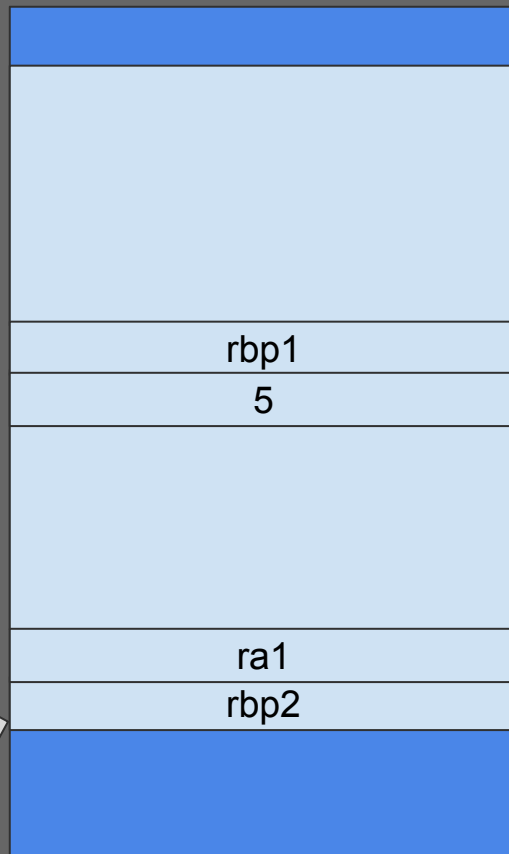
pop rbp

ret

%rbp



%rsp



main()

foo()

foo1()

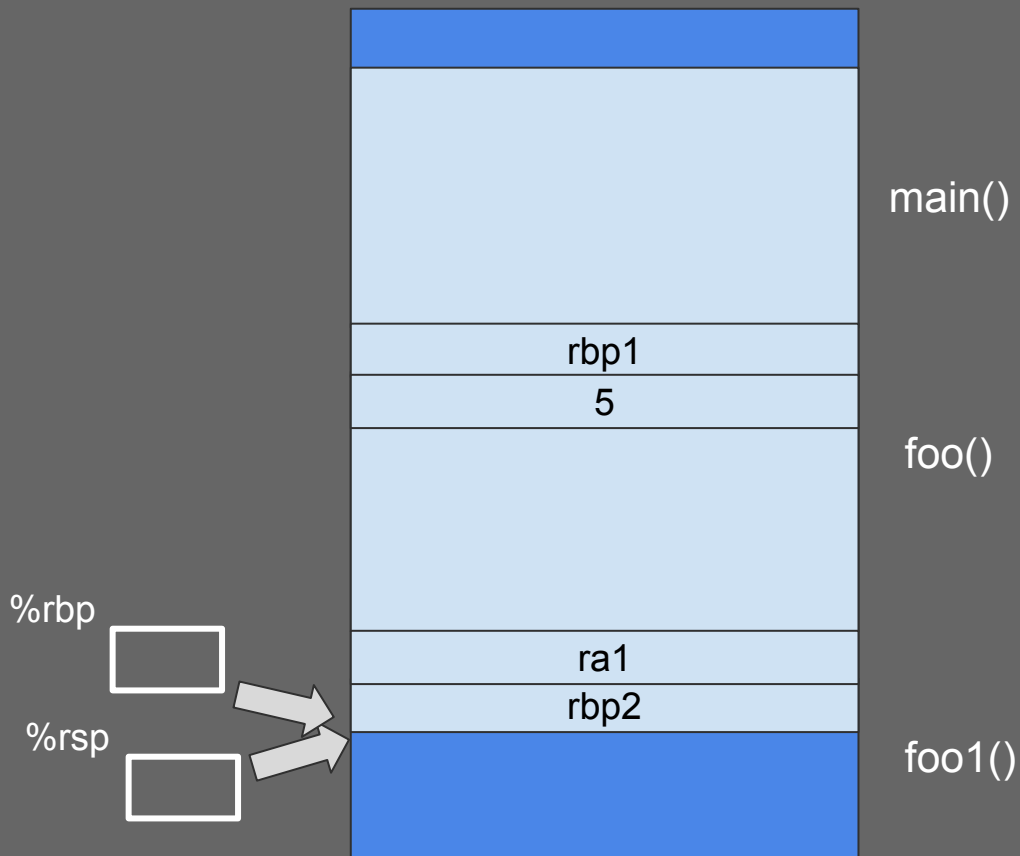
Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```



Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

foo()

foo1()

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

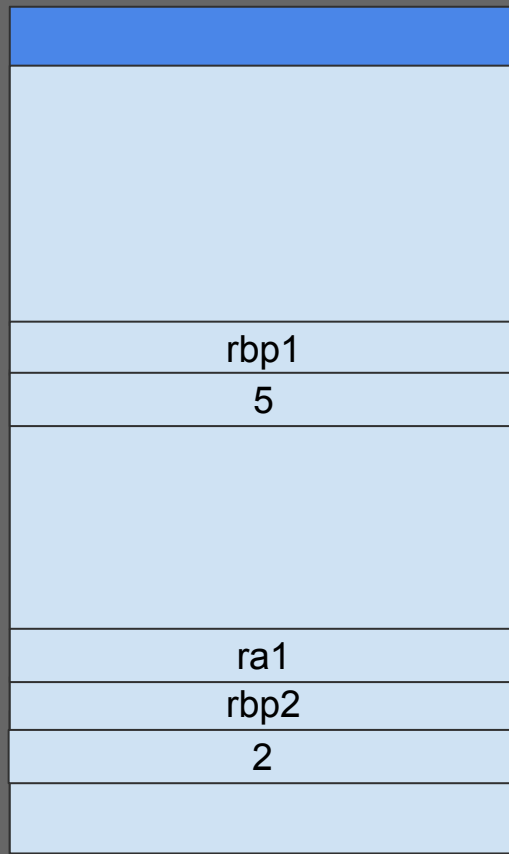
0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

foo()

foo1()

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

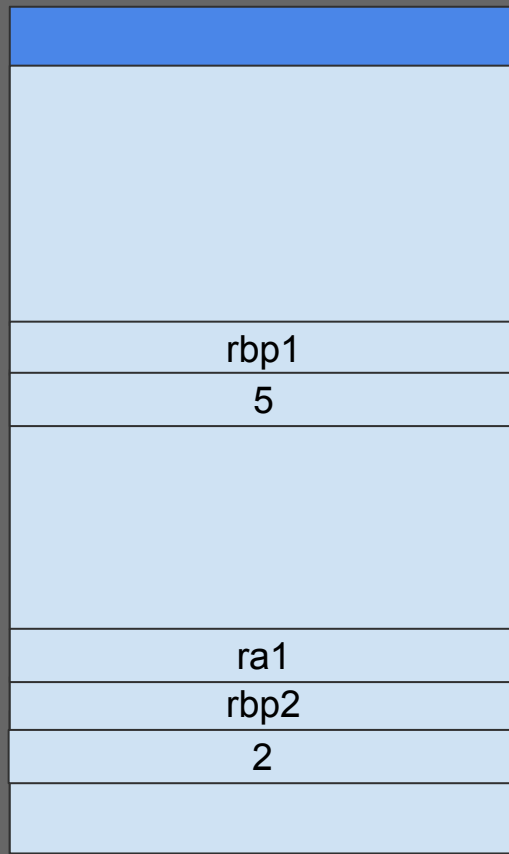
0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

foo()

foo1()

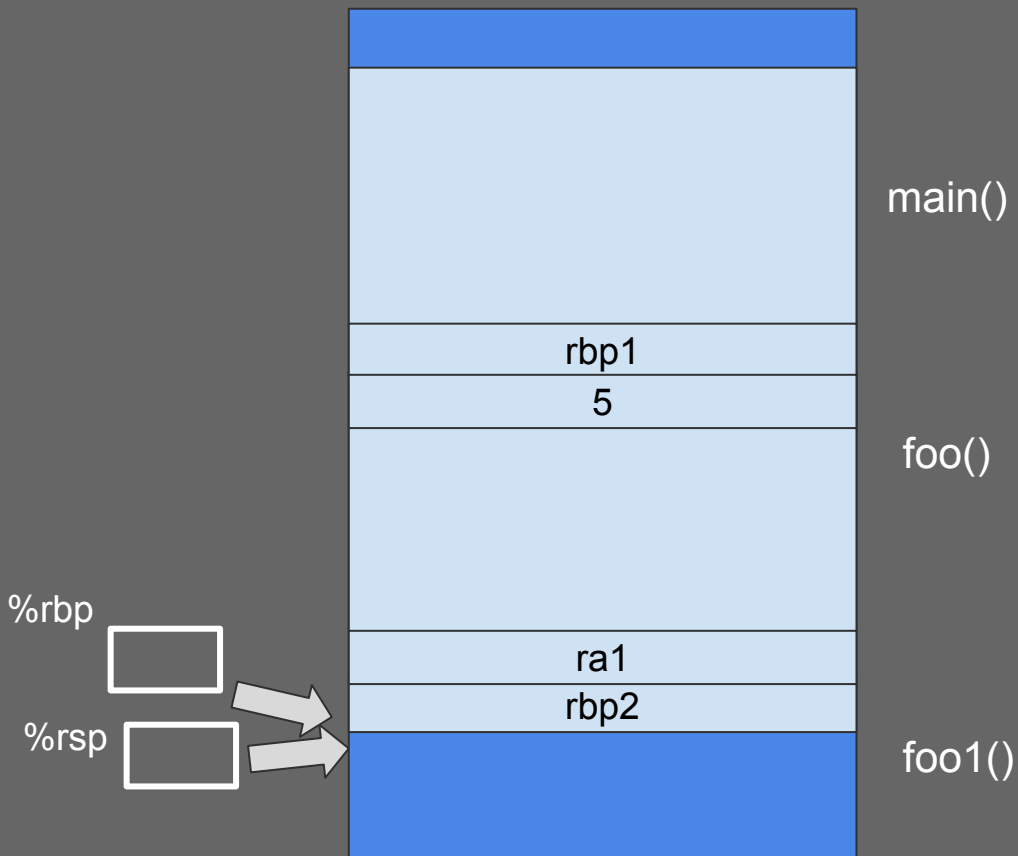
Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```



Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

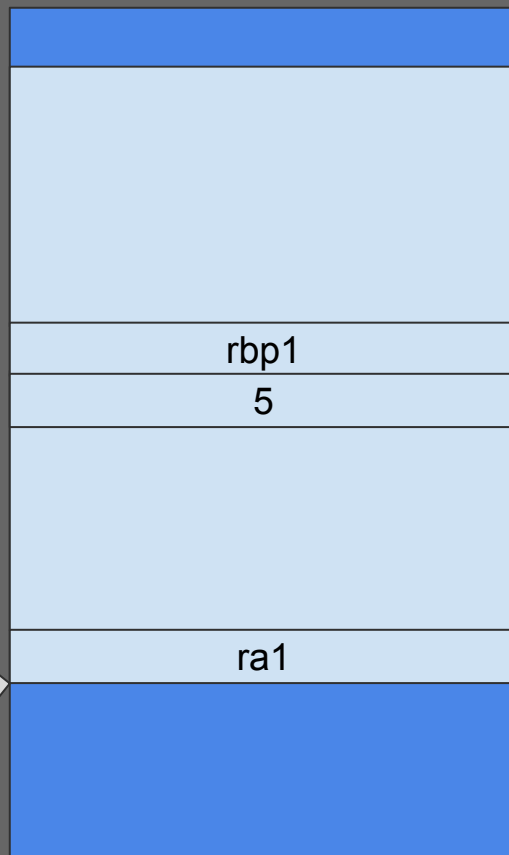
0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

foo()

Execution Flow

0x000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp ← ra1
pop rbp
ret
```

%rbp



%rsp



main()

foo()

ret pops the return address (ra1) from the stack and puts it to %rip.

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



rbp1

main()

foo()

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

Execution Flow

0x0000000000400546 <foo1>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $2, -0x4(rbp)
movl -0x4(rbp), eax
movq rbp, rsp
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
sub 16, rsp
movl $5, -0x4(rbp)
call 0x400546 <foo1>
movq rbp, rsp
pop rbp
ret
```

%rbp



%rsp



main()

How to pass parameters to a called function??

```
int foo1(int a, int b, int c)
{
    return a+b+c;
}
int foo()
{
    return foo1(1,2,3);
}
```



0x0000000000400546 <foo1>:

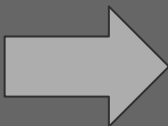
```
push rbp
movq rsp, rbp
movl edi, -0x4(rbp)
movl esi, -0x8(rbp)
movl edx, -0xc(rbp)
movl -0x4(rbp), edx
mov -0x8(rbp), eax
add eax, edx
mov -0xc(rbp), eax
add edx, eax
pop rbp
ret
```

0x0000000000400626 <foo>:

```
push rbp
movq rsp, rbp
movl $3, edx
movl $2, esi
movl $1, edi
call 0x400546 <foo1>
pop rbp
ret
```

How to pass parameters to a called function??

```
int foo1(int a, int b, int c, int d, int e,  
int f)  
{  
    .....  
}  
int foo()  
{  
    return foo1(1,2,3,4,5,6);  
}
```



0x0000000000400546 <foo1>:

.....

0x0000000000400626 <foo>:

push rbp

mov rsp, rbp

movl \$6, r9d

movl \$5, r8d

movl \$4, ecx

movl \$3, edx

movl \$2, esi

movl \$1, edi

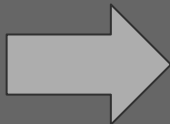
call 0x400546 <foo1>:

pop rbp

ret

How to pass parameters to a called function??

```
int foo1(int a, int b, int c, int d, int e,  
int f, int g, int h)  
{  
    .....  
}  
int foo()  
{  
    return foo1(1,2,3,4,5,6,7,8);  
}
```



0x0000000000400546 <foo1>:

.....

0x0000000000400626 <foo>:

```
push rbp  
mov rsp, rbp  
sub 16, rsp  
push 8  
push 7  
mov $6, r9d  
mov $5, r8d  
mov $4, ecx  
mov $3, edx  
mov $2, esi  
mov $1, edi  
call 0x400546 <foo1>  
add 16, rsp  
leave  
ret
```

Lab Practice