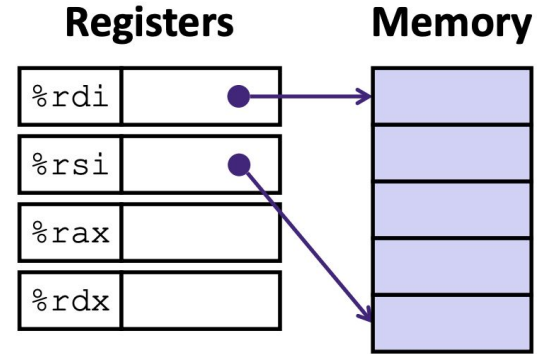


# Understanding swap ()

```
void swap(long* xp, long* yp) {  
    long t0 = *xp;  
    long t1 = *yp;  
    *xp = t1;  
    *yp = t0;  
}
```

swap:

```
movq (%rdi), %rax  
movq (%rsi), %rdx  
movq %rdx, (%rdi)  
movq %rax, (%rsi)  
ret
```



<u>Register</u>		<u>Variable</u>
%rdi	↔	xp
%rsi	↔	yp
%rax	↔	t0
%rdx	↔	t1

# Understanding `swap()`

## Registers

<code>%rdi</code>	<code>0x120</code>
<code>%rsi</code>	<code>0x100</code>
<code>%rax</code>	
<code>%rdx</code>	

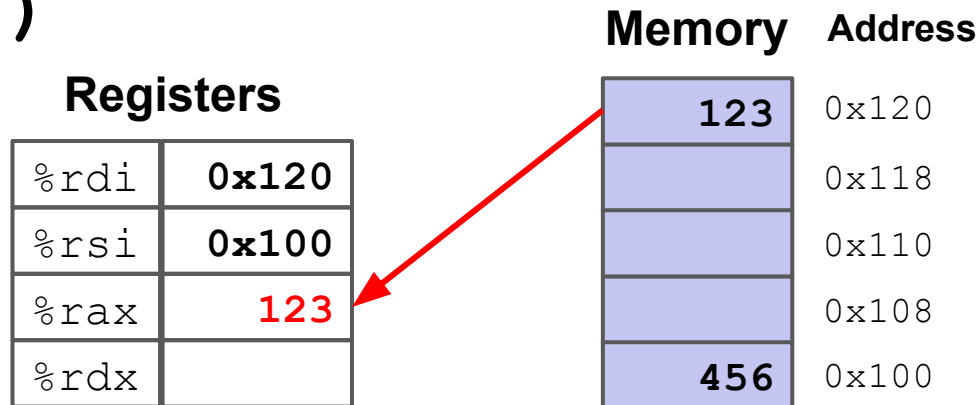
## Memory Address

<code>123</code>	<code>0x120</code>
	<code>0x118</code>
	<code>0x110</code>
	<code>0x108</code>
<code>456</code>	<code>0x100</code>

```
swap:
```

```
    movq (%rdi), %rax # t0 = *xp  
    movq (%rsi), %rdx # t1 = *yp  
    movq %rdx, (%rdi) # *xp = t1  
    movq %rax, (%rsi) # *yp = t0  
    ret
```

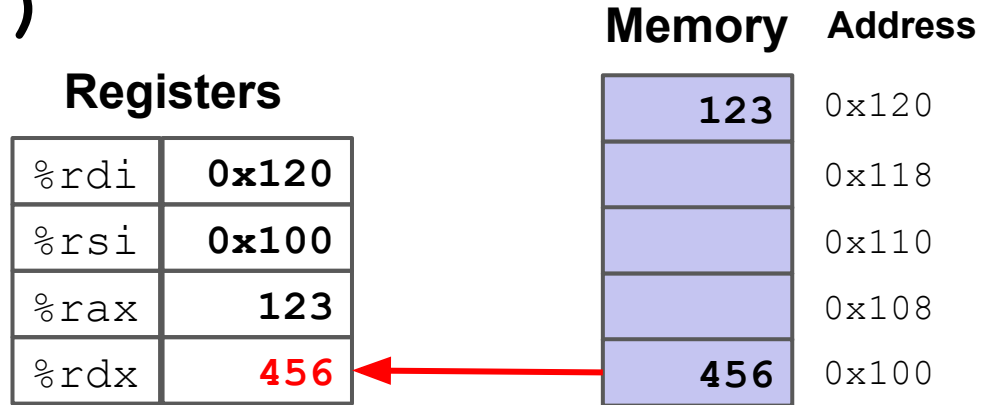
# Understanding `swap()`



```
swap:
```

```
movq (%rdi), %rax # t0 = *xp  
movq (%rsi), %rdx # t1 = *yp  
movq %rdx, (%rdi) # *xp = t1  
movq %rax, (%rsi) # *yp = t0  
ret
```

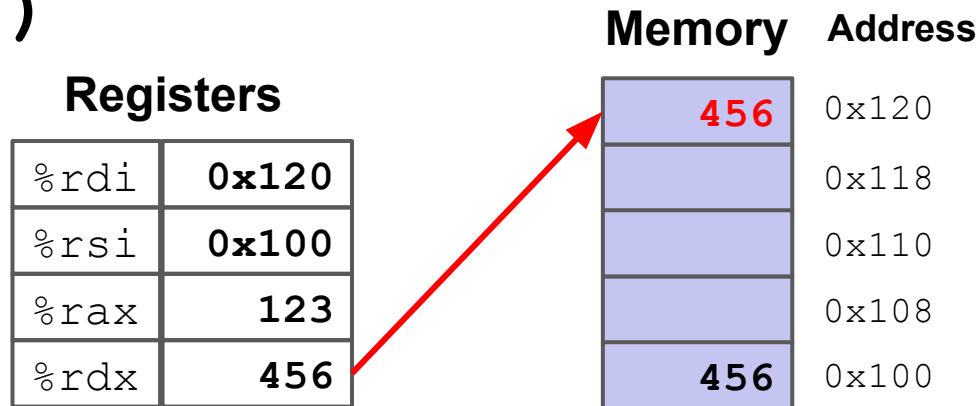
# Understanding `swap()`



`swap:`

```
movq (%rdi), %rax # t0 = *xp
movq (%rsi), %rdx # t1 = *yp
movq %rdx, (%rdi) # *xp = t1
movq %rax, (%rsi) # *yp = t0
ret
```

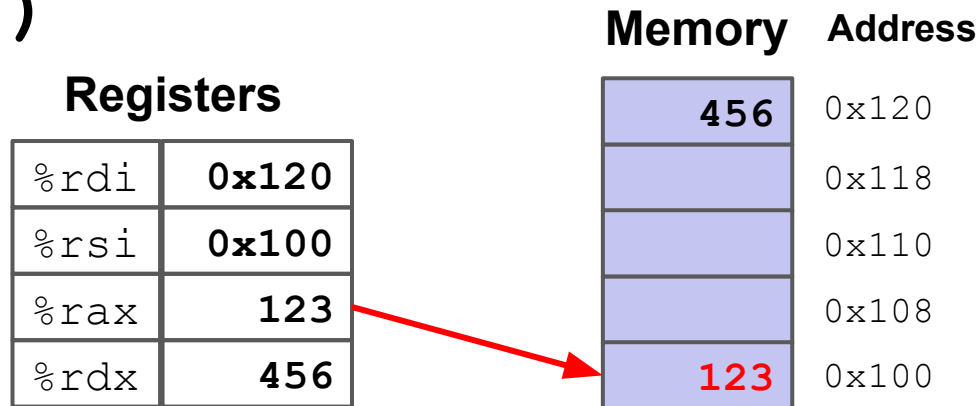
# Understanding `swap()`



```
swap:
```

```
    movq (%rdi), %rax # t0 = *xp  
    movq (%rsi), %rdx # t1 = *yp  
    movq %rdx, (%rdi) # *xp = t1  
    movq %rax, (%rsi) # *yp = t0  
    ret
```

# Understanding `swap()`



```
swap:
```

```
    movq (%rdi), %rax # t0 = *xp  
    movq (%rsi), %rdx # t1 = *yp  
    movq %rdx, (%rdi) # *xp = t1  
    movq %rax, (%rsi) # *yp = t0  
    ret
```